**一、mysql安装**

cd /usr/local/src

1、下载二进制包

wget http://mirrors.sohu.com/mysql/MySQL-5.6/mysql-5.6.35-linux-glibc2.5-x86\_64.tar.gz

2、解压mysql安装包

tar -zxvf mysql-5.6.35-linux-glibc2.5-x86\_64.tar.gz

3、将解压的目录重命名

mv mysql-5.6.35-linux-glibc2.5-x86\_64 /usr/local/mysql

4、创建用户

useradd mysql

5、创建目录

6、mkdir -p /data/mysql //创建datadir，数据库文件会放到这里在

7、cd /usr/local/mysql //在该目录下初始化数据库

初始化报错：



解决办法：yum -y install autoconf



解决办法： yum install libaio-devel

8、初始化数据库

./scripts/mysql\_install\_db --user=mysql --datadir=/data/mysql

9、编辑配置文件：

vim /etc/my.cnf //mysql启动时默认会去/etc/my.cnf找配置文件。

[mysqld]

datadir=/data/mysql

socket=/tmp/mysql.sock

10、拷贝启动文件

cp support-files/mysql.server /etc/init.d/mysqld

11、编辑启动文件

vim /etc/init.d/mysql

basedir=/usr/local/mysql

datadir=/data/mysql

12、chkconfig --add mysqld // 将mysql服务加入系统服务

13、chkconfig mysqld on // 设置mysql服务开机启动

14启动mysql

/etc/init.d/mysqld

**二、php安装**

1、下载PHP5.6.30安装包

wget http://cn2.php.net/distributions/php-5.6.30.tar.gz

2、解压安装包

tar -zxvf php-5.6.30.tar.gz

3、创建用户

useradd -s /sbin/nologin php-fpm

3、编译安装

./configure --prefix=/usr/local/php-fpm --with-config-file-path=/usr/local/php-fpm/etc --enable-fpm --with-fpm-user=php-fpm --with-fpm-group=php-fpm --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-pdo-mysql=/usr/local/mysql --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 --with-pear --with-curl --with-openssl

make && make install

4、拷贝配置文件

cd /usr/local/src/php-5.6.30

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

5、cd /usr/local/php-fpm/etc/

vim php-fpm.conf

[global]

pid = /usr/local/php-fpm/var/run/php-fpm.pid

error\_log = /usr/local/php-fpm/var/log/php-fpm.log

[www]

listen = /tmp/php-fcgi.sock //两种监听方式，任意选择一种

#listen = 127.0.0.1：9000

listen.mode = 666 //当监听方式是sock时这里生效，说明sock那个文件的权限是666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

6、拷贝启动脚本

cd /usr/local/src/php-5.6.30

cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

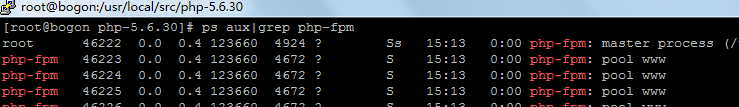
7、chmod 755 /etc/init.d/php-fpm

chkconfig --add php-fpm

chkconfig php-fpm on

8、/usr/local/php-fpm/sbin/php-fpm -t //检测配置文件是否有错

9、service php-fpm start //启动服务



**三、 Nginx安装**

cd /usr/local/src

1下载安装包

wget http://nginx.org/download/nginx-1.12.1.tar.gz

2、tar -zxvf nginx-1.12.1.tar.gz //解压安装包

3、编译安装

cd /usr/local/src/nginx-1.12.1

./configure --prefix=/usr/local/nginx

make && make install

4、/usr/local/nginx/sbin/nginx -t //测试配置文件

5、vim /etc/init.d/nginx //编辑启动脚本

#!/bin/bash

# chkconfig: - 30 21

# description: http service.

# Source Function Library

. /etc/init.d/functions

# Nginx Settings

NGINX\_SBIN="/usr/local/nginx/sbin/nginx"

NGINX\_CONF="/usr/local/nginx/conf/nginx.conf"

NGINX\_PID="/usr/local/nginx/logs/nginx.pid"

