**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/mysqld

basedir=/usr/local/mysql

datadir=/data/mysql

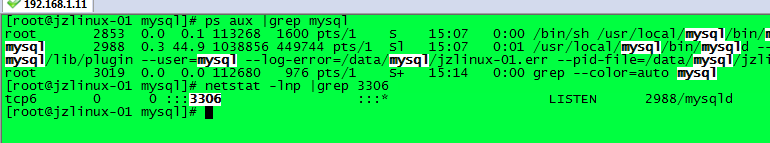
12、chkconfig --add mysqld // 将mysql服务加入系统服务

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

14启动mysql

/etc/init.d/mysqld start

15、启动后查看进程和端口



16、指定配置文件来启动mysql，及mysql配置文件没有在/etc/my.cnf里面

/usr/local/mysql/bin/mysqld\_safe --defaults-file=/etc/my.cnf --user=mysql --datadir=/data/mysql &

--defaults-file=指定mysql配置文件存放的路径

--user=mysql=以哪个用户启动

& //将程序后台启动

17、yum install psmisc //killall命令安装

18、killall比kill -9 杀进程安全些,如遇进程杀不死，有可能还在读写数据，切记不可反复杀，更不能用kill -9强制杀。

**二、MariaDB安装**

1、下载安装包

wget https://downloads.mariadb.com/MariaDB/mariadb-10.2.6/bintar-linux-glibc\_214-x86\_64/mariadb-10.2.6-linux-glibc\_214-x86\_64.tar.gz

2、解压安装包

tar zxvf mariadb-10.2.6-linux-glibc\_214-x86\_64.tar.gz

3、mv mariadb-10.2.6-linux-glibc\_214-x86\_64 /usr/local/mariadb

4、初始化数据库

cd /usr/local/mariadb

./scripts/mysql\_install\_db --user=mysql --basedir=/usr/local/mariadb/ --datadir=/data/mariadb

5、拷贝配置文件

cp support-files/my-small.cnf /usr/local/mariadb/my.cnf

6、拷贝启动脚本

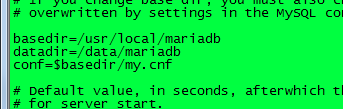
cp support-files/mysql.server /etc/init.d/mariadb

7、vim /etc/init.d/mariadb.cnf //编辑启动脚本，编辑下面内容

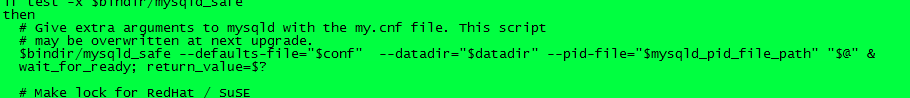
basedir=/usr/local/mariadb

datadir=/data/mariadb

conf=$basedir/my.cnf //因为虚拟机上安装了mysql，为了二者不冲突所以这应改，如果只安装mariadb,都不用动



$bindir/mysqld\_safe **--defaults-file="$conf"** --datadir="$datadir" --pid-file="$mysqld\_pid\_file\_path" "$@" & //因为虚拟机上安装了mysql，为了二者不冲突所以这应改



8、vim /usr/local/mariadb/my.cnf //编辑配置文件定义

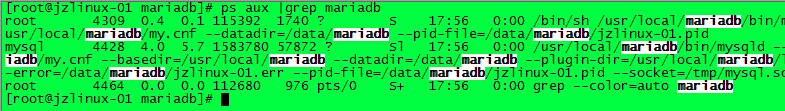


9、启动服务

/etc/init.d/mariadb start

10、查看启动的进程和端口（端口3306）





**三、 Apache安装**

备注：安装apache2.4,需要安装apr和apr-util，他们是一个通用的函数库，它让httpd可以不关心底层的操作系统平台，可以很方便地移植（从linux移植到windows）

