

## 三 服务器配置与管理

### 实验目的

1. 掌握 Linux 常用服务器安装方法；
2. 熟悉 Linux 常用服务器基本配置文件；
3. 理解 Linux 服务器管理思路。

### 实验环境

安装有 Linux 操作系统并连接互联网的计算机。

### 实验步骤

Linux 操作系统在服务器领域具有广泛的应用。Web 服务是服务器领域中应用最广泛的服务，常见 Web 服务有 WAMP（Windows+Apache+MySQL+PHP）和 LAMP（Linux+Apache+MySQL+PHP）架构，其中 Apache 是全世界最流行的 Web 服务软件，此外，Web 服务软件 nginx，因其具有优秀的性能也受到越来越多的用户欢迎。本实验中，演示在 Linux 环境下搭建支持 PHP 等网页的 Web 服务平台，即 LAMP。

Linux 环境下搭建 Web 服务器有三种方式，一是在安装操作系统时勾选相应服务组件；二是从网上下载或从 ISO 系统镜像包中拷贝安装包手动安装；三是在连接网络的情况下在线安装。

本实验演示以 ubuntu 64 位操作系统为例。

#### 1. 在线安装 Apache 服务器

如下图所示，ubuntu 可通过“apt”等命令在线安装，其它版本 Linux 请查询相关操作。

```
sudo apt-get install apache2
```

```

yql@kali:/$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 240 not upgraded.
Need to get 259 kB of archives.
After this operation, 622 kB of additional disk space will be used.
Get:1 http://mirrors.neusoft.edu.cn/kali kali-rolling/main amd64 apache2 amd64 2
.4.43-1 [259 kB]
Fetched 259 kB in 2s (134 kB/s)
Selecting previously unselected package apache2.
(Reading database ... 269026 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.43-1_amd64.deb ...
Unpacking apache2 (2.4.43-1) ...
Setting up apache2 (2.4.43-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for kali-menu (2020.2.2) ...
Processing triggers for systemd (245.4-3) ...

```

图 1 在线安装 Apache

安装完成后需要手动启动 apache 服务。

```
sudo /etc/init.d/apache2 start
```

```

yql@kali:/$ sudo /etc/init.d/apache2 start
Starting apache2 (via systemctl): apache2.service.

```

图 2 启动 Apache 服务

测试 Apache，在浏览器输入虚拟机 IP 地址，如下图所示，说明 Apache 安装成功。

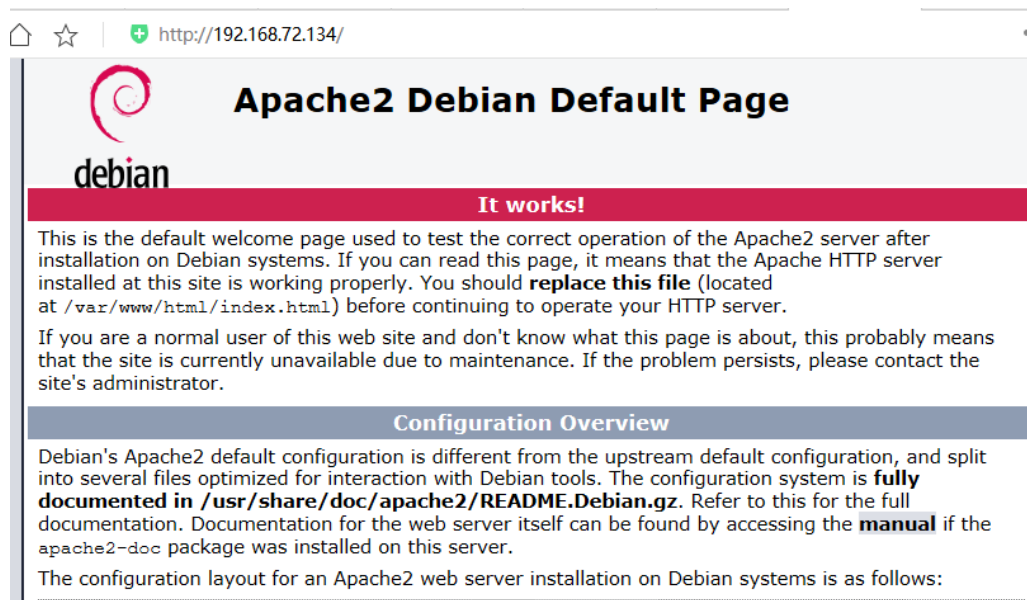


图 3 测试 Apache

测试页面详细介绍了 debian 发行版 Linux 中 Apache 基本信息，其中：

**Apache 根目录:**

/var/www/html

**Apache 配置目录和文件:**

/etc/apache2: Apache 主配置目录, Apache 所有配置文件均在此目录下;

