# 个人资料

# 计划任务

任务目标：准备学习环境，学习WEB服务器搭建过程，并做相应的加固学习。

操作系统：主要有Windows和Linux类，Linux在实际工作中主要以Centos/Ubuntu为常见。

推荐环境：Linux+nginx+php+mysql（简称lnmp）

报告要求：将整个搭建过程进行详细记录，收集网络上的加固文档，进行加固操作，学习加固并思考其加固原因及威胁。

最终目标：能够运行php代码并且可以试用php连接MySQL，成功执行MySQL的SQL语句。

拓展任务：尝试搭建其他类型的WEB环境，基于apache环境，或者Windows Server的IIS环境。

# 基础知识

1. 操作系统环境

虚拟化：在日常学习及练习过程中，需要安装各种操作系统环境，但我们无物理环境可用。推荐选择使用虚拟化软件，类似vmware workstation(需要激活注册码)、VirtualBox （是一款开源虚拟机软件），此类软件基于本地电脑的软硬件可以虚拟出多台主机可以使用，安装软件后，在软件内可以新建主机，然后选择操作系统镜像安装，可同时开启多台虚拟机系统（需要本地电脑大量内存和硬盘支撑）。

VMware Workstation v14 for Windows：

FF31K-AHZD1-H8ETZ-8WWEZ-WUUVA

CV7T2-6WY5Q-48EWP-ZXY7X-QGUWD

主流操作系统环境以Windows系列和Linux系列为主。Windows系统有服务器版本、PC（个人电脑）版本及其他数据中心版本等，常见服务器端使用的以Windows Server 2008(R2)、Windows Server 2012/2016等。一些原版的Windows操作系统镜像ISO文件，可以在这个网站上获取（https://msdn.itellyou.cn/），然后在虚拟机软件内安装。

Linux操作系统具有多个发行版，各公司系列的发行版可能在功能及命令使用上略有不同，但原理相同，比如[Centos](https://www.centos.org/download/)属于红帽Redhat系列的社区免费版，而[Ubuntu](https://ubuntu.com/download)属于Debian系，这两者之间的区别可能体现在配置文件路径不一样。个人使用推荐Ubuntu而服务器版本推荐使用Centos。

包管理器，在Linux操作系统上安装软件不同于Windows，Linux软件包可以网络上下载到操作系统环境上安装，也可以通过包管理器来安装。在centos上，包管理器多使用yum，在yum配置文件里面有所有软件包安装下载的网络地址，我们称之为“yum源”（可以多个），通过yum安装时，会去网络上去下载软件到本地进行安装，如果本地挂载了ISO镜像也可以安装。

Yum，基于rpm包管理，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载、安装。

1. WEB服务器环境

WEB服务器的搭建，需要操作系统、WEB服务中间件、数据库以及编程语言环境，常用的搭配有lamp(linux+apache+mysql+php)、lnmp(linux+ngnix+mysql+php)、wamp(Windows下的Apache+Mysql/MariaDB+Perl/PHP/Python)，或者Windows上使用IIS中间件的环境。（参考[在Windows Server 2008上安装IIS 7.0](https://msdn.microsoft.com/zh-cn/dd727782.aspx)）

常常可以使用lamp一键安装包实现安装，也可以在Windows上使用phpstudy/wampServer一类集成环境快速搭建。

1. 安全加固

操作系统及软件服务本身都存在默认配置，这些默认配置是为了快速体验以达到简单使用的目的，提供便利性的同时可能存在安全性问题，所以软件服务本身的安全加固是以配置整改和补丁升级为主的。及时升级补丁可以解决掉低版本中可能存在的大量BUG及漏洞，而对配置进行调整可以使安全性达到最佳状态。

不同软件服务在不同使用者手上，基于实际情况（同时兼顾安全性与并于管理使用的合理性）而调整的软件基本配置信息，一般称为安全基线，分为操作系统安全基线、tomcat安全基线、数据库软件安全基线等等。