1、下载apache2.4

wget https://mirrors.cnnic.cn/apache/httpd/httpd-2.4.29.tar.gz

2、下载apr

wget http://mirrors.hust.edu.cn/apache/apr/apr-1.6.3.tar.gz

3、下载apr-util

wget http://mirrors.hust.edu.cn/apache/apr/apr-util-1.6.1.tar.gz

4、解压安装包

tar -zxvf httpd-2.4.29.tar.gz

tar -zxvf apr-1.6.3.tar.gz

tar -zxvf apr-util-1.6.1.tar.gz

5、cd apr-1.6.3 //安装apr

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

make && make install

6、cd /usr/local/src/apr-util-1.6.1 //安装apr-util

./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr

make && make install

7、安装apache及httpd

./configure --prefix=/usr/local/apache2.4 --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util --enable-so --enable-mods-shared=most

make && make install

8、查看加载的模块，二者都可用。

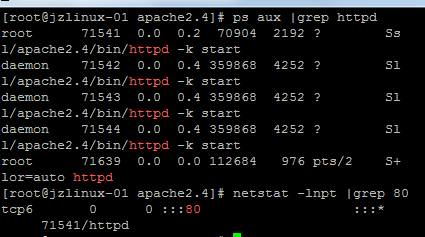
/usr/local/apache2.4/bin/httpd -M

/usr/local/apache2.4/bin/apachectl -M

9、启动服务

/usr/local/apache2.4/bin/apachectl start

10、查看进程和端口情况



**四、PHP5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、编译安装

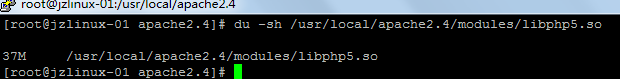
./configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache2.4/bin/apxs --with-config-file-path=/usr/local/php/etc --with-mysql=/usr/local/mysql --with-pdo-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-libxml-dir --with-gd --with-jpeg-dir --with-png-dir --with-freetype-dir --with-iconv-dir --with-zlib-dir --with-bz2 --with-openssl --with-mcrypt --enable-soap --enable-gd-native-ttf --enable-mbstring --enable-sockets --enable-exif

会有报错：通过yum list |grep -i bzip2来判断

yum install -y bzip2-devel //安装这个包解决

make && make install

4、安装完后，会生成相关模块



5、php加载的静态模块

/usr/local/php/bin/php -m

6、/usr/local/apache2.4/bin/apachectl -M //会多这个模块

php5\_module (shared)

7、vim /usr/local/apache2.4/conf/httpd.conf //自动增加了这行



8、拷贝配置文件cd /usr/local/src/php5.6.30

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

**五、php7安装**

1、下载安装包

cd /usr/local/src/

wget http://cn2.php.net/distributions/php-7.1.6.tar.bz2

2、解压安装包

tar -jxf php-7.1.6.tar.bz2

3、编译安装

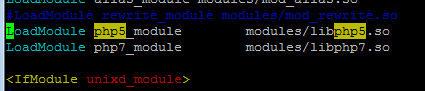
./configure --prefix=/usr/local/php7 --with-apxs2=/usr/local/apache2.4/bin/apxs --with-config-file-path=/usr/local/php7/etc --with-pdo-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql\_config --with-libxml-dir --with-gd --with-jpeg-dir --with-png-dir --with-freetype-dir --with-iconv-dir --with-zlib-dir --with-bz2 --with-openssl --with-mcrypt --enable-soap --enable-gd-native-ttf --enable-mbstring --enable-sockets --enable-exif

make && make install

4、会生成这个模块



vim /usr/local/apache2.4/conf/httpd.conf //自动生成一行，只能选择用其中一个版本的php，那么另外一个必须用“#”注释掉



**六、 Apache和PHP结合**

1、vim /usr/local/apache2.4/conf/httpd.conf //编辑配置文件

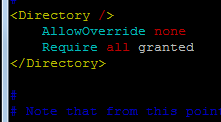
ServerName www.example.com:80 //将这行前面的“#”去掉

