配置163yum源

备份

# cp /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bak

修改

# vi /etc/yum.repos.d/CentOS-Base.repo

# CentOS-Base.repo

#

# The mirror system uses the connecting IP address of the client and the

# update status of each mirror to pick mirrors that are updated to and

# geographically close to the client. You should use this for CentOS updates

# unless you are manually picking other mirrors.

#

# If the mirrorlist= does not work for you, as a fall back you can try the

# remarked out baseurl= line instead.

#

#

[base]

name=CentOS-$releasever - Base

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os

baseurl=http://mirrors.163.com/centos/$releasever/os/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#released updates

[updates]

name=CentOS-$releasever - Updates

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates

baseurl=http://mirrors.163.com/centos/$releasever/updates/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#additional packages that may be useful

[extras]

name=CentOS-$releasever - Extras

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras

baseurl=http://mirrors.163.com/centos/$releasever/extras/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#additional packages that extend functionality of existing packages

[centosplus]

name=CentOS-$releasever - Plus

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus

baseurl=http://mirrors.163.com/centos/$releasever/centosplus/$basearch/

gpgcheck=1

enabled=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#contrib - packages by Centos Users

[contrib]

name=CentOS-$releasever - Contrib

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=contrib

baseurl=http://mirrors.163.com/centos/$releasever/contrib/$basearch/

gpgcheck=1

enabled=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

# yum clean all

# yum makecache #将服务器上的软件包信息缓存到本地,以提高搜索安装软件的速度

# yum install vim\* #测试域名是否可用

下面开始安装php

1. 安装 开发软件包：yum -y groupinstall "Development Tools"
2. 安装mysql : yum -y install mysql mysql-server mysql-devel
3. 安装组件:yum install libxml2 libxml2-devel libcrul libcurl-devel gd gd-devel libpng libpng-devel

需要安装libmcrypt

用第三方yum源安装或者用源码包安装，推荐用第三方源

wget <http://www.atomicorp.com/installers/atomic> && ./atomic

yum install php-mcrypt libmcrypt libmcrypt-devel

上面安装之后，然后安装php

# cd /usr/local/src/

# wget <http://cn2.php.net/distributions/php-5.6.2.tar.gz>

# tar -zxvf php-5.6.2.tar.gz

# cd php-5.6.2

# ./configure \

--prefix=/usr/local/php \

--with-config-file-path=/usr/local/php/etc \

--enable-fpm \

--with-fpm-user=php-fpm \

--with-fpm-group=pho-fpm \

--with-mysql=mysqlnd \

--with-mysql-sock=/tmp/mysql.sock \

--with-libxml-dir \

--with-gd \

--with-jpeg-dir \

--with-png-dir \

--with-freetype-dir \

--with-iconv-dir \

--with-zlib-dir \

--with-mcrypt \

--enable-soap \

--enable-gd-native-ttf \

--enable-ftp \

--enable-mbstring \

--enable-exif \

--disable-ipv6 \

--with-pear \

--with-curl \

--with-openssl

出现未安装的错误，直接用yum 进行安装即可 记得不要忘了装上 -devel    (libcurl libpng libcrul12等等)  
出现找不到文件路径的情况下  用find / -name 'name'去查找一下  
出现warning 的情况大多是因为版本已经默认安装了，可以去掉该行

# make

# make install

这里要先声明一下，针对Nginx的php安装和针对apache的php安装是有区别的，因为Nginx中的php是以fastcgi的方式结合nginx的，可以理解为nginx代理了php的fastcgi，而apache是把php作为自己的模块来调用的。