/etc/apache2/apache2.conf: 主配置文件, 可配置 Apache 全局配置;

/etc/apache2/ports.conf: 端口配置文件。默认情况下, 当启用提供 SSL 功能的模块时, Apache 监听端口 80, 并在端口 443 上进行监听。

其它文件略。

**Apache 日志文件:**

/var/log/apache2/access.log: 服务请求日志;

/var/log/apache2/error.log: 服务错误日志。

**Apache 基本操作:**

服务启动: /etc/init.d/apache2 start

服务停止: /etc/init.d/apache2 stop

服务重启: /etc/init.d/apache2 restart

部分 Linux 安装 Apache 后可能出现服务正常运行, 相应端口正常打开, 但其它主机无法访问情况, 此现象一般是因为系统防火墙未开放相应服务端口。

## 2. 安装 MySQL/MariaDB

本示例采用 MariaDB 代替 MySQL。MariaDB 是 MySQL 的一个分支, 现由开源社区维护, 采用 GPL 授权许可, 其大部分语法与 MySQL 都相同。因 MySQL 被 Oracle 收购后有闭源的风险, 大部分 Linux 组织均从其套件清单删除了 MySQL, 并以 MariaDB 代替 MySQL, 若一定要使用 MySQL, 可通过下载安装包的方式手动安装 MySQL。

安装 MariaDB 服务

```
sudo apt-get install mariadb-server
```

```
sudo apt-get install mariadb-client
```

```
yql@kali:/$ sudo apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  figlet finger firebird3.0-common firebird3.0-common-doc freerdp2-x11 gdal-data
  libarmadillo9 libarpack2 libbson-1.0-0 libcfitsio8 libcharls2 libdap25
  libdapclient6v5 libepsilon1 libfbclient2 libfreerdp-client2-2 libfreerdp2-2
  libfreexl1 libfyba0 libgeotiff5 libhdf4-0-alt libhdf5-hl-100 libkmlbase1 libkmldom1
  libkmlengine1 libmemcached11 libmongoc-1.0-0 libnetcdf18 libodbc1 libogdi4.1
  libqhull7 libspatialite7 libssh-4 libsuperlu5 liburiparser1 libwinpr2-2
  libxerces-c3.2 medusa odbcinst odbcinst1debian2 python3-aiohttp python3-aioredis
  python3-ajpy python3-apscheduler python3-async-timeout python3-git python3-gitdb
  python3-multidict python3-pyexploitdb python3-pyfiglet python3-pyshodan
  python3-pysmi python3-pysnmp4 python3-quamash python3-smmap python3-tld python3-yarl
  python3-yaswfp rwho rwhod sparta-scripts toilet-fonts wapiti xsltproc
```

图 4 在线安装 MariaDB 服务

```
yql@kali:/$ sudo apt-get install mariadb-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  figlet finger firebird3.0-common firebird3.0-common-doc freerdp2-x11 gdal-data libarm
  libcfitsio8 libcharls2 libdap25 libdapclient6v5 libepsilon1 libfbclient2 libfreerdp-d
  libgeotiff5 libhdf4-0-alt libhdf5-hl-100 libkmlbase1 libkmldom1 libkmlengine1 libmemc
  libodbc1 libogdi4.1 libqhull7 libspatialite7 libssh-4 libsuperlu5 liburiparser1 libwi
  odbcinst1debian2 python3-aiohttp python3-aioredis python3-ajpy python3-apscheduler py
  python3-gitdb python3-multidict python3-pyexploitdb python3-pyfiglet python3-pyshodar
  python3-quamash python3-smmap python3-tld python3-yarl python3-yaswfp rwho rwhod spar
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  mariadb-client
0 upgraded, 1 newly installed, 0 to remove and 235 not upgraded.
Need to get 0 B/31.2 kB of archives.
After this operation, 67.6 kB of additional disk space will be used.
Selecting previously unselected package mariadb-client.
(Reading database ... 268714 files and directories currently installed.)
Preparing to unpack .../mariadb-client_1%3a10.3.22-1_all.deb ...
Unpacking mariadb-client (1:10.3.22-1) ...
Setting up mariadb-client (1:10.3.22-1) ...
```

