#!/bin/bash

echo “It will install lamp or lnmp”

sleep 1

#在这里暂停一秒钟，告诉用户开始安装程序了，暂停一秒是为了让用户看的清楚

check\_ok() {

if [ $? !=0 ]

then

echo “Error, Check the error log.”

exit 1

fi

}

#定义的这个函数是判断上一条命令是否执行成功，如果不成功就及时退出脚本；这个函数将会在下面的脚本中反复调用，其目的是为了在安装过程中及时发现错误及时解决，避免在安装完成之后发现程序安装不全

ar=`arch`

#用于获取当前是64位还是32位操作系统，下面将在下载mysql的脚本中用到此信息

sed –i ‘s/SELINUX=enforcing/ SELINUX=disabled/’ /etc/selinux/comfig

#把selinux通过改配置文件的方式关闭

selinux\_s=`getenforce`

if [ $selinux\_s == “enforcing” ]

then

setenforce 0

fi

#通过改selinux配置文件关闭selinux必须重启才能生效，所以我们可以通过命令暂时关闭，当然，我们这里做了一个判断，当selinux状态是开启的时候我们才会关闭selinux

iptables-save > /etc/sysconfig/iptables\_`data +%s`

iptables –F

service iptables save

#这里是清空iptables规则，但是清空前，我们需要先备份一下，并且在备份文件名中加一个时间戳

myum() {

if ! rpm -qa|grep -q “^$1”

then

yum install -y $1

check\_ok

else

echo $1 already installed.

fi

}

#这里我们把yum源也做成了一个函数，目的是为了检测一下，我们下面要安装的rpm包是否已经安装过，如果安装过就不再安装，并提示“echo $1 already installed.”反则就自动安装，这里我们就引用了上面的check\_ok函数，目的则是为了检测安装rpm包的时候是否安装成功

**注：当B函数引用A函数时，A函数必须放在B函数前面**

for p in gcc wget perl perl-devel libaio libaio-devel pcre-devel zlib-devel

do

myum $p

done

#安装上面的rpm包，并引用myum函数来检测这些包是否已经安装，是否安装成功

if rpm -qa epel-release > /dev/null

then

rpm -e epel-release

fi

if ls /etc/yum.repos.d/epel-6.repo\* > /dev/null 2>&1

then

rm –f /etc/yum.repos.d/epel-6.repo\*

fi

#安装epel-release包，我们先判断一下是否已经安装了，安装了就卸载掉重新下载安装

wget -P /etc/yum.repos.d/ <http://mirrors.aliyun.com/repo/epel-6.repo>

install\_mysqld() {

echo “Chose the version of mysql.”

select mysql\_v in 5.1 5.6

do

case $mysql\_v in

5.1)

cd /usr/local/src

[ -f mysql-5.1.72-linux-$ar-glibc23.tar.gz ] || wget <http://mirrors.sohu.com/mysql/MySQL-5.1/mysql-5.1.72-linux-$ar-glibc23.tar.gz>

check\_ok

tar zxf mysql-5.1.72-linux--$ar-glibc23.tar.gz

check\_ok

[ -d /usr/local/mysql ] && /bin/mv /usr/local/mysql /usr/local/mysql\_bak

mv mysql-5.1.72-linux--$ar-glibc23 /usr/local/mysql

if ! grep ‘^mysql:’ /etc/passwd

then

useradd –M mysql -s /sbin/nologin

fi

myum compat-libstdc++33

[ -d /data/mysql ] && /bin/mv /data/mysql /data/mysql\_bak

mkdir -p /data/mysql

chown -R mysql:mysql /data/mysql

cd /usr/local/mysql

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

check\_ok

/bin/cp support-files/my-huge.cnf /etc/my.cnf

check\_ok

sed -i ‘/^\[mysqld\]$/a\datadir = /data/mysql’ /etc/my.cnf

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

sed -i ‘s#^datadir=#datadir=/data/mysql#’ /etc/init.d/mysqld

chmod 775 /etc/init.d/mysqld

chkconfig --add mysqld

chkconfig mysqld on

service mysqld start

check\_ok

break

;;

