CentOS+Nginx+PHP+MySQL详细配置(图解)

转载 2011-03-22 作者: 我要评论

> CentOS+Nginx+PHP+MySQL详细配置(带有图解)，需要的朋友可以参考下。

一、安装MySQL

目前web服务器已经很少有跑静态页面的，如果要跑动态网站那当然就离不开数据库，虽然在以前文章中有写MySQL是怎么安装的，但是感觉好久没装MySQL，现在只把步骤贴出来，就不做过多的讲解了

```
#useradd mysql
#tar zxvf mysql-5.0.40.tar.gz
#cd mysql-5.0.40
#./configure --prefix=/usr/local/mysql
#make && make install
#/usr/local/mysql/bin/mysql_install_db --user=mysql        //初始化MySQL数据库
#chown -R mysql /usr/local/mysql/var
#/usr/local/mysql/bin/mysqld_safe &        //启动MySQL
#/usr/local/mysql/bin/mysqladmin -u root password 123456    //设置MySQL密码
#cp support-files/my-medium.cnf /etc/my.cnf
#echo "/usr/local/mysql/bin/mysqld_safe &" >>/etc/rc.local
```

二、安装PCRE

PCRE是perl所用到的正则表达式，目的是让所装的软件支持正则表达式。默认情况下，Nginx只处理静态的网页请求，也就是html.如果是来自动态的网页请求，比如*.php，那么Nginx就要根据正则表达式查询路径，然后把*.PHP交给PHP去处理

```
#rpm -qa | grep pcre            //查询系统中有没有安装PCRE，一般装系统是默认装有，所以我们要删掉系统自带的
#cp /lib/libpcre.so.0 /         //在删除系统自带的PCRE之前，要先备份一下libpcre.so.0这个文件，因为RPM包的关联性太强，在删除后没libpcre.so.0这个文件时我们装PCRE是装不上的
#rpm -e --nodeps pcre-6.6-1.1        //删除系统自带的PCRE
# tar zxvf pcre-8.00.tar.gz
#cd pcre-8.00
#cp /libpcre.so.0 /lib/            //把我们删除系统自带的PCRE之前备份的libpcre.so.0拷贝到/lib 目录下
```

#./configure //配置PCRE，因为PCRE是一个库，而不是像pache、php、postfix等这样的程序，所以我们安装时选择默认路径即可，这样会在后面安装其它东西时避免一些不必要的麻烦，执行完这部后会显示出下图，上面显示了我们对PCRE的配置

#make && make install

```
pcre-8.00 configuration summary:

    Install prefix ................. : /usr/local
    C preprocessor ................. : gcc -E
    C compiler ..................... : gcc
    C++ preprocessor ............... : g++ -E
    C++ compiler ................... : g++
    Linker ......................... : /usr/bin/ld
    C preprocessor flags ........... :
    C compiler flags ............... : -O2
    C++ compiler flags ............. : -O2
    Linker flags ................... :
    Extra libraries ................ :

    Build C++ library .............. : yes
    Enable UTF-8 support ........... : no
    Unicode properties ............. : no
    Newline char/sequence .......... : lf
    \R matches only ANYCRLF ........ : no
    EBCDIC coding .................. : no
    Rebuild char tables ............ : no
    Use stack recursion ............ : yes
    POSIX mem threshold ............ : 10
    Internal link size ............. : 2
    Match limit .................... : 10000000
    Match limit recursion .......... : MATCH_LIMIT
    Build shared libs ......
    Build static libs ......
    Link pcregrep with libz
    Link pcregrep with libbz
    Link pcretest with libre
```


三、安装Nginx

在网上，看到不少人装Nginx 时非常麻烦，配置时用了一大堆选项，请问你们真实现那么多功能么？害的我越看越郁闷。此次安装Nginx如果是按着上面笔者的步骤一步步走下来，安装Nginx时只需指定Nginx的安装路径即可

