第一周：个人资料及环境准备

个人资料：

知识星球编号：52183367

常用名：莫不言

联系方式：**wechat**:15345313565 ;**QQ**:974466968;**Tel**:15345313565

目前职业：在职（7年）

所在地区：山东济南

熟悉的编程语言：asp.net /python

自我介绍：做过几年运维，一年前转型信息安全，属于大龄转型的情况，很多技术方面都是浅尝辄止，感谢信安之路，希望信安之路此次的活动对我有所帮助，扎实一下技术。

环境搭建：

基础环境:CentOS Linux release 7.2.1511

nginx安装

使用源码编译安装，包括具体的编译参数信息。

安装make：

```
yum -y install gcc automake autoconf libtool make
```

安装g++:

```
yum install gcc gcc-c++
```

1.选定源码目录

可以是任何目录，我这里选定的是/usr/local/src

```
cd /usr/local/src
```

2.安装PCRE库

ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.41.tar.gz 下载最新的 PCRE 源码包，使用下面命令下载编译和安装 PCRE 包：

```
cd /usr/local/src

wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.41.tar.gz

tar -zxvf pcre-8.41.tar.gz

cd pcre-8.41

./configure

make && make install
```

3.安装zlib库

http://zlib.net/zlib-1.2.11.tar.gz 下载最新的 zlib 源码包，使用下面命令下载编译和安装 zlib包：

```
cd /usr/local/src

wget http://zlib.net/zlib-1.2.11.tar.gz

tar -zxvf zlib-1.2.11.tar.gz

cd zlib-1.2.11

./configure

make && make install
```

4.安装ssl

```
cd /usr/local/src

wget https://www.openssl.org/source/openssl-1.1.0g.tar.gz

tar -zxvf openssl-1.1.0g.tar.gz
```

5.安装nginx

以安装ngixn-1.10.2为例子

下面是把 Nginx 安装到 /usr/local/nginx 目录下的详细步骤：

```
cd /usr/local/src

wget http://nginx.org/download/nginx-1.10.2.tar.gz

tar -zxvf nginx-1.10.2.tar.gz

cd nginx-1.10.2
```

1、添加nginx用户和用户组

```
groupadd -r nginx

useradd -r -g nginx nginx
```

2、配置nginx安装参数

```
./configure \
```

```
--prefix=/usr/local/nginx \
--sbin-path=/usr/local/nginx/sbin/nginx \
--conf-path=/usr/local/nginx/nginx.conf \
--pid-path=/usr/local/nginx/nginx.pid \
--user=nginx \
--group=nginx \
--with-http_ssl_module \
--with-http_flv_module \
--with-http_mp4_module  \
--with-http_stub_status_module \
--with-http_gzip_static_module \
--http-client-body-temp-path=/var/tmp/nginx/client/ \
--http-proxy-temp-path=/var/tmp/nginx/proxy/ \
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ \
--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \
--http-scgi-temp-path=/var/tmp/nginx/scgi \
--with-pcre=/usr/local/src/pcre-8.41 \
--with-zlib=/usr/local/src/zlib-1.2.11 \
--with-openssl=/usr/local/src/openssl-1.1.0g
```

编译安装

```
make && make install
```

其中/var/tmp/nginx/client 需要手动创建。

安装成功后 /usr/local/nginx 目录下如下:

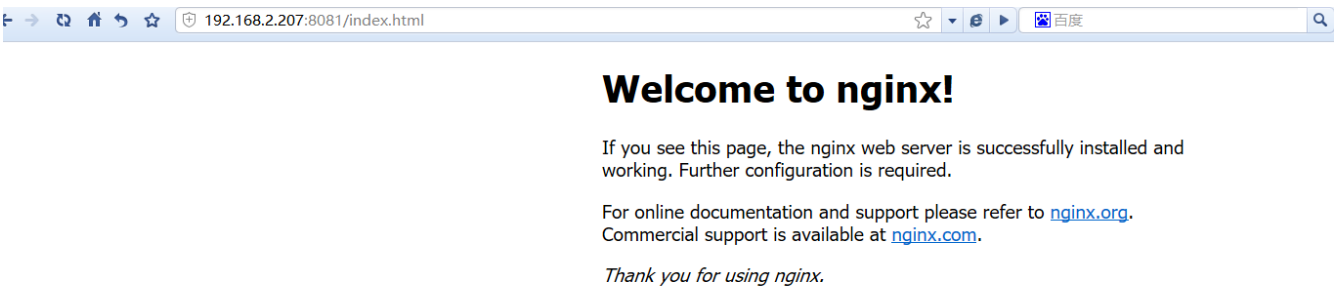
```
[root@localhost nginx]# cd /usr/local/nginx/;ls
conf                  fastcgi_params.default  logs               nginx.conf.default  scgi_params.default
fastcgi.conf          html                    mime.types         nginx.pid           uwsgi_params
fastcgi.conf.default  koi-utf                 mime.types.default sbin                uwsgi_params.default
fastcgi_params        koi-win                 nginx.conf         scgi_params         win-utf
```

3、启动

我这里修改了nginx.conf的端口号为8081，运行/usr/local/nginx/sbin/nginx 命令来启动 Nginx。

```
netstat -ano|grep 8081
```

打开浏览器访问对应 IP，浏览器出现 Welcome to nginx! 则表示 Nginx 已经安装并运行成功。



php-fpm php7安装配置

nginx本身不能处理PHP，它只是个web服务器，当接收到请求后，如果是php请求，则发给php解释器处理，并把结果返回给客户端。

nginx一般是把请求发fastcgi管理进程处理，fascgi管理进程选择cgi子进程处理结果并返回被nginx，

下面以php-fpm为例介绍如何使nginx支持PHP。

一、编译安装php-fpm

什么是PHP-FPM

PHP-FPM是一个PHP FastCGI管理器，是只用于PHP的,可以在 http://php-fpm.org/download下载得到.

PHP-FPM其实是PHP源代码的一个补丁，旨在将FastCGI进程管理整合进PHP包中。必须将它patch到你的PHP源代码中，在编译安装PHP后才可以使用。

新版PHP已经集成php-fpm了，不再是第三方的包了，推荐使用。PHP-FPM提供了更好的PHP进程管理方式，可以有效控制内存和进程、可以平滑重载PHP配置，比spawn-fcgi具有更多优点，所以被PHP官方收录了。在./configure的时候带 –enable-fpm参数即可开启PHP-FPM，其它参数都是配置php的，具体选项含义可以查看这里。

安装前准备 centos下执行

```
 yum -y install gcc gcc-c++ glibc
```

```
yum -y install libmcrypt-devel mhash-devel libxslt-devel \
libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel \
zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel \
ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel \
krb5 krb5-devel libidn libidn-devel openssl openssl-devel
```

```
cd /usr/local/src
wget http://php.net/get/php-5.6.27.tar.gz/from/a/mirror
tar -zvxf mirror
cd php-5.6.27

./configure --prefix=/usr/local/php  --enable-fpm --with-mcrypt \
--enable-mbstring --enable-pdo --with-curl --disable-debug  --disable-rpath \
--enable-inline-optimization --with-bz2  --with-zlib --enable-sockets \
--enable-sysvsem --enable-sysvshm --enable-pcntl --enable-mbregex \
--with-mhash --enable-zip --with-pcre-regex --with-mysql --with-mysqli \
--with-gd --with-jpeg-dir --with-freetype-dir --enable-calendar
```

```
make && make install
```

编译过程，出现两次报错：

1、“PHP configure: error: mcrypt.h not found. Please reinstall libmcrypt.”，解决如下：

```
wget ftp://mcrypt.hellug.gr/pub/crypto/mcrypt/attic/libmcrypt/libmcrypt-2.5.7.tar.gz
tar -zxvf libmcrypt-2.5.7.tar.gz
cd libmcrypt-2.5.7
./configure  -prefix=/usr/local
make && make install
```