5.6)

cd /usr/local/src

[ -f mysql-5.6.24-linux-glibc2.5-$ar.tar.gz ] || wget [http://mirrors.sohu.com/mysql/MySQL-5.6/mysql-5.6.24-linux -glibc2.5-$ar.tar.gz](http://mirrors.sohu.com/mysql/MySQL-5.6/mysql-5.6.24-linux%20-glibc2.5-$ar.tar.gz)

check\_ok

tar zxf mysql-5.6.24-linux -glibc2.5-$ar.tar.gz

check\_ok

[ -d /usr/local/mysql ] && /bin/mv /usr/local/mysql /usr/local/mysql\_`tate +%s`

mv mysql-5.6.24-linux -glibc2.5-$ar /usr/local/mysql

check\_ok

if ! grep ‘^mysql:’ /etc/passwd

then

useradd –M mysql -s /sbin/nologin

fi

myum compat-libstdc++33

[ -d /data/mysql ] && /bin/mv /data/mysql /data/mysql\_`date +%s`

mkdir -p /data/mysql

chown -R mysql:mysql /data/mysql

cd /usr/local/mysql

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

check\_ok

/bin/cp support-files/my-default.cnf /etc/my.cnf

check\_ok

sed -i ‘/^\[mysqld\]$/a\datadir = /data/mysql’ /etc/my.cnf

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

sed -i ‘s#^datadir=#datadir=/data/mysql#’ /etc/init.d/mysqld

chmod 775 /etc/init.d/mysqld

chkconfig --add mysqld

chkconfig mysqld on

service mysqld start

check\_ok

break

;;

\*)

echo “only 1(5.1) or 2(5.6)”

exit 1

;;

esac

done

}

#安装MySQL5.1或者MySQL5.6

install\_httpd() {

echo “Install apache version 2.2.”

cd /usr/local/src

[ -f https-2.2.16.tar.gz ] || wget http://syslab.comsenz.com/downloads/linux/httpd-2.2.16.tar.gz

check\_ok

tar zxf httpd-2.2.16.tar.gz && cd httpd-2.2.16

check\_ok

./configure \

--prefix=/usr/local/apache2 \

--with-included-apr \

--enabled-so \

--enabled-deflate=shared \

--enabled-expires=shared \

--enabled-rewrite=sharesd \

--with-pcre

check\_ok

make && make install

check\_ok

}

#安装httpd包

install\_php() {

echo -e “Install php. \nPlease chose the version of php.”

select php\_v in 5.3 5.6

do

case $php\_v in

5.3)

cd /usr/local/src/

[ -f php-5.3.10.tar.bz2 ] || wget <http://syslin.comsenz.com/downloads/linux/php-5.3.10.tar.bz2>

check\_ok

tar jxf php-5.3.10.tar.bz2 && cd php-5.3.10

for p in openssl-devel bzip2-devel \

libxml2-devel curl-devel libpng-devel \

libjpeg-devel freetype-devel libmcrypt-devel \

libtool-ltdl-devel perl-devel

do

myum $p

done

check\_ok

./configure \

--prefix=/usr/local/php \

--with-apxs2=/usr/local/apache2/bin/apxs \

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

--with-mysql=/usr/local/mysql \

--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 \

--enabled-soap \

--enabled-gd-native-ttf \

--enabled-mbstring \

--enabled-sockets \

--enabled-exif \

--disable-ipv6

check\_ok

make && make install

check\_ok

[ -f /usr/local/php/etc/php.ini ] || /bin/cp php.ini-production /usr/local/php/etc/php.ini

break

;;

5.6)

cd /usr/local/src/

[ -f php-5.6.6.tar.gz ] || wget <http://mirrors.sohu.com/php/php-5.6.6.tar.gz>

check\_ok

tar jxf php-5.6.6.tar.gz && cd php-5.6.6

for p in openssl-devel bzip2-devel \

libxml2-devel curl-devel libpng-devel \

libjpeg-devel freetype-devel libmcrypt-devel \

libtool-ltdl-devel perl-devel

do

myum $p

done

check\_ok

./configure \

--prefix=/usr/local/php \

--with-apxs2=/usr/local/apache2/bin/apxs \

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

--with-mysql=/usr/local/mysql \

--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 \

--enabled-soap \

--enabled-gd-native-ttf \

--enabled-mbstring \

--enabled-sockets \

--enabled-exif \

--disable-ipv6

check\_ok

make && make install

check\_ok

[ -f /usr/local/php/etc/php.ini ] || /bin/cp php.ini-production /usr/local/php/etc/php.ini

break

;;