RETVAL=0

prog="Nginx"

start()

{

echo -n $"Starting $prog: "

mkdir -p /dev/shm/nginx\_temp

daemon $NGINX\_SBIN -c $NGINX\_CONF

RETVAL=$?

echo

return $RETVAL

}

stop()

{

echo -n $"Stopping $prog: "

killproc -p $NGINX\_PID $NGINX\_SBIN -TERM

rm -rf /dev/shm/nginx\_temp

RETVAL=$?

echo

return $RETVAL

}

reload()

{

echo -n $"Reloading $prog: "

killproc -p $NGINX\_PID $NGINX\_SBIN -HUP

RETVAL=$?

echo

return $RETVAL

}

restart()

{

stop

start

}

configtest()

{

$NGINX\_SBIN -c $NGINX\_CONF -t

return 0

}

case "$1" in

start)

start

;;

stop)

stop

;;

reload)

reload

;;

restart)

restart

;;

configtest)

configtest

;;

\*)

echo $"Usage: $0 {start|stop|reload|restart|configtest}"

RETVAL=1

esac

exit $RETVAL

6、设置权限

chmod 755 /etc/init.d/nginx

chkconfig --add nginx

chkconfig nginx on

7、vim /usr/local/nginx/conf/nginx.conf//编辑配置文件

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 //server开始就是第一个虚拟主机，后面可跟很多个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 unix:/tmp/php-fcgi.sock; //监听方式要与php-fpm.conf里面一致

#fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

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

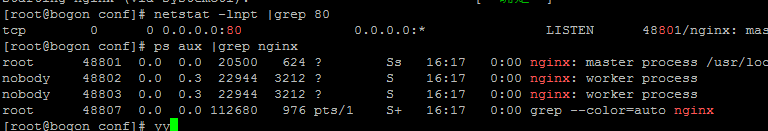
}

}

}

8、/usr/local/nginx/sbin/nginx -t //检查配置文件

9、/etc/init.d/nginx start //启动



# 10、测试nginx，在浏览器上输入服务器IP会显示“Welcome to nginx!”

# 默认是访问/usr/local/nginx/html/这个目录里面的东西

# 11、测试PHP，vim /usr/local/nginx/html/1.php

# <?php

# echo "some one like you";

# ?>



**四、Nginx默认虚拟主机**

1、vim /usr/local/nginx/conf/nginx.conf //编辑配置文件

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;

include vhost/\*.conf;

｝

2、cd /usr/local/nginx/conf

mkdir vhost

cd /usr/local/nginx/conf/vhost

3、vim aaa.com.conf

server

{

listen 80 default\_server; // 有这个标记的就是默认虚拟主机

server\_name aaa.com;

index index.html index.htm index.php;

root /data/wwwroot/default;

}

4、mkdir -p /data/wwwroot/default

cd /data/wwwroot/default

vim index.html //加入下面内容



5、重新加载服务

/usr/local/nginx/sbin/nginx -t

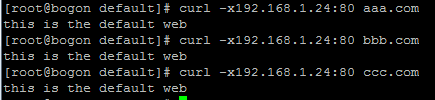
/usr/local/nginx/sbin/nginx -s reload

6、测试

[root@bogon default]# curl 192.168.1.24

this is the default web

不管是什么域名，解析后，都是访问的这个默认站点



**五、 Nginx用户认证**

1、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

location /

{

auth\_basic "Auth";

auth\_basic\_user\_file /usr/local/nginx/conf/htpasswd;

}

}

2、htpasswd -c /usr/local/nginx/conf/htpasswd jz //创建用户认证的用户名和密码

如果没有htpasswd，yum -y install -y httpd

3、htpasswd /usr/local/nginx/conf/htpasswd JZ //创建第二个用户，不加“-c”

4、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

5、测试，401表示访问需要输入用户名和密码



6、mkdir /data/wwwroot/test.com

vim /data/wwwroot/test.com/index.html

this is test web

7、测试

curl -ujz:123456 -x192.168.1.24:80 test.com



在PC hosts定义 192.168.1.24 test.com

在浏览器上输入test.com 会提示输入用户名和密码。

8、针对“admin”目录做

vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

location /admin/

{

auth\_basic "Auth";