2、报错"configure: error: Don't know how to define struct flock on this system"，解决如下：

```
echo /usr/local/lib >> /etc/ld.so.conf
echo /usr/local/lib64 >> /etc/ld.so.conf
ldconfig -v                         # 使之生效
```

重新编译：

```
./configure --prefix=/usr/local/php  --enable-fpm --with-mcrypt \
--enable-mbstring --enable-pdo --with-curl --disable-debug  --disable-rpath \
--enable-inline-optimization --with-bz2  --with-zlib --enable-sockets \
--enable-sysvsem --enable-sysvshm --enable-pcntl --enable-mbregex \
--with-mhash --enable-zip --with-pcre-regex --with-mysql --with-mysqli \
--with-gd --with-jpeg-dir --with-freetype-dir --enable-calendar

make && make install
```

```
Installing PHP FPM man page:      /usr/local/php/php/man/man8/
Installing PHP FPM status page:   /usr/local/php/php/php/fpm/
Installing PHP CGI binary:        /usr/local/php/bin/
Installing PHP CGI man page:      /usr/local/php/php/man/man1/
Installing build environment:     /usr/local/php/lib/php/build/
Installing header files:           /usr/local/php/include/php/
Installing helper programs:       /usr/local/php/bin/
  program: phpize
  program: php-config
Installing man pages:             /usr/local/php/php/man/man1/
  page: phpize.1
  page: php-config.1
Installing PEAR environment:      /usr/local/php/lib/php/
[PEAR] Archive_Tar    - installed: 1.4.0
[PEAR] Console_Getopt - installed: 1.4.1
[PEAR] Structures_Graph- installed: 1.1.1
[PEAR] XML_Util       - installed: 1.3.0
[PEAR] PEAR           - installed: 1.10.1
Wrote PEAR system config file at: /usr/local/php/etc/pear.conf
You may want to add: /usr/local/php/lib/php to your php.ini include_path
/usr/local/src/php-5.6.27/build/shtool install -c ext/phar/phar.phar /usr/local/php/bin
ln -s -f phar.phar /usr/local/php/bin/phar
Installing PDO headers:           /usr/local/php/include/php/ext/pdo/
```

以上就完成了php-fpm的安装

下面是对php-fpm运行用户进行设置

二、php-fpm设置

1、为php提供配置文件

```
 cp php.ini-production /usr/local/php/lib/php.ini
```

2、为php-fpm提供配置文件

```
  cd /usr/local/php
  cp etc/php-fpm.conf.default etc/php-fpm.conf
  vim etc/php-fpm.conf
```

修改 user = www-data group = www-data

如果www-data用户不存在，那么先添加www-data用户

```
groupadd www-data
useradd -g www-data www-data
```

```
修改
pm.max_children = 150

pm.start_servers = 8

pm.min_spare_servers = 5

pm.max_spare_servers = 10

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

3、启动php-fpm

执行