安全加固常常要理解此操作的原因，未进行加固的可能危害、具体执行的操作步骤、注意事项及回滚等等，其中理解其危害与合理加固最重要。

# 搭建环境

此搭建过程，默认读者已经有相关的Linux操作基础、MySQL数据库基础及PHP语言基础，如果你还不太了解，建议通过一下链接进行初步学习：

* <https://www.linuxprobe.com/chapter-00.html>
* <https://github.com/OpenCyberTranslationProject/Linux-Basics-for-Hackers>
* <https://www.runoob.com/mysql/mysql-tutorial.html>
* <https://www.runoob.com/php/php-tutorial.html>
* <https://msdn.microsoft.com/zh-cn/dd727782.aspx>

基础环境需要，安装VMware Workstaion软件，点击“创建新的虚拟机”，按照要求一步步操作，在“安装程序光盘映像文件(iso)(M)：”处选择本地电脑下载的操作系统ISO文件，然后直至操作系统安装完成。

## lnmp环境搭建

环境构成：

Centos 7.5.1804

[root@localhost ~]# ifconfig ens33

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.228.136 netmask 255.255.255.0 broadcast 192.168.228.255

[root@localhost ~]# cat /etc/redhat-release

CentOS Linux release 7.5.1804 (Core)

[root@localhost ~]#

PHP 7.3.7

Nginx 1.6.1

MariaDB 5.7

### 安装nginx

主要命令如下：

1. 首先更新系统软件

#yum update

1. 安装nginx源

# yum localinstall http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm

1. 安装nginx

# yum install nginx

1. 启动nginx

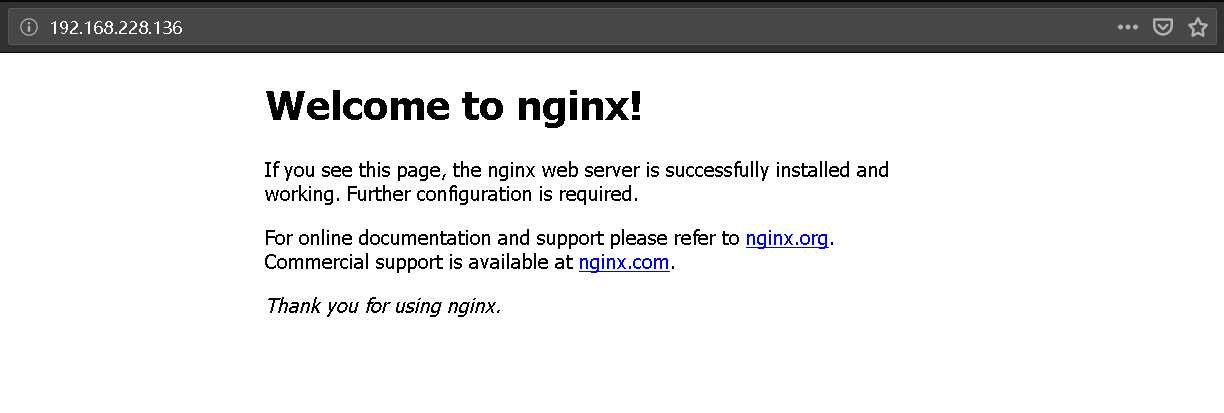
# service nginx start

或者

# systemctl start nginx

1. 访问验证

浏览器访问 http://[你的IP]，打开是否有“Welcome to nginx!”的欢迎页面。



Nginx官方安装文档：http://nginx.org/en/linux\_packages.html#RHEL-CentOS（Linux平台安装apache及httpd的方法类似，都可以参照官方技术文档）。

### 安装MySQL/ MariaDB

#### MySQL

MySQL原本未开源数据库，被Oracle收购之后，作者利用源码重新包装成了MariaDB，但MySQL依然可以使用，两种在使用上几乎没有区别（所以如果你无法安装MySQL，可以安装MariaDB）。

1. 首先安装MySQL源

# yum localinstall http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm

1. 安装MySQL

# yum install mysql-community-server

1. 安装MySQL的开发包或者称之为相关依赖的软件包

# yum install mysql-community-devel

1. 启动MySQL

# service mysqld start

或者

# systemctl start mysqld

1. 查看MySQL服务状态

# service mysqld status

或者

# systemctl status mysqld

或者mariadb

[root@localhost ~]# systemctl status mariadb

{ MySQL安装默认生成随机密码，需要我们重置密码。

1. 获取MySQL默认密码

# grep 'temporary password' /var/log/mysqld.log

1. 使用默认密码登陆数据库

# mysql -uroot –p

输入密码登录

1. 更换密码

mysql>  ALTER USER  'root'@'localhost' IDENTIFIED BY  '123456';