\*)

echo “only 1(5.3) or 2(5.6)”

;;

esac

done

}

#安装php包

join\_apa\_php() {

sed -i ‘/AddType .\*.gz .tgz$/a\AddType application\ /x-httpd-php .php’ /usr/local/apache2/conf/httpd.conf

#找到含有AddType .\*.gz .tgz字符的行，并在此行下面添加AddType application\ /x-httpd-php .php行

check\_ok

sed –i ‘s/DirectoryIndex index.html/ DirectoryIndex index.php index.html index.htm/’ /usr/local/apache2/conf/httpd.conf

check\_ok

cat > /usr/local/apache2/htdocs/index.php <<EOF

<?php

phpinfo();

?>

EOF

#EOF可以把以EOF开始到EOF结束中间的字符重定向到某文件中去

if /usr/local/php/bin/php -I |grep -iq ‘date.timezone => no value’

then

sed -i ‘/;date.timezone =$/a\date.timezone = “Asia\ /Chongqing”’ /usr/local/php/etc/php.ini

fi

/usr/local/apache2/bin/apachectl restart

check\_ok

}

#这个函数定义php和Apache联动的配置动作

check\_service() {

if [ “$1” == “phpfpm” ]

then

s=”php-fpm”

else

s=$1

fi

n=`ps aux |grep “$s”|wc -l`

if [ $n -gt 1 ]

then

echo “$1 service is already started.”

else

if [ -f /etc/init.d/$1 ]

then

/etc/init.d/$1 start

check\_ok

else

install\_$1

fi

fi

}

#这个函数主要用于判断比如httpd等服务是否已经安装和启动，如果已经安装就不用再安装了

lamp() {

check\_service mysqld

check\_service httpd

install\_php

join\_apa\_php

echo “LAMP done, Please use ‘http://your ip/index.php’ to access.”

}

#这个函数其实作用就是把上面的所有函数揉合到一起去

install\_nginx() {

cd /usr/local/src

[ -f nginx-1.8.0.tar.gz ] wget <http://nginx.org/download/nginx-1.8.0.tar.gz>

check\_ok

tar zxf nginx-1.8.0.tar.gz

cd nginx-1.8.0

myum pcre-devel

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

check\_ok

make && make install

check\_ok

if [ -f /etc/init.d/nginx ]

then

/bin/mv /etc/init.d/nginx /etc/init.d/nginx\_`date +%s`

fi

curl <http://www.apelearn.com/stuby_v2/.nginx_init> -o /etc/init.d/nginx

check\_ok

chmod 755 /etc/init.d/nginx

chkconfig --add nginx

chkconfig nginx on

curl http://www.apelearn.com/stuby\_v2/.nginx\_conf -o /usr/local/nginx/conf/nginx.conf

check\_ok

service nginx start

check\_ok

echo -e “<?php\n phpinfo();\n?>” > /usr/local/nginx/html/index.php

#\n代表回车的意思，这样就可以把几行内容也成一行了

check\_ok

}

#这个函数是用来安装nginx的，其中定义了nginx编译安装时的全过程

install\_phpfpm() {

echo -e “Install php. \nPlease chose the version of php.”

select php\_v in 5.3 5.6

do

case $php\_v in

5.3)

cd /usr/local/src/

[ -f php-5.3.10.tar.bz2 ] || wget <http://syslab.comsenz.com/downloads/linux/php-5.3.10.tar.bz2>