```
/usr/local/php/sbin/php-fpm
```

使用如下命令来验正（如果此命令输出有中几个php-fpm进程就说明启动成功了）：

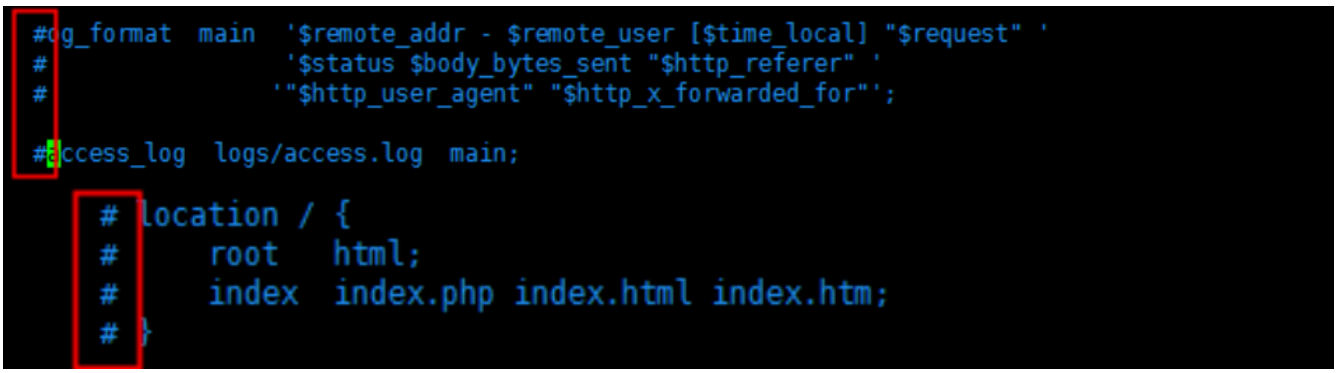
```
ps aux | grep php-fpm
```

4、nginx和php-fpm整合

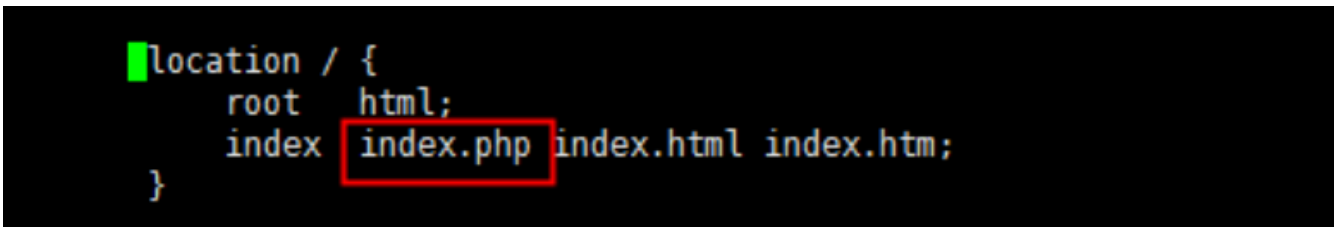
```
vim /usr/local/nginx/nginx.conf
```

修改如下：

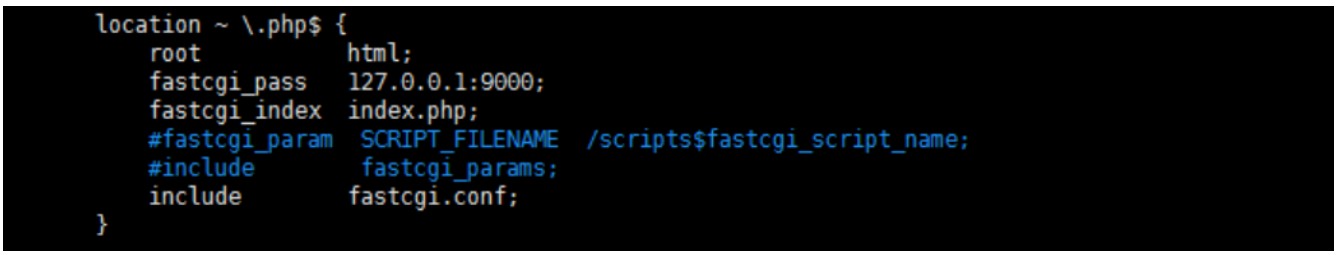
红框中“#”号去掉



添加红框中的内容：



去掉注释：



重新载入nginx的配置文件：

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

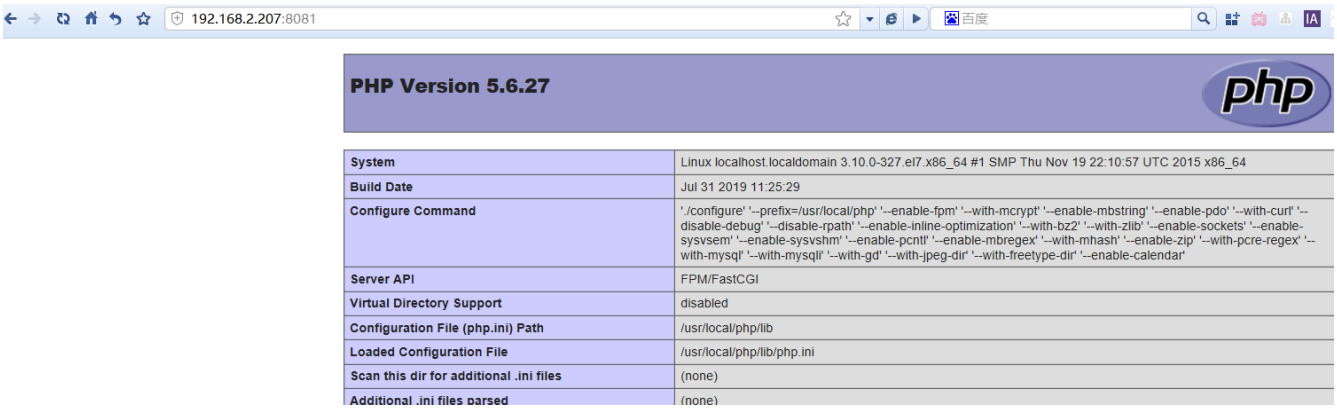
5、测试php文件

在/usr/local/nginx/html下创建index.php文件，输入如下内容：