#tar zxvf nginx-0.8.24.tar.gz

#cd nginx-0.8.24

#./configure --prefix=/usr/local/nginx //此处在本环节只需指定一个路径

#make && make install

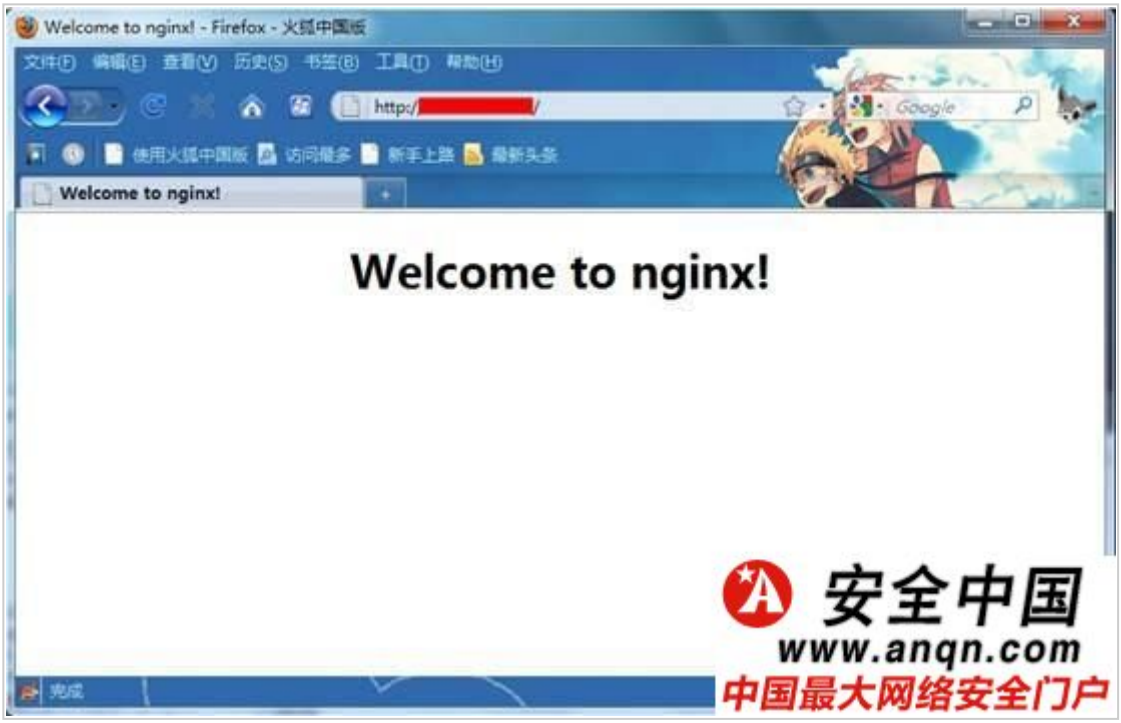
#/usr/local/nginx/sbin/nginx //启Nginx

#echo "/usr/local/nginx/sbin/nginx" >>/etc/rc.local

Nginx启动后有两个进程，master为主进程，worker为工作进程，如下图

```
[root@luwenju nginx-0.8.24]# ps -aux | grep nginx
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/
root     11578  0.0  0.3   3712   500 ?        Ss   21:23   0:00 nginx: ma
nobody   11579  0.0  0.6   3888   844 ?        S    21:23   0:00 nginx: wo
root     11583  0.0  0.5   3880   684 pts/2    S+   21:23   0:00 grep ngin
```


在启动完NGINX后，我们可以在浏览器中输入http://localhost查看，如下图



四、安装PHP

既然安装PHP，那GD便是不可少的，在此GD的安装不再进行描述

1、安装libpng

```
#tar xvf libpng-1.2.10.tar.tar
#cd libpng-1.2.10
#./configure --prefix=/usr/local/png
#make;make install
#ln -s /usr/local/png/lib/* /usr/lib/
```