<Directory />

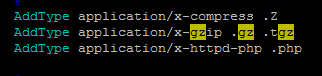
AllowOverride none

Require all denied // 将denied改为granted

</Directory>



AddType application/x-httpd-php .php //配置文件中添加



index.php //在配置文件中添加



iptables -I INPUT -p tcp --dport 80 -j ACCEPT //开放80端口

2、测试配置文件语法是否有错误

/usr/local/apache2.4/bin/apachectl -t

3、重新加载配置文件，不涉及程序重启

/usr/local/apache2.4/bin/apachectl graceful

4、cd /usr/local/apache2.4/htdocs/ //默认网页是访问这里

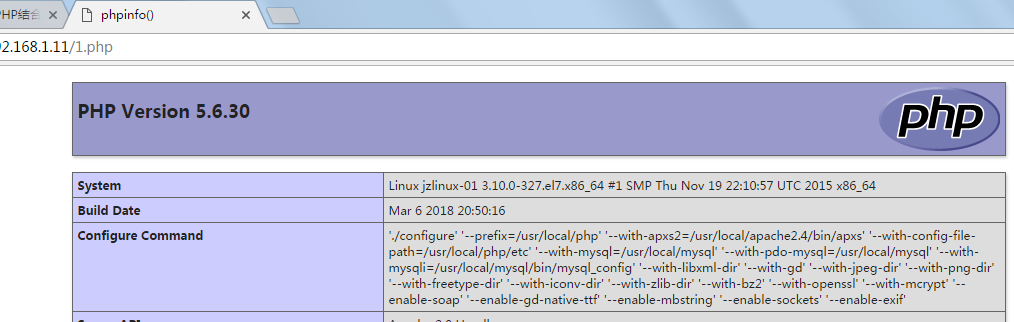
vim 1.php

<?php

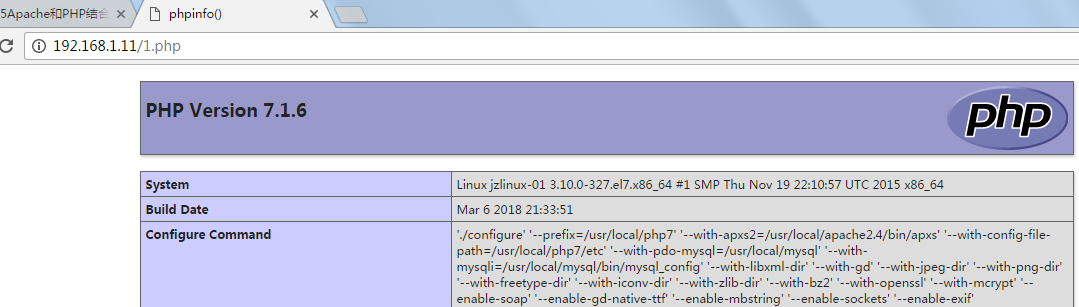
phpinfo();

?>

在浏览器上 192.168.1.11/1.php (关闭防火墙)



PHP7也能加载成功



**七、Apache默认虚拟主机**

1、vim /usr/local/apache2.4/conf/httpd.conf //打开虚拟主机配置文件



2、编辑虚拟主机配置文件，添加2个web程序

vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/abc.com" //网站根目录，及网站程序存放的目录

ServerName abc.com //网站域名

ServerAlias www.abc.com www.123.com

ErrorLog "logs/abc.com-error\_log" //错误日志

CustomLog "logs/abc.com-access\_log" common //访问日志

</VirtualHost>

上面的一个是默认虚拟主机，任何的域名解析到该IP上，默认就是访问上面这个站点的。

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

ErrorLog "logs/111.com-error\_log"

CustomLog "logs/111.com-access\_log" common

</VirtualHost>

3、创建web程序目录

mkdir -p /data/wwwroot/abc.com

mkdir -p /data/wwwroot/111.com

4、在创建的web程序目录下写测试页面

cd /data/wwwroot/abc.com

vim index.php

<?php

echo "anything is possible"

?>

cd /data/wwwroot/111.com

vim index.php

<?php

echo "some one like you"

?>

5、测试配置语法

/usr/local/apache2.4/bin/apachectl -t

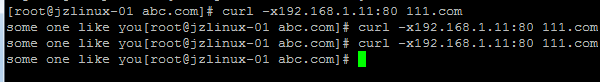
6、重新加载服务

/usr/local/apache2.4/bin/apachectl graceful

7、curl 测试，默认就是访问的第一个虚拟主机



8、如果指定的域名在虚拟主机配置文件中能匹配到的话，就访问指定的域名，如果匹配不到域名，就访问第一个及默认虚拟主机



**八、 Apache用户认证**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf //根据目录做认证

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

<Directory /data/wwwroot/111.com> //指定认证的目录

AllowOverride AuthConfig //这个相当于打开认证的开关

AuthName "111.com user auth" //自定义认证的名字，作用不大

AuthType Basic //认证的类型，一般为Basic，其他类型阿铭没用过

AuthUserFile /data/.htpasswd //指定密码文件所在位置

require valid-user //指定需要认证的用户为全部可用用户

</Directory>

ErrorLog "logs/111.com-error\_log"