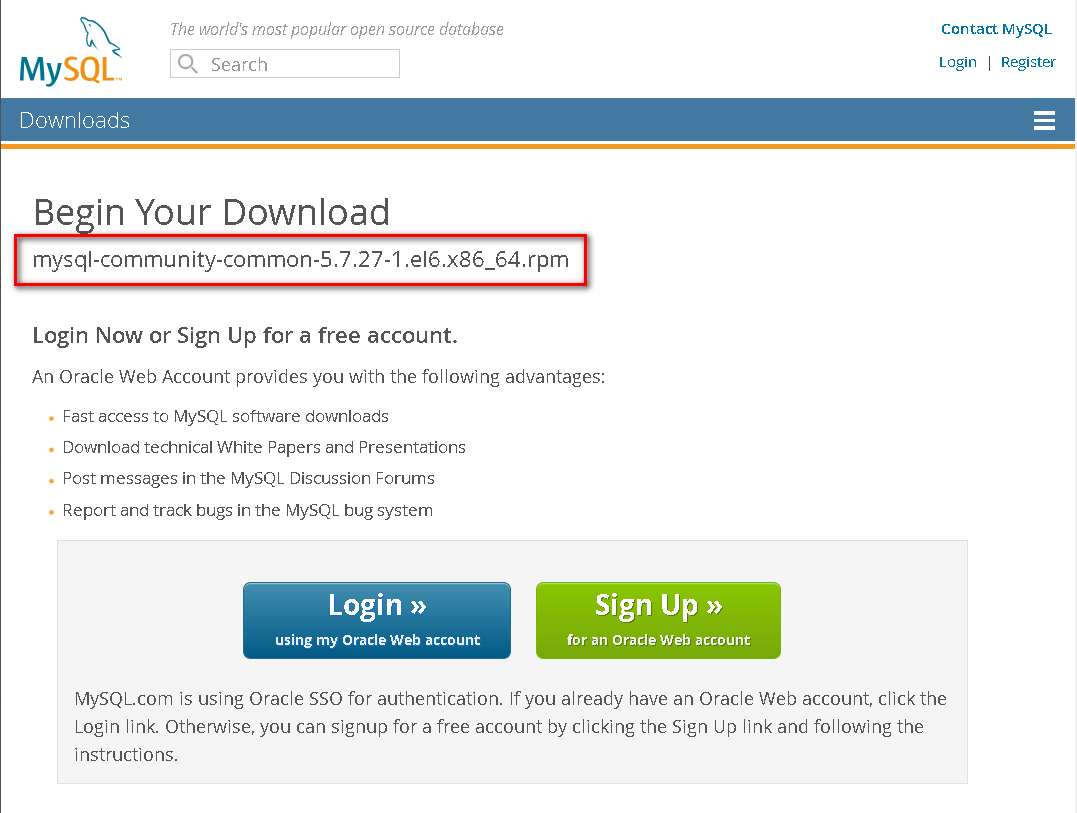
1. 退出并使用新密码尝试登陆数据库

mysql> quit;

mysql -uroot -p

}

软件包的版本号及名称可以现在官方网站上获取，比如：



可以将rpm格式软件包下载到操作系统上，用以下命令安装：

[root@localhost ~]# yum -y install mysql-community-common-5.7.27-1.el6.x86\_64.rpm

#### MariaDB

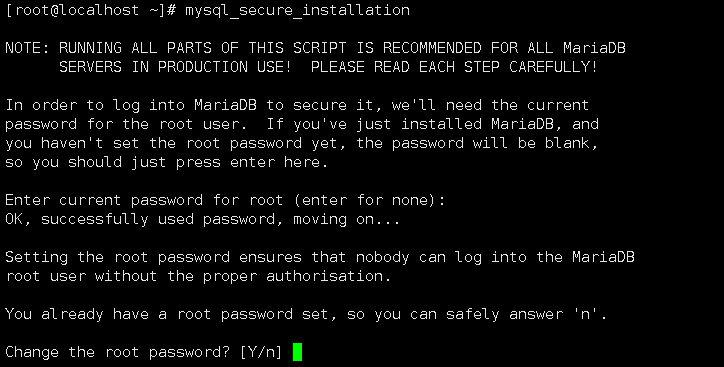
MariaDB的安装以下为例：

# yum install mariadb\*

# systemctl enable mariadb

MariaDB安装完成，可以使用以下命令进行交互式的数据库相关的设置：

[root@localhost ~]# mysql\_secure\_installation



然后登录数据库查看信息：

[root@localhost]# mysql -u root -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 2

Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

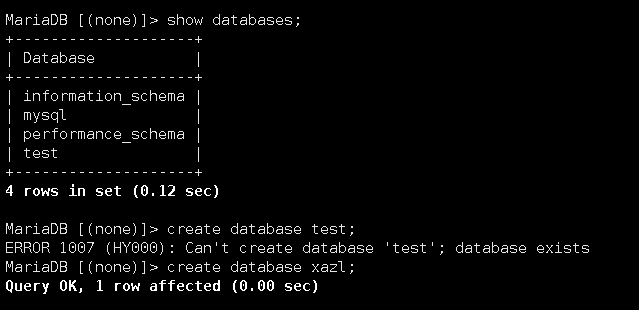
| test |

+--------------------+

4 rows in set (0.00 sec)

MariaDB [(none)]>

管理数据库，可以使用命令行方式直接登录操作，也可以使用图形化软件navicat等客户端（需开放远程连接），同样可以安装phpmyadmin这样的web管理服务。



MySQL/MariaDB官方文档：

<https://dev.mysql.com/downloads/mysql/5.7.html#downloads>

<https://dev.mysql.com/doc/refman/8.0/en/linux-installation-yum-repo.html>

<https://mariadb.com/kb/en/library/yum/>

### 安装PHP

#### 源码编译安装

1. 下载PHP软件包，<https://www.php.net/downloads.php>
2. 解压下载的PHP 源代码包

tar zxf php-x.x.x #(解压.tar.gz)

tar jxf php-x.x.x #(解压.tar.bz2)

1. 配置并构建 PHP，在此步骤您可以使用很多选项自定义PHP，例如启用某些扩展等。 运行 ./configure --help 命令来获得完整的可用选项清单。在本示例中，我们仅进行包含 PHP-FPM 和 MySQL 支持的简单配置。

yum -y install libxml2 libxml2-devel openssl openssl-devel curl-devel libjpeg-devel libpng-devel freetype-devel libmcrypt-devel libzip-devel pcre-devel

然后进行编译安装

cd ../php-x.x.x

./configure --enable-mysqlnd --enable-fpm --enable-xml --enable-mbstring --enable-opcache --enable-session --enable-soap --with-mysqli --with-pdo-mysql

make

sudo make install

{

其中在执行第三步命令 make 时，出现以下错误：

checking for pkg-config... /usr/bin/pkg-config

configure: error: libxml2 not found. Please check your libxml2 installation.

但是通过rpm -qa | grep libxml2就可以查到已经安装：

[root@localhost php-7.3.7]# rpm -qa | grep libxml2

libxml2-2.9.1-6.el7\_2.3.x86\_64

其原因时未安装libxml2-dev软件包：

ubuntu/debian:

***apt-get install libxml2-dev***

Centos/redhat:

***yum install libxml2-devel***

}

1. 创建配置文件，并将其复制到正确的位置。

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

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

cp sapi/fpm/php-fpm /usr/local/bin

1. 需要着重提醒的是，如果文件不存在，则阻止 Nginx 将请求发送到后端的 PHP-FPM 模块， 以避免遭受恶意脚本注入的攻击。

将 php.ini 文件中的配置项 [cgi.fix\_pathinfo](https://www.php.net/manual/zh/ini.core.php" \l "ini.cgi.fix-pathinfo) 设置为 *0* 。

打开 php.ini:

vim /usr/local/php/php.ini

定位到 *cgi.fix\_pathinfo=* 并将其修改为如下所示：

cgi.fix\_pathinfo=0

1. 在启动服务之前，需要修改php-fpm.conf 配置文件，确保 php-fpm 模块使用 www-data 用户和 www-data 用户组的身份运行。

1.首先打开以下文件：

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

2. 找到以下内容并修改：

; Unix user/group of processes

; Note: The user is mandatory. If the group is not set, the default user's group

; will be used.