2、安装jpeg

```
#mkdir /usr/local/jpeg
#mkdir /usr/local/jpeg/bin
#mkdir /usr/local/jpeg/lib
#mkdir /usr/local/jpeg/include
#mkdir /usr/local/jpeg/man
#mkdir /usr/local/jpeg/man/man1
#tar xvf jpegsrc.v7.tar.tar
#cd jpeg-7
#./configure --prefix=/usr/local/jpeg --enable-shared --enable-static
#make;make install
#ln -s /usr/local/jpeg/lib/* /usr/lib/
```

3、安装 freetype

```
#tar xvf freetype-2.3.9.tar.tar
#cd freetype-2.3.9
#./configure --prefix=/usr/local/freetype
#make;make install
```

4、安装fontconfig

```
#tar zxvf fontconfig-2.4.2.tar.gz
```

#cd fontconfig-2.4.2

#./configure --prefix=/usr/local/fontconfig --with-freetype-config=/usr/local/freetype/bin/freetype-config

#make;make install

5、安装GD

#tar zxvf gd-2.0.32.tar.gz

#cd gd-2.0.32

#./configure --prefix=/usr/local/gd --with-png=/usr/local/png --with-jpeg=/usr/local/jpeg --with- freetype=/usr/local/freetype --with-fontconfig=/usr/local/fontconfig

#cp /usr/local/png/include/png.h ./

#cp /usr/local/png/include/pngconf.h ./

#make;make install

6、安装PHP

这个地方是最重要的地方，因为默认情况下Nginx和PHP他俩之间是一点感觉没有的。在之前，很多朋友都搭建过Apache+PHP，Apache+PHP编译后生成的是模块文件，而Nginx+PHP需要PHP生成可执行文件才可以，所以要利用fastcg技术来实现N ginx与PHP的整合，这个只要我们安装是启用FastCGI即可。此次我们安装PHP不仅使用了FastCGI，而且还使用了PHP-FPM这么一个东东，PHP-FPM说白了是一个管理FastCGI的一个管理器，它作为PHP的插件纯在，在安装PHP要想使用PHP-FPM时就需要把PHP-FPM以补丁的形式安装到PHP中，而且PHP要与PHP-FPM版本一致，这是必须的，切记!

首先我们把PHP和PHP-FPM下载到同一目录下，此次用的为php-5.3.0.tar.bz2和php-5.3.0-fpm-0.5.12.diff.gz，下载到了同一目录下