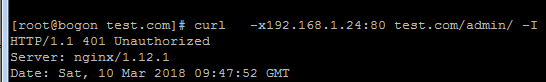
auth\_basic\_user\_file /usr/local/nginx/conf/htpasswd;

}

}

/usr/local/nginx/sbin/nginx -t

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



9、mkdir /data/wwwroot/test.com/admin

vim /data/wwwroot/test.com/admin/index.html //index.html是默认索引页

admin test is ok

10、在浏览器上输入 http://test.com/admin/ ，会让输入用户名和密码



11、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

location ~ admin.php

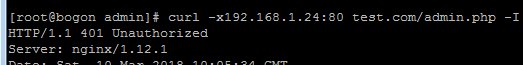
{

auth\_basic "Auth";

auth\_basic\_user\_file /usr/local/nginx/conf/htpasswd;

}

}



**六、Nginx域名重定向**

1、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com test1.com test2.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

if ($host != 'test.com' ) {

rewrite ^/(.\*)$ http://test.com/$1 permanent;

}

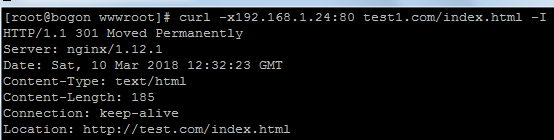
}

2、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

3、测试



**七、Nginx访问日志**

1、vim /usr/local/nginx/conf/nginx.conf //找到定义日志的地方

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

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

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

// combined\_realip：日志的格式的名字，可以自定义，我这里改为JZ

$remote\_addr 客户端IP(公网IP)

$http\_x\_forwarded\_for 代理服务器的IP

$time\_local 服务器本地时间

$host 访问主机名（域名）

$request\_uri 访问的url地址

$status 状态码

$http\_referer referer

$http\_user\_agent user\_agent

2、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com test1.com test2.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

if ($host != 'test.com' ) {

rewrite ^/(.\*)$ http://test.com/$1 permanent;

}

access\_log /tmp/1.log JZ; //就这一行，JZ就是定义的日志格式名字

}

3、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

**八、Nginx日志切割**

1、vim /usr/local/sbin/nginx\_log\_rotate.sh

#!/bin/bash

## 假设nginx的日志存放路径为/data/logs/

d=`date -d "-1 day" +%Y%m%d` //昨天的日期

logdir="/tmp/"

nginx\_pid="/usr/local/nginx/logs/nginx.pid" //程序的pid

cd $logdir

for log in `ls \*.log`

do

mv $log $log-$d

done

/bin/kill -HUP `cat $nginx\_pid` //重新加载程序

2、加任务计划

0 0 \* \* \* /bin/bash /usr/local/sbin/nginx\_log\_rotate.sh

**九、静态文件不记录日志和过期时间**

vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com test1.com test2.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

if ($host != 'test.com' ) {

rewrite ^/(.\*)$ http://test.com/$1 permanent;

}

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$

{

expires 7d; //过期时间

access\_log off; //不记录日志

}

location ~ .\*\.(js|css)$

{

expires 12h;

access\_log off;

}

access\_log /tmp/test.com.log JZ;

}

2、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

**十、Nginx防盗链**

1、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com test1.com test2.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

if ($host != 'test.com' ) {

rewrite ^/(.\*)$ http://test.com/$1 permanent;

}

location ~\* ^.+\.(gif|jpg|png|swf|flv|rar|zip|doc|pdf|gz|bz2|jpeg|bmp|xls)$

{

expires 7d;

valid\_referers none blocked server\_names \*.test.com ; //白名单

if ($invalid\_referer) {

return 403;

}

access\_log off;

}

location ~ .\*\.(js|css)$

{

expires 12h;

access\_log off;

}

access\_log /tmp/test.com.log JZ;

}

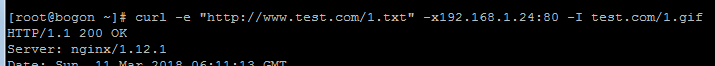
2、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