CustomLog "logs/111.com-access\_log" common

</VirtualHost>

2、设置密码，"jz"是我配置的用户名

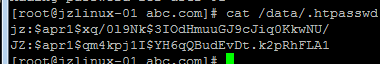
/usr/local/apache2.4/bin/htpasswd -cm /data/.htpasswd jz

如设置成功，会显示下列文件



再次创建用户，不要加"-c"选项

/usr/local/apache2.4/bin/htpasswd -m /data/.htpasswd JZ



3、查看配置文件是否有错误，然后重新加载配置文件

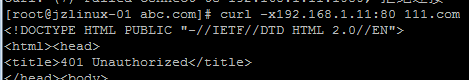
/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

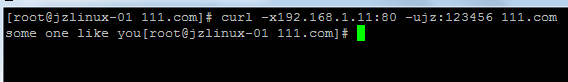
4、测试

在浏览器上输入111.com,会让你输入用户名和密码，将IP 111.com加入到hosts中去

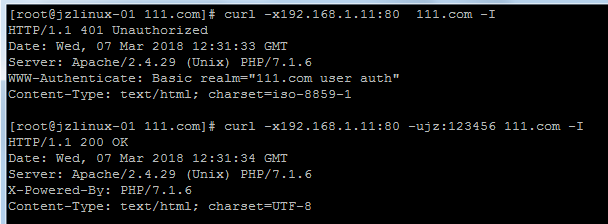
curl -x192.168.1.11:80 111.com //状态码为401，表示需要用户验证



curl -x192.168.1.11:80 -ujz:123456 111.com//正常



curl -x192.168.1.11:80 -ujz:123456 111.com -I



5、1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf //根据文件做认证

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

# <Directory /data/wwwroot/111.com>

<FilesMatch 123.php> //针对111.com/123.php做认证

AllowOverride AuthConfig

AuthName "111.com user auth"

AuthType Basic

AuthUserFile /data/.htpasswd

require valid-user

</FilesMatch>

# </Directory>

ErrorLog "logs/111.com-error\_log"

CustomLog "logs/111.com-access\_log" common

</VirtualHost>

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

cd /data/wwwroot/111.com

vim 123.php

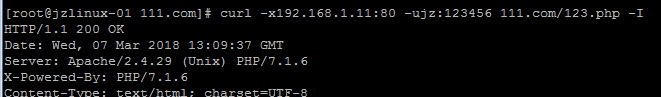
<?php

echo "123.php";

?>

curl -x192.168.1.11:80 111.com/123.php -I //401,





**九、 域名跳转**

1、编辑虚拟主机配文件

vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

# <Directory /data/wwwroot/111.com>

# <FilesMatch 123.php>

# AllowOverride AuthConfig

# AuthName "111.com user auth"

# AuthType Basic

# AuthUserFile /data/.htpasswd

# require valid-user

# </FilesMatch>

# </Directory>

<IfModule mod\_rewrite.c>

RewriteEngine on

RewriteCond %{HTTP\_HOST} !^111.com$

RewriteRule ^/(.\*)$ http://111.com/$1 [R=301,L] //^/:代表除开111.com，（.\*）$:代表111.com后面的如：111.com/123.php;$1代表前面的（.\*）.,123.php代表$1

</IfModule>

ErrorLog "logs/111.com-error\_log"

CustomLog "logs/111.com-access\_log" common

</VirtualHost>

备注：RewriteRule ^/(.\*)/([1-9]+)$ http://111.com/$1/$2 [R=301,L]//（[1-9]+）就是$2

2、编辑主配置文件，打开rewrite模块前的“#”

vim /usr/local/apache2.4/conf/httpd.conf



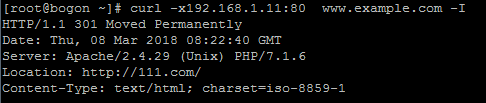
3、加载配置文件

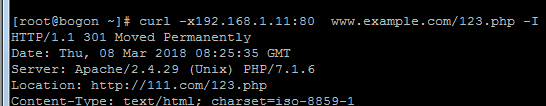
/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