user = www-data

group = www-data

(如果本操作系统不存在www-data用户，可使用nginx用户，已安装nginx此用户可用)如文件中不存在此上内容，请转入步骤3，否则修改完直接进入步骤4。

3.首先查看/usr/local/etc/php-fpm.conf，查看最后一行内容：

; - /usr/local otherwise

include=NONE/etc/php-fpm.d/\*.conf

修改为：include=/usr/local/etc/php-fpm.d/\*.conf

cd /usr/local/etc/php-fpm.d/

cp www.conf.default www.conf

然后打开/usr/local/etc/php-fpm.d/www.conf文件，修改第23-24行：

user = nginx

group = nginx

4.然后启动 php-fpm 服务：

/usr/local/bin/php-fpm

本文档未涵盖对 php-fpm 进行进一步配置的信息，比如可以把服务添加至系统服务。

打开文件/usr/local/etc/php-fpm.conf，将（第17行） ;pid = run/php-fpm.pid前面的分号去掉，然后执行：

# cp php-7.3.7/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

#chmod +x /etc/init.d/php-fpm

#systemctl start php-fpm

1. 配置 Nginx 使其支持 PHP 应用：

vim /etc/nginx/conf.d/default.conf

修改默认的 location 块(第9-10行)，使其支持 .php 文件：

location / {

root /usr/share/nginx/html; #注意此处为nginx的WEB默认路径

index index.php index.html index.htm;

}

下一步配置来保证对于 .php 文件的请求将被传送到后端的 PHP-FPM 模块，取消默认的 PHP 配置块的注释，并修改为下面的内容（第30-36行）：

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

include fastcgi\_params;

}

重启 Nginx。

systemctl status nginx

systemctl restart nginx

这里有个注意点，如果重启服务完成后，访问页面显示“file not found”，可以把以上内容（第34行）：

fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

修改成以下，将/scripts修改成nginx的WEB路径：

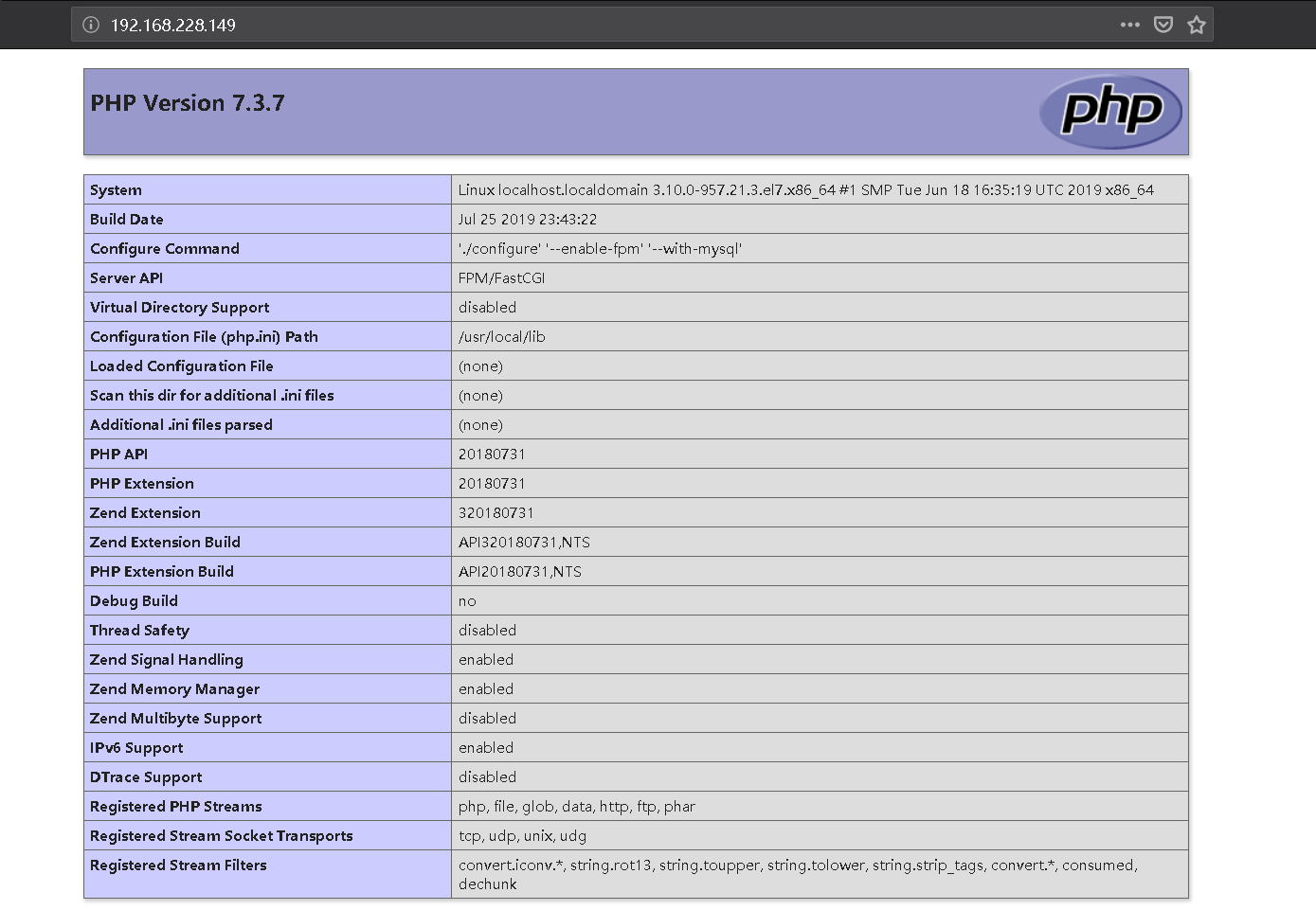
fastcgi\_param SCRIPT\_FILENAME ***/usr/share/nginx/html*** $fastcgi\_script\_name;