3、测试





**十一、Nginx访问控制**

1、vim /usr/local/nginx/conf/vhost/test.com.conf

server

{

listen 80;

server\_name test.com test1.com test2.com;

index index.html index.htm index.php;

root /data/wwwroot/test.com;

if ($host != 'test.com' ) {

rewrite ^/(.\*)$ http://test.com/$1 permanent;

}

location ~\* ^.+\.(gif|jpg|png|swf|flv|rar|zip|doc|pdf|gz|bz2|jpeg|bmp|xls)$

{

expires 7d;

valid\_referers none blocked \*.test.com;

if ($invalid\_referer) {

return 403;

}

access\_log off;

}

location ~ .\*\.(js|css)$

{

expires 12h;

access\_log off;

}

location /admin/ //允许指定的IP访问该目录

{

allow 192.168.1.1; //上面的匹配了，下面就不管了

allow 127.0.0.1;

deny all;

}

access\_log /tmp/test.com.log JZ;

}

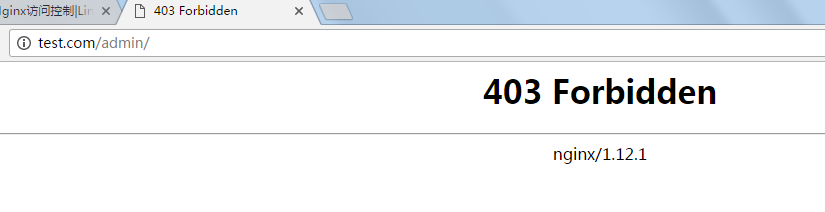
2、重新加载服务

/usr/local/nginx/sbin/nginx -t

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

3、测试

在浏览器上输入 test.com/admin/ 返回的403，我pc的IP：192.168.1.24







4、vim /usr/local/nginx/conf/vhost/test.com.conf //匹配正则

location ~ .\*\.(js|css)$

{

expires 12h;

access\_log off;

}

location /

{

allow 192.168.1.1;

allow 127.0.0.1;

deny all;

}

location ~ .\*(upload|image)/.\*\.php$ //匹配这些直接deny

{

deny all;

}

/usr/local/nginx/sbin/nginx -t

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

mkdir /data/wwwroot/upload

vim /data/wwwroot/upload/1.php

hello





5、根据user\_agent限制

vim /usr/local/nginx/conf/vhost/test.com.conf

location /

{

allow 192.168.1.1;

allow 127.0.0.1;

deny all;

}

location ~ .\*(upload|image)/.\*\.php$

{

deny all;

}

if ($http\_user\_agent ~\* 'Spider/3.0|YoudaoBot|Tomato') //不区分大小写

{

return 403;

}

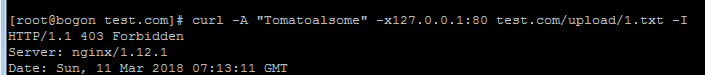
/usr/local/nginx/sbin/nginx -t

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

测试



curl -A 指定user\_agent





**十二、Nginx解析php相关配置**

1、vim /usr/local/nginx/conf/vhost/test.com.conf

if ($http\_user\_agent ~\* 'Spider/3.0|YoudaoBot|Tomato')

{

return 403;

}

location ~ \.php$

{

include fastcgi\_params;

fastcgi\_pass unix:/tmp/php-fcgi.sock;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /data/wwwroot/test.com$fastcgi\_script\_name;

}

监听方式要和/usr/local/php-fpm/etc/php-fpm.conf里面的一样

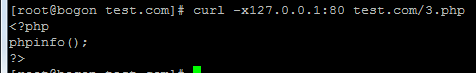
2、加载配置文件前，测试能否解析PHP

vim /data/wwwroot/test/com/3.php

<?php

phpinfo();