```
<?php
    phpinfo();
?>
```

6、浏览器访问

访问http://ip/index.php，皆可以见到php信息了。



安装Mysql5.7

1、下载安装包

http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.16-linux-glibc2.5-x86_64.tar 推荐下载通用安装方法的TAR包

2.检查库文件是否存在，如有删除。

```
[root@localhost Desktop]$ rpm -qa | grep mysql
mysql-libs-5.1.52-1.el6_0.1.x86_64
[root@localhost ~]$ rpm -e mysql-libs-5.1.52.x86_64 --nodeps
[root@localhost ~]$
```

3.检查mysql组和用户是否存在，如无创建。

```
[root@localhost ~]$ cat /etc/group | grep mysql
mysql:x:490:
[root@localhost ~]$ cat /etc/passwd | grep mysql
mysql:x:496:490::/home/mysql:/bin/bash
```

以上为默认存在的情况，如无，执行添加命令：

```
[root@localhost ~]$groupadd mysql
[root@localhost ~]$useradd -r -g mysql mysql
//useradd -r参数表示mysql用户是系统用户，不可用于登录系统。
```

4.解压TAR包，更改所属的组和用户

```
[root@localhost ~]$ cd /usr/local/
[root@localhost local]$ tar xvf mysql-5.7.12-linux-glibc2.5-x86_64.tar
[root@localhost local]$ ls -l
total 1306432
-rwxr--r--. 1 root root  668866560 Jun  1 15:07 mysql-5.7.12-linux-glibc2.5-x86_64.tar
-rw-r--r--. 1 7161 wheel 638960236 Mar 28 12:54 mysql-5.7.12-linux-glibc2.5-x86_64.tar.gz
-rw-r--r--. 1 7161 wheel  29903372 Mar 28 12:48 mysql-test-5.7.12-linux-glibc2.5-
x86_64.tar.gz
[root@localhost local]$ tar xvfz mysql-5.7.12-linux-glibc2.5-x86_64.tar.gz
[root@localhost local]$ mv mysql-5.7.12-linux-glibc2.5-x86_64 mysql
```

```
[root@localhost local]$ ls -l
total 1306436
drwxr-xr-x. 2 root root       4096 Dec  4  2009 bin
drwxr-xr-x. 2 root root       4096 Dec  4  2009 etc
drwxr-xr-x. 2 root root       4096 Dec  4  2009 games
drwxr-xr-x. 2 root root       4096 Dec  4  2009 include
drwxr-xr-x. 2 root root       4096 Dec  4  2009 lib
drwxr-xr-x. 3 root root       4096 Dec  2 14:36 lib64
drwxr-xr-x. 2 root root       4096 Dec  4  2009 libexec
drwxr-xr-x. 9 7161 wheel      4096 Mar 28 12:51 mysql
-rwxr--r--. 1 root root  668866560 Jun  1 15:07 mysql-5.7.12-linux-glibc2.5-x86_64.tar
-rw-r--r--. 1 7161 wheel 638960236 Mar 28 12:54 mysql-5.7.12-linux-glibc2.5-x86_64.tar.gz
-rw-r--r--. 1 7161 wheel  29903372 Mar 28 12:48 mysql-test-5.7.12-linux-glibc2.5-
x86_64.tar.gz
drwxr-xr-x. 2 root root       4096 Dec  4  2009 sbin
drwxr-xr-x. 6 root root       4096 Dec  2 14:36 share
drwxr-xr-x. 2 root root       4096 Dec  4  2009 src
[root@localhost local]$ chown -R mysql mysql/
[root@localhost local]$ chgrp -R mysql mysql/
```

5.安装和初始化数据库