4、测试

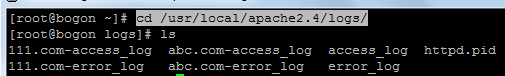
curl -x192.168.1.11:80 www.example.com -I





**十、 Apache访问日志**

1、cd /usr/local/apache2.4/logs/ //默认这里是有日志的



2、编辑主配置文件



3、编辑虚拟主机配置文件

vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf //将common改为combined



/usr/local/apache2.4/bin/apachectl graceful

**十一、访问日志不记录静态文件**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

# <Directory /data/wwwroot/111.com>

# <FilesMatch 123.php>

# AllowOverride AuthConfig

# AuthName "111.com user auth"

# AuthType Basic

# AuthUserFile /data/.htpasswd

# require valid-user

# </FilesMatch>

# </Directory>

<IfModule mod\_rewrite.c>

RewriteEngine on

RewriteCond %{HTTP\_HOST} !^111.com$

RewriteRule ^/(.\*)$ http://111.com/$1 [R=301,L]

</IfModule>

ErrorLog "logs/111.com-error\_log"

ServerAlias 111.com

SetEnvIf Request\_URI ".\*\.gif$" img

SetEnvIf Request\_URI ".\*\.jpg$" img

SetEnvIf Request\_URI ".\*\.png$" img

SetEnvIf Request\_URI ".\*\.bmp$" img

SetEnvIf Request\_URI ".\*\.swf$" img

SetEnvIf Request\_URI ".\*\.js$" img

SetEnvIf Request\_URI ".\*\.css$" img

CustomLog "logs/111.com-access\_log" combined env=!img

</VirtualHost>

2、测试

curl -x192.168.1.11:80 111.com/agc.jpg -I //会记录jpg的日志



3、加载配置文件

/usr/local/apache2.4/bin/apachectl graceful

curl -x192.168.1.11:80 111.com/agc.jpg -I //不会记录jpg的日志

**十二、访问日志切割**

1、编辑虚拟主机配置文件

vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

# <Directory /data/wwwroot/111.com>

# <FilesMatch 123.php>

# AllowOverride AuthConfig

# AuthName "111.com user auth"

# AuthType Basic

# AuthUserFile /data/.htpasswd

# require valid-user

# </FilesMatch>

# </Directory>

<IfModule mod\_rewrite.c>

RewriteEngine on

RewriteCond %{HTTP\_HOST} !^111.com$

RewriteRule ^/(.\*)$ http://111.com/$1 [R=301,L]

</IfModule>

ErrorLog "logs/111.com-error\_log"

ServerAlias 111.com

SetEnvIf Request\_URI ".\*\.gif$" img

SetEnvIf Request\_URI ".\*\.jpg$" img

SetEnvIf Request\_URI ".\*\.png$" img

SetEnvIf Request\_URI ".\*\.bmp$" img

SetEnvIf Request\_URI ".\*\.swf$" img

SetEnvIf Request\_URI ".\*\.js$" img

SetEnvIf Request\_URI ".\*\.css$" img

CustomLog "|/usr/local/apache2.4/bin/rotatelogs -l logs/111.com-access\_%Y%m%d.log 86400" combined env=!img

</VirtualHost>

2、加载配置文件

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

3、访问页面后，自动生成以日期为单位的文件



**十四、 静态元素过期时间**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<VirtualHost \*:80>

DocumentRoot "/data/wwwroot/111.com"

ServerName 111.com

ServerAlias www.example.com

# <Directory /data/wwwroot/111.com>

# <FilesMatch 123.php>

# AllowOverride AuthConfig

# AuthName "111.com user auth"

# AuthType Basic

# AuthUserFile /data/.htpasswd

# require valid-user

# </FilesMatch>

# </Directory>

<IfModule mod\_rewrite.c>

RewriteEngine on

RewriteCond %{HTTP\_HOST} !^111.com$

RewriteRule ^/(.\*)$ http://111.com/$1 [R=301,L]

</IfModule>

<IfModule mod\_expires.c>

ExpiresActive on

ExpiresByType image/gif "access plus 1 days"

ExpiresByType image/jpeg "access plus 24 hours"

ExpiresByType image/png "access plus 24 hours"

ExpiresByType text/css "now plus 2 hour"