# cd php-5.6.2

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

# cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

# /usr/local/php/sbin/php-fpm –t

# cp /usr/local/src/php-5.6.2/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

# chmod 755 /etc/init.d/php-fpm

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

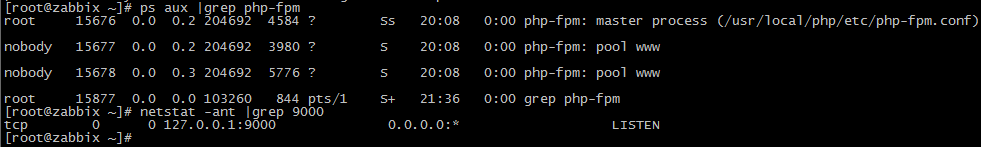
user = nobody

group = nobody

# /etc/init.d/php-fpm start

# ps aux |grep php-fpm

# netstat -ant |grep 9000



开始安装nginx

安装之前先安装pcre pcre-devel openssl openssl-devel

# yum install –y pcre pcre-devel openssl openssl-devel

# wget <http://nginx.org/download/nginx-1.6.2.tar.gz> [\\安装稳定版，最新版是1.9.9](file:///\\\\安装稳定版，最新版是1.9.9)

# wget <http://nginx.org/download/nginx-1.9.9.tar.gz>

# tar zxvf nginx-1.6.2.tar.gz

# cd nginx-1.6.2

# ./configure \

--prefix=/usr/local/nginx \

--with-http\_ssl\_module \

--with-http\_realip\_module \

--with-http\_sub\_module \

--with-http\_gzip\_static\_module \

--with-http\_stub\_status\_module \

--with-pcre \

--with-http\_addition\_module \

--with-http\_dav\_module \

--without-http\_rewrite\_module \

--add-module=/usr/local/src/nginx\_upstream\_hash-/ \

--add-module=/usr/local/src/gnosek-nginx-upstream-fair-/ \

--add-module=/path/to/nginx-rtmp-module/ \

# make

# make install

添加一个nginx主程序的符号链接

# ln -sf /usr/local/nginx/sbin/nginx /usr/sbin

# nginx –t //显示test is successful表示成功

添加openssl证书 https

#cd /usr/local/nginx/conf

#openssl genrsa -des3 -out server.key 1024

#openssl req -new -key server.key -out server.csr

#openssl rsa -in server.key -out server\_nopwd.key

#openssl x509 -req -days 365 -in server.csr -signkey server\_nopwd.key -out server.crt

首先把原来的配置文件清空:

> /usr/local/nginx/conf/nginx.conf

# vim /usr/local/nginx/conf/nginx.conf

添加如下内容

user nobody nobody;

worker\_processes 2;

error\_log /usr/local/nginx/logs/nginx\_error.log crit;

pid /usr/local/nginx/logs/nginx.pid;

worker\_rlimit\_nofile 51200;

events

{

use epoll;

worker\_connections 6000;

}

http

{

include mime.types;

default\_type application/octet-stream;

server\_names\_hash\_bucket\_size 3526;

server\_names\_hash\_max\_size 4096;

log\_format combined\_realip '$remote\_addr $http\_x\_forwarded\_for [$time\_local]'

'$host "$request\_uri" $status'

'"$http\_referer" "$http\_user\_agent"';

sendfile on;

tcp\_nopush on;

keepalive\_timeout 30;

client\_header\_timeout 3m;

client\_body\_timeout 3m;

send\_timeout 3m;

connection\_pool\_size 256;

client\_header\_buffer\_size 1k;

large\_client\_header\_buffers 8 4k;

request\_pool\_size 4k;

output\_buffers 4 32k;

postpone\_output 1460;

client\_max\_body\_size 10m;

client\_body\_buffer\_size 256k;

client\_body\_temp\_path /usr/local/nginx/client\_body\_temp;

proxy\_temp\_path /usr/local/nginx/proxy\_temp;

fastcgi\_temp\_path /usr/local/nginx/fastcgi\_temp;

fastcgi\_intercept\_errors on;

tcp\_nodelay on;

gzip on;

gzip\_min\_length 1k;

gzip\_buffers 4 8k;

gzip\_comp\_level 5;

gzip\_http\_version 1.1;

gzip\_types text/plain application/x-javascript text/css text/htm application/xml;

server

{

listen 80;

server\_name localhost;

index index.html index.htm index.php;

root /usr/local/nginx/html;

location ~ \.php$ {

include fastcgi\_params;

fastcgi\_pass 127.0.0.1：9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /usr/local/nginx/html$fastcgi\_script\_name;

}

server {

listen 443;

ssl on;

ssl\_certificate /usr/local/nginx/conf/server.crt;

ssl\_certificate\_key /usr/local/nginx/conf/server\_nopwd.key;

}

}

}

# nginx //启动nginx

创建测试文件:

vim /usr/local/nginx/html/test.php

内容如下:

<?php

    Phpinfo();

?>

测试:

 curl localhost/test.php