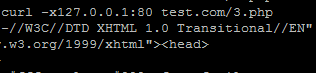
?>



/usr/local/nginx/sbin/nginx -t

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

已经解析了PHP



**十三、Nginx代理**

1、cd /usr/local/nginx/conf/vhost

vim proxy.conf //加入如下内容

server

{

listen 80;

server\_name ask.apelearn.com;

location /

{

proxy\_pass http:// 47.91.145.78/;

proxy\_set\_header Host $host;

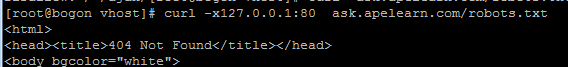
proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

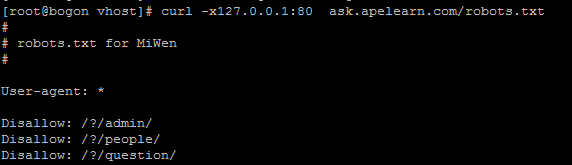
}

加载配置文件前



/usr/local/nginx/sbin/nginx -t

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



**十四、Nginx负载均衡**

1、vim /usr/local/nginx/conf/vhost/load.conf

upstream qq\_com //自定义的名字

{

ip\_hash; //始终访问同一设备

server 61.135.157.156:80;

server 125.39.240.113:80;

}

server

{

listen 80;

server\_name www.qq.com;

location /

{

proxy\_pass http://qq\_com; //和上面自定义 的名字对应

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

curl -x127.0.0.1:80 www.qq.com //加载配置文件前，访问的是默认虚拟主机

/usr/local/nginx/sbin/nginx -t

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

curl -x127.0.0.1:80 www.qq.com //访问的是www.qq.com

**十五、生产ssl密钥对**

1、cd /usr /local/nginx/conf

openssl genrsa -des3 -out tmp.key 2048 //生成私钥，需要设置密码

2、取消密码

openssl rsa -in tmp.key -out aminglinux.key //这里需要输入上面设置的密码

3、rm -f tmp.key //删除带密码的私钥，保留没有密码的私钥aminglinux.key

4、生成证书请求文件，需要拿这个文件和私钥一起生产公钥文件

openssl req -new -key aminglinux.key -out aminglinux.csr

5、生成公钥aminglinux.crt

openssl x509 -req -days 365 -in aminglinux.csr -signkey aminglinux.key -out aminglinux.crt

**十六、Nginx配置ssl**

1、cd /usr/local/src/nginx-1.12.1 //重新编译nginx.之前nginx没有编译ssl

./configure --prefix=/usr/local/nginx --with-http\_ssl\_module

make && make install

2、vim /usr/local/nginx/conf/vhost/ssl.conf

server

{

listen 443; //监听端口

server\_name aming.com;

index index.html index.php;

root /data/wwwroot/aming.com; //网站根目录

ssl on;

ssl\_certificate aminglinux.crt; //公钥

ssl\_certificate\_key aminglinux.key; //私钥

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;

}

/usr/local/nginx/sbin/nginx -t

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

/etc/init.d/nginx restart

3、mkdir /data/wwwroot/aming.com

vim index.html

test ssl

4、cp /usr/local/nginx/conf/aminglinux.crt /etc/pki/ca-trust/source/anchors/

5、/bin/update-ca-trust

curl https://adming.com //正常显示 test ssl

4、在PC的hosts上定义好192.168.1.24 aming.com，正常可以去浏览器上访问https://aming.com

**十七、php-fpm的pool**

1、vim /usr/local/php-fpm/etc/php-fpm.conf //增加一个池子

[JZ]

listen = /tmp/JZ.sock

#listen = 127.0.0.1:9000

listen.mode = 666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

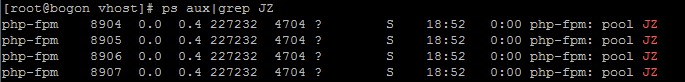
pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

2、/usr/local/php-fpm/sbin/php-fpm -t //测试是否有语法错误

3、/etc/init.d/php-fpm reload //重新加载服务



3、不同的站点使用不同的池子

cd /usr/local/nginx/conf/vhost

vim aaa.com.conf

server

{

listen 80 default\_server;

server\_name aaa.com;

index index.html index.htm index.php;

root /data/wwwroot/default;

location ~ \.php$

{

include fastcgi\_params;

fastcgi\_pass unix:/tmp/JZ.sock; //根据sock来区分

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /data/wwwroot/default.com$fastcgi\_script\_name;