ExpiresByType application/x-javascript "now plus 2 hours"

ExpiresByType application/javascript "now plus 2 hours"

ExpiresByType application/x-shockwave-flash "now plus 2 hours"

ExpiresDefault "now plus 0 min"

</IfModule>

ErrorLog "logs/111.com-error\_log"

ServerAlias 111.com

SetEnvIf Request\_URI ".\*\.gif$" img

SetEnvIf Request\_URI ".\*\.jpg$" img

SetEnvIf Request\_URI ".\*\.png$" img

SetEnvIf Request\_URI ".\*\.bmp$" img

SetEnvIf Request\_URI ".\*\.swf$" img

SetEnvIf Request\_URI ".\*\.js$" img

SetEnvIf Request\_URI ".\*\.css$" img

CustomLog "|/usr/local/apache2.4/bin/rotatelogs -l logs/111.com-access\_%Y%m%d.log 86400" combined env=!img

</VirtualHost>

2、vim /usr/local/apache2.4/conf/httpd.conf



3、加载配置文件

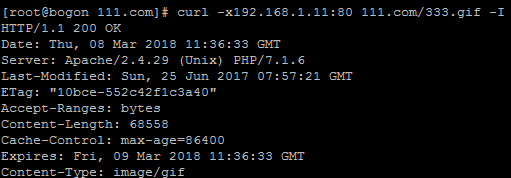
/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

4、测试

浏览器上输入111.com/333.gif,显示max-age=86400





**十五、配置防盗链**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<Directory /data/wwwroot/111.com>

SetEnvIfNoCase Referer "111.com" local\_ref //白名单

SetEnvIfNoCase Referer "aaa.com" local\_ref //白名单

SetEnvIfNoCase Referer "^$" local\_ref //空ref也设置成白名单，要不然直接111.com/123.gif打不开

<filesmatch "\.(txt|doc|mp3|zip|rar|jpg|gif)"> //针对这些文件做的防盗链

Order Allow,Deny

Allow from env=local\_ref //允许白名单

</filesmatch>

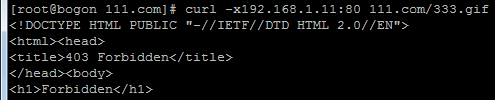
</Directory>

2、加载配置文件

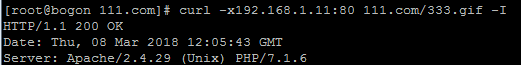
/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

3、测试如果空ref不设置成白名单那么，访问会403



设置后正常



4、curl -e "http://aab.com/1.txt" -x192.168.1.11:80 111.com/333.gif -I //-e "模拟ref"测试





**十六、访问控制目录**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<Directory /data/wwwroot/111.com/admin/>

Order deny,allow

Deny from all

Allow from 127.0.0.1

Allow from 192.168.1.28 //我PC 的IP

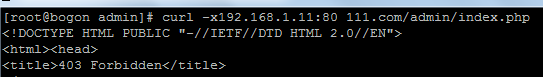
</Directory>

2、加载配置文件

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

3、测试,在/data/wwwroot/111.com/admin/index.php加入点内容





**十七、访问控制FilesMatch**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<Directory /data/wwwroot/111.com>

<FilesMatch "admin.php(.\*)">

Order deny,allow

Deny from all

Allow from 127.0.0.1

Allow from 192.168.1.28

</FilesMatch>

</Directory>

2、加载配置文件

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

3、测试

不满足条件的IP，访问时，/data/wwwroot/111.com/admin.php\*时会403

**十八、限定某个目录禁止解析php**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<Directory /data/wwwroot/111.com/upload>

php\_admin\_flag engine off

</Directory>

2、加载配置文件

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

mkdir /data/wwwroot/111.com/upload

vim 123.php

<?php

echo "123.php";

?>

3、测试，不会解析PHP，会直接显示文件的源码，如果在浏览器上输入111.com/upload/123.php,会直接将源码下载

curl -x192.168.1.11:80 111.com/upload/123.php

<?php

echo "123.php";