check\_ok

tar jxf php-5.3.10.tar.bz2 && cd php-5.3.10

for p in openssl-devel bzip2-devel \

libxml2-devel curl-devel libpng-devel \

libjpeg-devel freetype-devel libmcrypt-devel \

libtool-ltdl-devel perl-devel

do

myum $p

done

if ! grep -q ‘^php-fpm:’ /etc/passwd

then

useradd -M -s /sbin/nologin php-fpm

check\_ok

fi

./configure \

--prefix=/usr/local/php-fpm \

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

--enabled-fpm \

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

--with-fpm-group=php-fpm \

--with-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 \

--enabled-soap \

--enabled-gd-native-ttf \

--enabled-ftp \

--enabled-mbstring \

--enabled-exif \

--enabled-zend-multibyte \

--disable-ipv6 \

--with-pear \

--with-curl \

--with-openssl \

check\_ok

make && make install

check\_ok

[ -f /usr/local/php-fpm/etc/php.ini ] || /bin/cp php.ini-production /usr/local/php-fpm/etc/php.ini

if /usr/local/php-fpm/bin/php -i |grep -iq ‘date.timezone => no value’

then

sed -i ‘/;date.timezone =$/a\date.timezone = “Asia\ /Chongqing”’ /usr/local/php-fpm/etc/php.ini

check\_ok

fi

[ -f /usr/local/php-fpm/etc/php-fpm.conf ] || curl <http://www.apelearn.com/study/.phpfpm_conf> -o /usr/local/hph-fpm/etc/php-fpm.conf

check\_ok

chmod 755 /etc/init.d/phpfpm

chkconfig phpfpm on

service phprpm start

check\_ok

break

;;

5.6)

cd /usr/local/src/

[ -f php-5.6.6.tar.gz ] || wget http://syslab.comsenz.com/downloads/linux/php-5.6.6.tar.gz

check\_ok

tar jxf php-5.6.6.tar.gz && cd php-5.6.6

for p in openssl-devel bzip2-devel \

libxml2-devel curl-devel libpng-devel \

libjpeg-devel freetype-devel libmcrypt-devel \

libtool-ltdl-devel perl-devel

do

myum $p

done

if ! grep -q ‘^php-fpm:’ /etc/passwd

then

useradd -M -s /sbin/nologin php-fpm

check\_ok

fi

./configure \

--prefix=/usr/local/php-fpm \

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

--enabled-fpm \

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

--with-fpm-group=php-fpm \

--with-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 \

--enabled-soap \

--enabled-gd-native-ttf \

--enabled-ftp \

--enabled-mbstring \

--enabled-exif \

--enabled-zend-multibyte \

--disable-ipv6 \

--with-pear \

--with-curl \

--with-openssl \

check\_ok

make && make install

check\_ok

[ -f /usr/local/php-fpm/etc/php.ini ] || /bin/cp php.ini-production /usr/local/php-fpm/etc/php.ini

if /usr/local/php-fpm/bin/php -i |grep -iq ‘date.timezone => no value’

then

sed -i ‘/;date.timezone =$/a\date.timezone = “Asia\ /Chongqing”’ /usr/local/php-fpm/etc/php.ini

check\_ok

fi

[ -f /usr/local/php-fpm/etc/php-fpm.conf ] || curl <http://www.apelearn.com/study/.phpfpm_conf> -o /usr/local/hph-fpm/etc/php-fpm.conf

check\_ok

chmod 755 /etc/init.d/phpfpm

chkconfig phpfpm on

service phprpm start

check\_ok

break

;;

\*)

echo “only 1(5.3) or 2(5.6)”

;;

esac

done

}

#此函数定义安装php，其实同上面的那个安装PHP函数差不多，只是编译安装方法不一样

lnmp() {

check\_service mysql

check\_service nginx

check\_service phpfpm

echo “The LNMP done, Please use ‘http://your ip/index.php’ to access.”

#此函数定义了在选择安装lnmp时要根据mysql 、nginx、phpfpm这三个函数安装相对应的程序

read -p “Please chose which type env you install, (lamp|lnmp)?” t

case $t in

lamp)

lamp

;;

lnmp)

lnmp

;;

\*)

echo “Only ‘lamp’ or ‘lnmp’ your can input.”

;;

esac

这一段脚本主要是询问用户是安装lamp还是lnmp，比如选择lamp，就会执行lamp函数里面相对应程序的函数，安装指定包

**注：所有函数或者脚本里的下载地址和程序包名由于是预定义的，所以肯定不准确，如果重新下载，下载源地址和下载的程序名称就和脚本里的有冲突，所以在使用时要根据当时环境找下载源并对安装包解压后的目录名做更改**