图 5 在线安装 MariaDB 服务

配置 Mariadb 的安全选项:

```
sudo mysql_secure_installation
```

此时系统会提示输入数据库 root 用户密码, 因系统并未设置相应密码, 此时直接按提示回车即可。若提示错误, 则 `sudo /etc/init.d/mysql restart` 命令重启 mysql 服务并重新执行安全选项命令。

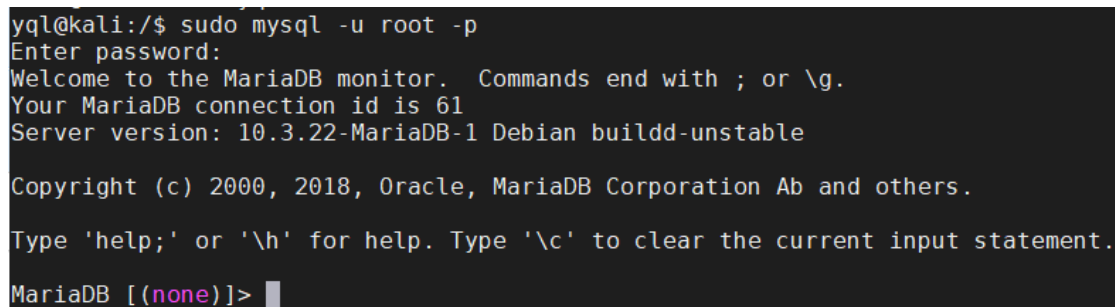
随后数据库会提示以下安全设置信息, 一般设置数据库 root 用户密码即可, 其它选项按回车选择默认:

- 1、Enter current password for root (enter for none): 输入当前 root 的密码(因新数据库无密码, 回车即可);
- 2、Set root password? [Y/n] 回车, 默认为输入 Y;
- 3、New password: 输入新密码;
- 4、Re-enter new password 确认密码;
- 5、Remove anonymous users? [Y/n] 移除匿名用户;

- 6、Disallow root login remotely? [Y/n] 禁止 root 远程登录;
- 7、Remove test database and access to it? [Y/n] 移除测试数据库;
- 8、Reload privilege tables now? [Y/n] 重新加载权限表。

测试数据库，如图所示，若进入数据库则表示数据库安装成功。

```
sudo mysql -u root -p
```



```
yql@kali:/$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 61
Server version: 10.3.22-MariaDB-1 Debian builddd-unstable

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)]>
```

图 6 测试数据库

数据库其它设置，如设置开机启动等，请自行参考相关资料。

### 3. 安装 PHP

安装 PHP 除了 PHP 应用程序外，还需安装 PHP 与 Apache、MySQL/MariaDB 相关扩展包，扩展包需与软件对应，本例中安装 php7.3 版本，对应扩展包可通过以下命令模糊查询。

```
sudo apt-cache search php7
```