?>

4、不想让下载源码

vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

<Directory /data/wwwroot/111.com/upload>

php\_admin\_flag engine off

<FilesMatch (.\*)\.php(.\*)>

Order allow,deny

Deny from all

</FilesMatch>

</Directory>

2、加载配置文件

/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

3、测试，访问直接403，不会显示源码



在浏览器上访问也是403，不会下载源码

**十九、限制user\_agent**

1、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf//下面的东西写到指定虚拟主机里面

<IfModule mod\_rewrite.c>

RewriteEngine on

RewriteCond %{HTTP\_USER\_AGENT} .\*curl.\* [NC,OR]

RewriteCond %{HTTP\_USER\_AGENT} .\*Mozilla/5.0.\* [NC]

RewriteRule .\* - [F]

</IfModule>

2、加载配置文件

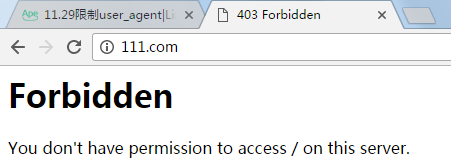
/usr/local/apache2.4/bin/apachectl -t

/usr/local/apache2.4/bin/apachectl graceful

测试



这时候谷歌浏览器访问111.com报403



curl -A "指定浏览器"

curl -A "www.baidu.com/123" -x192.168.1.11:80 111.com -I



**二十、PHP相关配置**

1、拷贝配置文件

cd /usr/local/src/php-7.1.6

cp php.ini-development /usr/local/php7/etc/php.ini

2、加载配置文件

/usr/local/apache2.4/bin/apachectl graceful

3、禁止相关安全函数

vim /usr/local/php7/etc/php.ini

disable\_functions = eval,assert,popen,passthru,escapeshellarg,escapeshellcmd,passthru,exec,system,chroot,scandir,chgrp,chown,escapeshellcmd,escapeshellarg,shell\_exec,proc\_get\_status,ini\_alter,ini\_restore,dl,pfsockopen,openlog,syslog,readlink,symlink,leak,popepassthru,stream\_socket\_server,popen,proc\_open,proc\_close,phpinfo

4、/usr/local/apache2.4/bin/apachectl graceful //加载配置文件

测试

在浏览器上输入 111.com/123.php //回报下面的错误

**Warning**: phpinfo() has been disabled for security reasons in **/data/wwwroot/111.com/123.php** on line **2**

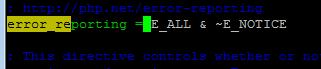
5、在/usr/local/php7/etc/php.ini定义时区

date.timezone = Asia/Chongqing

display\_errors = Off //关闭在浏览器上显示错误信息

log\_errors = On //打开错误日志

//定义错误日志存放位置

 //定义日志级别

6、测试、重新访问111.com/123.php，tmp目录下会生成错误日志信息





7、vim /usr/local/apache2.4/conf/extra/httpd-vhosts.conf

根据不同的主机现在不同的open\_basedir

php\_admin\_value open\_basedir "/data/wwwroot/111.com:/tmp/"

**二十一、PHP扩展模块安装**

1、cd /usr/local/src/

2、wget https://codeload.github.com/phpredis/phpredis/zip/develop //下载，因为自带的没有

3、mv develop phpredis-develop.zip

4、unzip phpredis-develop.zip

5、cd phpredis-develop

6、/usr/local/php7/bin/phpize //生成config文件

报错

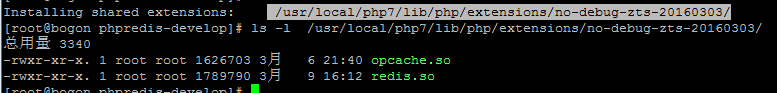


yum install -y autoconf

./configure --with-php-config=/usr/local/php7/bin/php-config

make && make install

7、查看生成的模块



8、查看扩展模块放在哪个目录

/usr/local/php7/bin/php -i |grep -i extension\_dir

9、加载扩展模块

vim /usr/local/php7/etc/php.ini

extension=redis.so

/usr/local/php7/bin/php -m|grep redis



10、安装自带的有，待未安装的模块

cd /usr/local/src/php-7.1.6/ext //进入存放模块的目录

cd zip //安装zip这个扩展模块

/usr/local/php7/bin/phpize

./configure --with-php-config=/usr/local/php7/bin/php-config

make && make install

vim /usr/local/php7/etc/php.ini

extension=zip.so

/usr/local/php7/bin/php -m|grep zip