1. 创建测试文件。

rm /usr/share/nginx/html/index.html

echo "<?php phpinfo(); ?>" >> /usr/local/nginx/html/index.php

打开浏览器，访问 http://localhost，将会显示 phpinfo() 。



#### 源安装

也可以通过yum源直接安装，使用以下命令：

首先获取PHP7.0的yum源：

yum install epel-release

rpm -ivh http://rpms.famillecollet.com/enterprise/remi-release-7.rpm

安装扩展依赖包：

yum --enablerepo=remi install php73-php php73-php-pear php73-php-bcmath php73-php-pecl-jsond-devel php73-php-mysqlnd php73-php-gd php73-php-common php73-php-fpm php73-php-intl php73-php-cli php73-php php73-php-xml php73-php-opcache php73-php-pecl-apcu php73-php-pdo php73-php-gmp php73-php-process php73-php-pecl-imagick php73-php-devel php73-php-mbstring php73-php-zip php73-php-ldap php73-php-imap php73-php-pecl-mcrypt

（其中包含php-fpm）

启动php-fpm：

systemctl restart php73-php-fpm

systemctl enable php73-php-fpm

安装完成后，使用下面命令查看版本信息：

php73 –v

修改 /etc/nginx/conf.d/default.conf 文件，找到下面这段并改为如下所示：

location ~ \.php$ {

root /usr/share/nginx/html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

接着到 /usr/share/nginx/html 目录下创建一个index.php文件，内容为phpinfo():

vi /usr/share/nginx/html/test.php

#内容

<?php

phpinfo();

保存退出。接着重启nginx：

nginx -s reload

PHP官方文档：<https://www.php.net/manual/zh/install.unix.nginx.php>

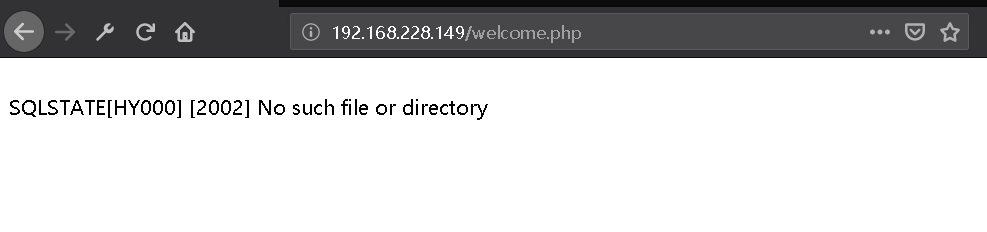
### PHP连接数据库

参考菜鸟教程，将以下使用 PDO 实例创建数据库 的代码保存为welcome.php:

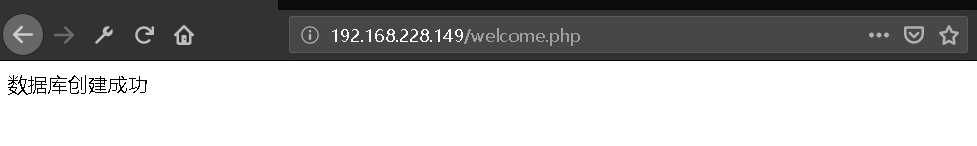
使用 PDO代码：

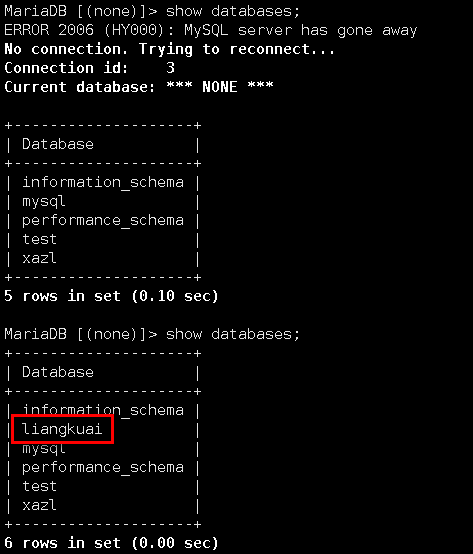
<?php   
$servername = "localhost";   
$username = "root";   
$password = "123456";   
  
try {   
    $conn = new PDO("mysql:host=$servername", $username, $password);   
  
    // 设置 PDO 错误模式为异常   
    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);   
    $sql = "CREATE DATABASE liangkuai";   
  
    // 使用 exec() ，因为没有结果返回   
    $conn->exec($sql);   
  
    echo "数据库创建成功<br>";   
}   
catch(PDOException $e)   
{   
    echo $sql . "<br>" . $e->getMessage();   
}   
  
$conn = null;   
?>

如果出现以下错误，可将 $servername = "localhost"; 修改为$servername = "127.0.0.1";



本PDO实例会连接数据库，并进入MySQL新建一个库名为：liangkuai。





参考资料：https://www.runoob.com/php/php-mysql-connect.html

## 集成包安装

[Wamp Server](http://www.wampserver.com/en/#download-wrapper)是功能强大的PHP集成安装环境，是在windows下将Apache+PHP+Mysql 集成安装的软件包，拥有简单的图形和菜单安装和配置环境。

[phpStudy](http://phpstudy.php.cn/)是PHP环境的程序集成包。该程序包集成Apache+PHP+MySQL+phpMyAdmin等一次性安装，无须配置即可使用，非常方便、好用。

[LAMP](https://lamp.sh/) 一键安装包是用 Linux Shell编写的可以为 Linux（CentOS/ Ubuntu）系统的服务器安装 LAMP(Linux + Apache + MySQL/MariaDB + PHP) 生产环境的 Shell 脚本。

# 安全加固

Windows与Linux作为主机服务器，虽然架构不同，但在安全加固着眼点和方法论层面是统一的。我认为安全加固可以从三个点入手：一是从防御者角度，着眼于具体的主机资产，关注操作系统和软件本身，针对其所有可配置项进行梳理，结合安全要求情况做最优化的配置，形成安全配置白名单，避免太多精力放在未知的威胁上面；二是从攻击者来看，对WEB服务、其所在主机及数据库软件，梳理所有攻击测试的方法及其能达到的危害效果，然后针对其提出具体的整改措施进行加固；三是以运营者观察，从攻防两面性上都是考虑安全性，运维层面必需要同时兼顾加固可操作性、运维便捷性、工具自动化及周期性改进等。

兼顾安全性与便利性，结合实际情况适当调整（力争三位一体立体环绕防御），一般分为用户策略（密码策略、账户安全）、本地策略（审核、用户权限、群组权限、安全选项设置）、日志审计、系统服务、文件系统、防火墙策略、网络设置、公钥设施策略（加密）、软件及应用程序管理、管理配置模板等方面的安全，具体操作表现为更改用户认证授权配置、添加审核策略、权限划分、访问控制、补丁升级、病毒防护、端口与服务管理、加密、审计日志及其他安全性配置。此处不展开了，可对WEB服务器及操作系统的安全加固具体操作项进行列表整理。