从查询结果可知，php7.3 对应 Apache、MySQL 扩展包分别为：libapache2-mod-php7.3，php7.3-mysql。如下图所示为安装 PHP 相关软件包。

```
sudo apt-get install php7.3 libapache2-mod-php7.3 php7.3-mysql
```

```

yql@kali:/$ sudo apt-get install php7.3 libapache2-mod-php7.3 php7.3-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
libapache2-mod-php7.3 is already the newest version (7.3.15-3).
libapache2-mod-php7.3 set to manually installed.
php7.3 is already the newest version (7.3.15-3).
php7.3-mysql is already the newest version (7.3.15-3).
php7.3-mysql set to manually installed.
The following packages were automatically installed and are no longer required:
  figlet finger firebird3.0-common firebird3.0-common-doc freerdp2-x11 gdal-data
  libarmadillo9 libarpack2 libbson-1.0-0 libcfitsio8 libcharls2 libdap25
  libdapclient6v5 libepsilon1 libfbclient2 libfreerdp-client2-2 libfreerdp2-2
  libfreeex11 libfyba0 libgeotiff5 libhdf4-0-alt libhdf5-hl-100 libkmlbase1 libkmldom1
  libkmlengine1 libmemcached11 libmongoc-1.0-0 libnetcdf18 libodbc1 libogdi4.1
  libqhull7 libspatialite7 libssh-4 libsuperlu5 liburiparser1 libwinpr2-2
  libxerces-c3.2 medusa odbcinst odbcinstdebian2 python3-aiohttp python3-aioredis
  python3-ajpy python3-apscheduler python3-async-timeout python3-git python3-gitdb
  python3-multidict python3-pyexploitdb python3-pyfiglet python3-pyshodan
  python3-pysmi python3-pysnmp4 python3-quamash python3-smmmap python3-tld python3-yar
  python3-yaswfp rwho rwho sparta-scripts toilet-fonts wapiti xsltproc
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 235 not upgraded.

```

图 7 在线安装 PHP 及相应扩展包

PHP 安装完毕后需重启 Apache。

#### 4. 测试 PHP 页面


编辑测试文件，如下图所示，在“/var/www/html”目录下新建“test.php”文件，并输入如下图所示的测试代码。

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

图 8 编辑测试文件

在浏览器中输入测试页面地址，并确认已安装组件。

> ↻ ⌂ ☆ | + http://192.168.72.134/test.php

**PHP Version 7.3.15-3**


System	Linux kali 5.5.0-kali2-amd64 #1 SMP Debian 5.5.17-1kali1 (2020-04-21) x86_64
Build Date	Feb 23 2020 07:15:44
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqld.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini

图 9 测试 PHP

## 5. 创建文件上传页面

文件上传功能由上传文件的 HTML 表单和文件上传脚本构成。

在根目录下创建 “upload.html” 文件，编辑表单

---

```
<html>
<body>

<form action="upload.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>
```

---

在根目录下创建 upload.php 脚本文件，编写文件上传功能代码，并将上传文件保存至网站根目录下的 upload 目录中。

---

```
<?php
// 允许上传的图片后缀
$allowedExts = array("gif", "jpeg", "jpg", "png");
$temp = explode(".", $_FILES["file"]["name"]);
echo $_FILES["file"]["size"];
$extension = end($temp); // 获取文件后缀名
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/pjpeg")
|| ($_FILES["file"]["type"] == "image/x-png")
|| ($_FILES["file"]["type"] == "image/png")))
&& ($_FILES["file"]["size"] < 204800) // 小于 200 kb
&& in_array($extension, $allowedExts))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "错误: " . $_FILES["file"]["error"] . "<br>";
    }
    else
    {
        echo "上传文件名: " . $_FILES["file"]["name"] . "<br>";
        echo "文件类型: " . $_FILES["file"]["type"] . "<br>";
```

---

```
echo "文件大小: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
echo "文件临时存储的位置: " . $_FILES["file"]["tmp_name"] . "<br>";

// 判断当前目录下的 upload 目录是否存在该文件
if (file_exists("upload/" . $_FILES["file"]["name"]))
{
    echo $_FILES["file"]["name"] . " 文件已经存在。 ";
}
else
{
    // 如果 upload 目录不存在该文件则将文件上传到 upload 目录下
    move_uploaded_file($_FILES["file"]["tmp_name"], "upload/" . $_FILES["file"]["name"]);
    echo "文件存储在: " . "upload/" . $_FILES["file"]["name"];
}
}
else
{
    echo "非法的文件格式";
}
?>
```

在根目录下创建“upload”目录，用于保存上传的图片。

如下图所示，为上传文件页面。



图 10 上传文件页面

## 实验内容

1. 完成 LAMP 安装;
2. 编辑文件上传页面，服务器文件保存于/var/www/html2/upload，设置服务器使正常使用文件上传功能;
3. 修改默认端口至 8080，并将根目录设置为/var/www/html2。