```
   [root@localhost mysql]$ bin/mysql_install_db --user=mysql --basedir=/usr/local/mysql/ --
datadir=/usr/local/mysql/data/
2016-06-01 15:23:25 [WARNING] mysql_install_db is deprecated. Please consider switching to
mysqld --initialize
2016-06-01 15:23:30 [WARNING] The bootstrap log isn't empty:
2016-06-01 15:23:30 [WARNING] 2016-06-01T22:23:25.491840Z 0 [Warning] --bootstrap is
deprecated. Please consider using --initialize instead
2016-06-01T22:23:25.492256Z 0 [Warning] Changed limits: max_open_files: 1024 (requested
5000)
2016-06-01T22:23:25.492260Z 0 [Warning] Changed limits: table_open_cache: 431 (requested
2000)
```

> 如果改变默认安装路径，则需要 1）/etc/my.cnf、/etc/init.d/mysqld中修改 basedir='/apps/mysql' datadir='/apps/mysql/data' 2）创建ln mkdir -p /usr/local/mysql/bin ln -s /apps/mysql/bin/mysqld /usr/local/mysql/bin/mysqld

```
[root@localhost mysql]$

[root@localhost mysql]$ cp -a ./support-files/my-default.cnf /etc/my.cnf
[root@localhost mysql]$ cp -a ./support-files/mysql.server  /etc/init.d/mysqld
[root@localhost mysql]$ cd bin/
[root@localhost bin]# ./mysqld_safe --user=mysql &
[1] 2932
[root@localhost bin]# 2016-06-01T22:27:09.708557Z mysqld_safe Logging to
'/usr/local/mysql/data/localhost.localdomain.err'.
2016-06-01T22:27:09.854913Z mysqld_safe Starting mysqld daemon with databases from
/usr/local/mysql/data

[root@localhost bin]# /etc/init.d/mysqld restart
Shutting down MySQL..2016-06-01T22:27:50.498694Z mysqld_safe mysqld from pid file
```

```
/usr/local/mysql/data/localhost.localdomain.pid ended
 SUCCESS!
Starting MySQL. SUCCESS!
[1]+  Done                    ./mysqld_safe --user=mysql
[root@localhost bin]$
//设置开机启动
[root@localhost bin]$ chkconfig --level 35 mysqld on
[root@localhost bin]$
```

6.初始化密码

mysql5.7会生成一个初始化密码，而在之前的版本首次登陆不需要登录。

```
[root@localhost bin]$ cat /root/.mysql_secret
# Password set for user 'root@localhost' at 2016-06-01 15:23:25
,xxxxxR5H9
[root@localhost bin]$./mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.12

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SET PASSWORD = PASSWORD('mysql');
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

7.添加远程访问权限（从安全角度，不建议开放）

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A


Database changed
mysql> update user set host = '%' where user = 'root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0