}

}

vim test.com.conf

location ~ \.php$

{

include fastcgi\_params;

fastcgi\_pass unix:/tmp/php-fcgi.sock;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /data/wwwroot/test.com$fastcgi\_script\_name;

}

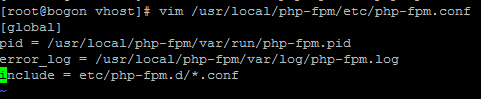
4、php是支持vhost的

vim /usr/local/php-fpm/etc/php-fpm.conf//添加，之前配置的池子（www,和JZ都删掉）

pid = /usr/local/php-fpm/var/run/php-fpm.pid

error\_log = /usr/local/php-fpm/var/log/php-fpm.log

include = etc/php-fpm.d/\*.conf



5、mkdir /usr/local/php-fpm/etc/php-fpm.d //在这个目录下创建

vim www.conf

[www]

listen = /tmp/www.sock

listen.mode=666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

vim JZ.conf

[JZ]

listen = /tmp/JZ.sock

listen.mode=666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

重启服务

/usr/local/php-fpm/sbin/php-fpm -t

/etc/init.d/php-fpm restart

**十八、php-fpm慢执行日志**

1、vim /usr/local/php-fpm/etc/php-fpm.d/www.conf

[www]

listen = /tmp/www.sock

listen.mode=666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

request\_slowlog\_timeout = 1

slowlog = /usr/local/php-fpm/var/log/www-slow.log

2、加载服务

/usr/local/php-fpm/sbin/php-fpm -t

/etc/init.d/php-fpm reload

3、vim /data/wwwroot/test.com/sleep.php //慢日志测试

<?php

echo "test slow log";

sleep(2);

echo "done";

?>

curl -x127.0.0.1:80 test.com/sleep.php

4、cat /usr/local/php-fpm/var/log/www-slow.log //查看慢日志里面的内容



**十九、open\_basedir（限制php在指定的目录里活动）**

1、vim /usr/local/php-fpm/etc/php-fpm.d/www.conf

[www]

listen = /tmp/www.sock

listen.mode=666

user = php-fpm

group = php-fpm

pm = dynamic

pm.max\_children = 50

pm.start\_servers = 20

pm.min\_spare\_servers = 5

pm.max\_spare\_servers = 35

pm.max\_requests = 500

rlimit\_files = 1024

request\_slowlog\_timeout = 1

slowlog = /usr/local/php-fpm/var/log/www-slow.log

php\_admin\_value[open\_basedir]=/data/wwwroot/test.com:/tmp/



2、vim /usr/local/php-fpm/php.ini



3、定义错误日志存放路径



4、定义错误日志级别

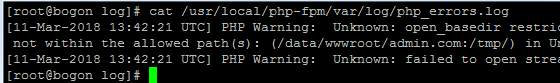


5、创建定义的日志存放路径

touch /usr/local/php-fpm/var/log/php\_errors.log

chmod 777 /usr/local/php-fpm/var/log/php\_errors.log

6、模拟错误，可以查看到错误日志里面有内容



**二十、 php-fpm进程管理**

1、pm = dynamic //动态进程管理，也可以是static（如果是static,只有第2点生效，下面都无效）

2、pm.max\_children = 50 //最大子进程数，ps aux可以查看 ,

3、pm.start\_servers = 20 //启动服务时会启动的进程数

4、 pm.min\_spare\_servers = 5 //定义在空闲时段，子进程数的最少数量，如果达到这个数值时，php-fpm服务会自动派生新的子进程。

5、 pm.max\_spare\_servers = 35 //定义在空闲时段，子进程数的最大值，如果高于这个数值就开始清理空闲的子进程。

6、 pm.max\_requests = 500 //定义一个子进程最多处理的请求数，也就是说在一个php-fpm的子进程最多可以处理这么多请求，当达到这个数值时，它会自动退出。