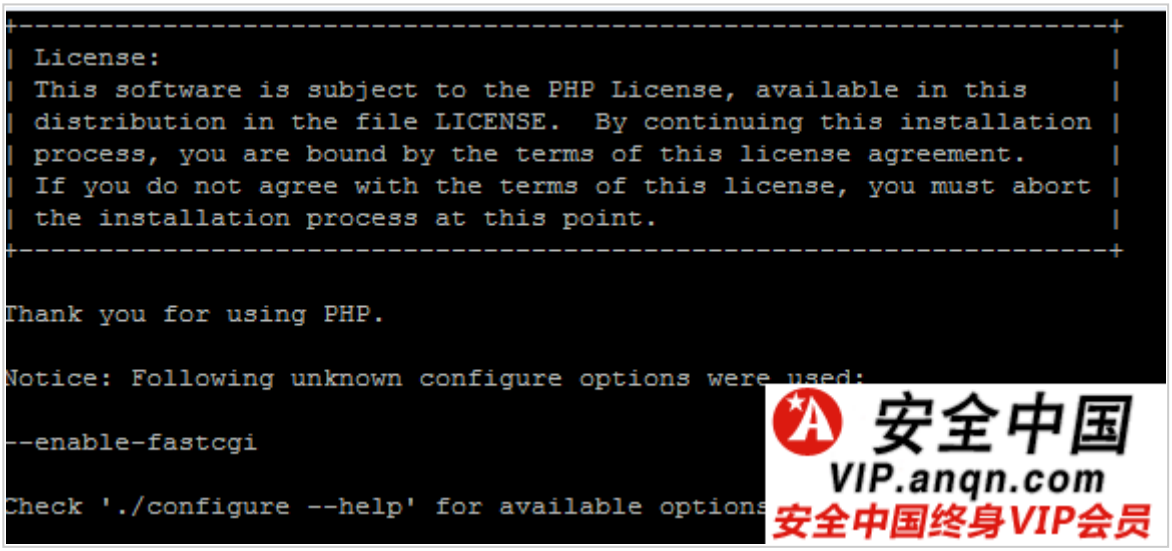
#tar xvf php-5.3.0.tar.bz2

#gzip -cd php-5.3.0-fpm-0.5.12.diff.gz | patch -d php-5.3.0 -p1 //将php-5.3.0-fpm-0.5.12.diff.gz以补丁形式加到php-5.3.0里面

#cd php-5.3.0

#./configure --prefix=/usr/local/php --with-gd=/usr/local/gd --with-jpeg-dir=/usr/local/jpeg --with-png-dir=/usr/local/png --with-freetype-dir=/usr/local/freetype --with-mysql=/usr/local/mysql --enable-fastcgi --enable-fpm

注：Nginx+PHP整合，在安装时必须启用--enable-fastcg和 --enable-fpm，这两个选项是做什么的上面已经描述。执行完后系统会提示--enable-fastcgi是一个未知选项，我们不必理会



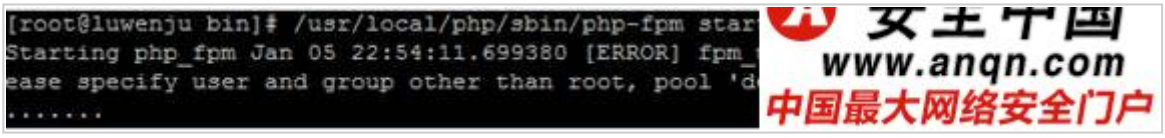
#make

#make install

#cp php.ini-dist /usr/local/php/etc/php.ini

下面我们就要启动PHP-FPM

#/usr/local/php/sbin/php-fpm start



在启动PHP-FPM时会报上面这个错误，原因是PHP-FPM自己不知道以那个用户和组运行PHP，所以我们要修改一个文件，把文件中的注释去掉即可（打开文件把红色部分删除），然后PHP-FPM会以nobody用户和组去运行PHP。

#vi /usr/local/php/etc/php-fpm.conf

#/usr/local/php/sbin/php-fpm start

#ps -aux | grep php

#echo "/usr/local/php/sbin/php-fpm start" >>/etc/rc.local

五、整合Nginx与PHP

上面已经讲过，Nginx自己并不处理动态网页的请求，而且Nginx将得到的动态请求转交给PHP，下面我们打开Nginx的配置文件看一下

#vi /usr/local/nginx/conf/nginx.conf　　//标的部分是我们后面要修改的

看上图，Nginx已经知道怎么把得到的请求传达给PHP，Nginx在得到*.php请求时，会把请求通过9000端口传给PHP。下面我们把这些注释给去掉即可，如下图

注：上面的/usr/local/nginx/html 是我们PHP网站放置的路径

那么只有Nginx自己知道咋找PHP了还不行，还需要PHP知道咋找Nginx，PS：你见过大街上的JJMM约会时有不是相互认识对方，或者是不知道用啥方法和对方接头的？这点我们不需要担心，PHP-FPM已经在配置文件中定义了从哪接受PHP请求，我们可以打开配置文件看一下

#vi /usr/local/php/etc/php-fpm.conf

如上图所示，我们在前面已经看到过Nginx是通过本机的9000端口将PHP请求转发给PHP的，而上图我们可以看到PHP自己是从本机的9000端口侦听数据 ， Nginx与PHP通过本机的9000端口完成了数据请求。

六、测试

我们在nginx的配置文件里面已经定义了PHP网站的存放路径，路径问/usr/local/nginx/html

下面我们在这个目录下新建一个PHP页面测试网页，文件名为test.php，内容如下

重启PHP与nginx后(可以用杀死进程的方式关闭，然后在启动)我们在浏览器中输入http://localhost/test.php，出现如下界面算成功