mysql> select host, user from user;
+-----------+-----------+
| host      | user      |
```

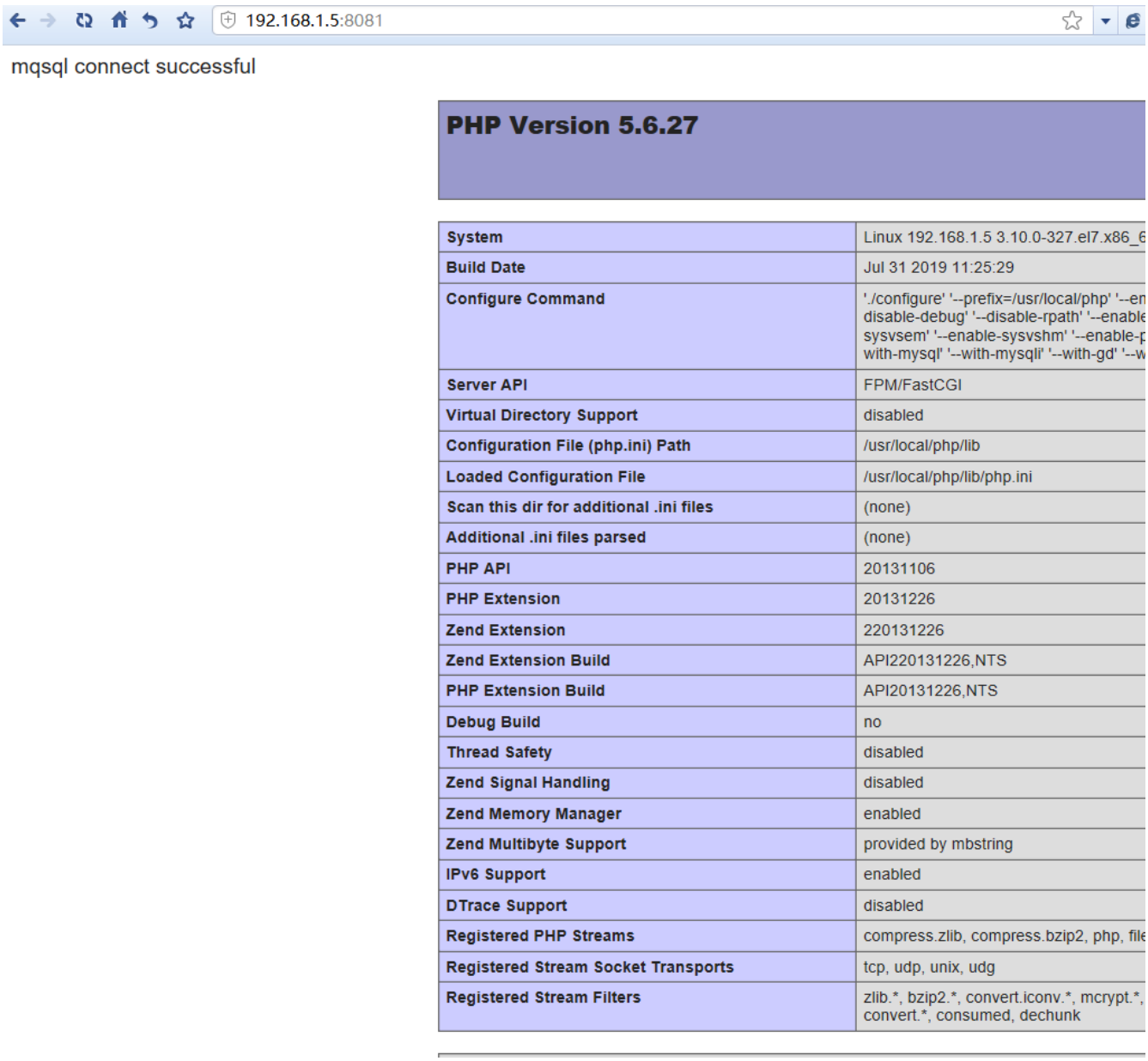
```
+-----------+-----------+
| %         | root      |
| localhost | mysql.sys |
+-----------+-----------+
//重启生效
/etc/init.d/mysqld restart
```

```
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select host,user from user;
+-----------+-----------+
| host      | user      |
+-----------+-----------+
| %         | root      |
| localhost | mysql.sys |
+-----------+-----------+
2 rows in set (0.05 sec)

mysql>
```

php连接mysql

在/usr/local/nginx/html/新建index.php文件，编辑代码如下：

```
<?php

  $con = mysql_connect("localhost","root","mysql");
  if (!$con)
  {
        die(mysql_error());
  }
  else
  {
        echo "mqsql connect successful";
  }
  mysql_close($con);
  phpinfo();
?>
```



安全加固

nginx加固

禁止目录浏览

编辑配置文件，HTTP模块添加如下一行内容：

```
autoindex off;
```

隐藏版本信息

编辑配置文件，添加http模块中如下一行内容：

```
server_tokens off;
```

限制HTTP请求方法

```
if ($request_method !~ ^(GET|HEAD|POST)$ ) {
return 444;
}
```

备注：只允许常用的GET和POST方法，顶多再加一个HEAD方法

控制超时时间

```
client_body_timeout 10;  #设置客户端请求主体读取超时时间
client_header_timeout 10;  #设置客户端请求头读取超时时间
keepalive_timeout 5 5;  #第一个参数指定客户端连接保持活动的超时时间，第二个参数是可选的，它指定了消息头保持活动的有效时间
send_timeout10;  #指定响应客户端的超时时间
```

封杀各种user-agent

```
if ($http_user_agent ~*
"java|python|perl|ruby|curl|bash|echo|uname|base64|decode|md5sum|select|concat|httprequest|httpclient|nmap|scan" ) {
    return 403;
}
if ($http_user_agent ~* "" ) {
    return 403;
}
```

url 参数过滤敏感字

```
if ($query_string ~* "union.*select.*\(") {
    rewrite ^/(.*)$  $host  permanent;
}

if ($query_string ~* "concat.*\(") {
    rewrite ^/(.*)$  $host  permanent;
}
```

mysql加固

修改mysql管理员名称

改变默认MySQL管理员的名称，将系统的默认管理员root 改为admin，防止被列举，将执行过程

```
mysql> update  user set  user="admin" where user="root";
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
mysql> flush privileges;
```

禁止MySQL对本地文件进行存取

```
# vi /etc/my.cnf
增加语句为
set-variable=local-infile=0
```

限制一般用户浏览其他用户数据库

```
可以在启动MySQL服务器时加-skip-show-database启动参数就能够达到目的。
-skip-show-database
```

可以在启动MySQL服务器时加–skip-show-database启动参数就能够达到目的。

禁止远程连接mysql

为我们的mysql只需要本地的php脚本进行连接，所以我们无需开socket进行监听，那么我们完全可以关闭监听的功能。 有两个方法实现： *配置my.cnf文件，在[mysqld]部分添加skip-networking参数 *mysqld服务器中参数中添加-skip-networking启动参数来使mysql不监听任何TCP/IP连接，增加安全性。如果要进行mysal的管理的话，可以在服务器本地安装一个phpMyadmin来进行管理。

数据库文件的安全

默认的mysql是安装在/usr/local/mysql目录下的，那么对应的数据库文件就是在 /usr/local/mysql/var目录下，要保证该目录不能让未经授权的用户访问后把数据库打包拷贝，所以要限制对该目录的访问。 我们修改该目录的所属用户和组是mysql，同时改变访问权限：

```
# chown-R mysql.mysql/usr/local/mysql/var
# chmod-R go-rwx/usr/local/mysql/var
```

php加固

PHP版本信息

为了防止黑客获取服务器中 PHP 版本的信息，您可以禁止该信息在 HTTP 头部内容中泄露：

```
expose_php = off
```

这样设置之后，黑客在执行telnet 80尝试连接您的服务器的时候，将无法看到 PHP 的版本信息。

SQL 注入防护

magic_quotes_gpc选项默认是关闭的。如果打开该选项，PHP 将自动把用户提交对 SQL 查询的请求进行转换（例如，把 ' 转换为 \' 等），这对于防止 SQL 注入攻击有很大作用，因此建议您将该选项设置为：

```
magic_quotes_gpc = on
```

注意： 该选项参数在 PHP 5.4.0 以后的版本中已被移除。

错误信息控制

一般 PHP 环境在没有连接到数据库或者其他情况下会有错误提示信息，错误信息中可能包含 PHP 脚本当前的路径信息或者查询的 SQL 语句等信息，这类信息如果暴露给黑客是不安全的，因此建议您禁止该错误提示：

```
display_errors = Off
```