

第17章 LAMP环境搭建

学习Linux请加QQ群： 群1(163262181) 群2(148412746) 群3(246401509) 群4(173884211)
跟阿铭学Linux邀请函 (<http://www.aminglinux.com>)，猿课已上线，请加微信aminglinux84索要配套视频教程。

经过前部分章节的学习，你已经掌握了linux的基础知识了。但是想成为一名系统管理员恐怕还有点难度，因为好多单位招聘这个职位的时候都要求有一定的工作经验。然而真正的经验一天两天是学不来的，是靠长时间积累得来的。不过你也不要灰心，所谓的工作经验无非也就是一些运行在linux系统上的软件的配置以及应用。就好像是装在windows上的office一样，大部分人都会装，但是十分会用的却不多。是因为office太难吗，当然不是，只是因为只有一小部分人花费了很长很长的时间去使用和研究office而已。

LAMP 是Linux Apache MySQL PHP的简写，其实就是把Apache，MySQL以及PHP安装在Linux系统上，组成一个环境来运行php的脚本语言。至于什么是php脚本语言，阿铭不介绍，请自己查资料吧。Apache是最常用的WEB服务软件，而MySQL是比较小型的数据库软件，这两个软件以及PHP都可以安装到windows的机器上。下面阿铭就教你如何构建这个LAMP环境。

安装MySQL

我们平时安装MySQL都是源码包安装的，但是由于它的编译需要很长的时间，所以，阿铭建议你安装二进制免编译包。你可以到MySQL官方网站去下载 <http://dev.mysql.com/downloads/> 具体版本根据你的平台和需求而定，目前比较常用的为mysql-5.0/mysql-5.1，5.5版本虽然已经发布有段时间了，但是貌似用在线上跑服务的还是少数。所以，阿铭建议你下载一个5.1的版本。可以使用阿铭提供的地址下载。下面是安装步骤(注意，下面的安装步骤是基于32位操作系统的，如果你的系统为64位，则需要下载一个64位的包)：

1. 下载mysql到/usr/local/src/

```
cd /usr/local/src/  
wget http://syslab.comsenz.com/downloads/linux/mysql-5.1.40-linux-i686-icc-glibc23.tar.gz
```

2. 解压

```
[root@localhost src]# tar zxvf /usr/local/src/mysql-5.1.40-linux-i686-icc-glibc23.tar.gz
```

3. 把解压完的数据移动到/usr/local/mysql

```
[root@localhost src]# mv mysql-5.1.40-linux-i686-icc-glibc23 /usr/local/mysql
```

4. 建立mysql用户

```
[root@localhost src]# useradd -s /sbin/nologin mysql
```

5. 初始化数据库

```
[root@localhost src]# cd /usr/local/mysql  
[root@localhost mysql]# mkdir -p /data/mysql ; chown -R mysql:mysql /data/mysql  
[root@localhost mysql]# ./scripts/mysql_install_db --user=mysql --datadir=/data/mysql
```

--user 定义数据库的所属主， --datadir 定义数据库安装到哪里，建议放到大空间的分区上，这个目录需要自行创建。这一步骤很关键，如果你看到两个“OK”说明执行正确，否则请仔细查看错误信息，如果你实在解决不了，请把问题发到论坛教程答疑版块(<http://www.aminglinux.com/bbs/forum-40-1.html>)阿铭会来帮你解决问题。

5. 拷贝配置文件

```
[root@localhost mysql]# cp support-files/my-large.cnf /etc/my.cnf
```

7. 拷贝启动脚本文件并修改其属性

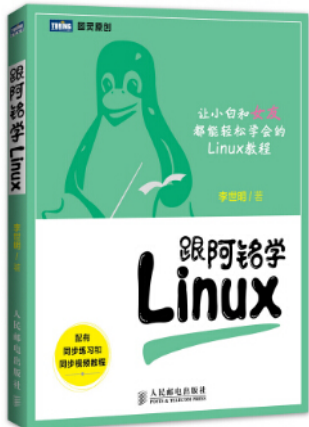
```
[root@localhost mysql]# cp support-files/mysql.server /etc/init.d/mysqld  
[root@localhost mysql]# chmod 755 /etc/init.d/mysqld
```

3. 修改启动脚本

目录列表

| |
|------------------------------|
| 第1章 前言 |
| 第2章 关于Linux的历史 |
| 第3章 对Linux系统管理员的建议 |
| 第4章 安装Linux操作系统 |
| 第5章 初步认识Linux |
| 第6章 Linux系统的远程登陆 |
| 第7章 Linux文件与目录管理 |
| 第8章 Linux系统用户及用户组管理 |
| 第9章 Linux磁盘管理 |
| 第10章 文本编辑工具vim |
| 第11章 文档的压缩与打包 |
| 第12章 安装RPM包或者安装源码包 |
| 第13章 学习 shell脚本之前的基础知识 |
| 第14章 正则表达式 |
| 第15章 shell脚本 |
| 第16章 linux系统日常管理 |
| 第17章 LAMP环境搭建 |
| 第18章 LNMP环境搭建 |
| 第19章 学会使用简单的MySQL操作 |
| 第20章 NFS服务配置 |
| 第21章 配置FTP服务 |
| 第22章 配置Squid服务 |
| 第23章 配置Tomcat |
| 第24章 配置Samba服务器 |
| 第25章 MySQL replication(主从)配置 |
| 结语 |

阿铭著作：



微信扫码获取最新版linux电子书和视频

```
[root@localhost mysql]# vim /etc/init.d/mysqld
```

需要修改的地方有“**datadir=/data/mysql**”（前面初始化数据库时定义的目录）

3. 把启动脚本加入系统服务项，并设定开机启动，启动**mysql**

```
[root@localhost mysql]# chkconfig --add mysqld
[root@localhost mysql]# chkconfig mysqld on
[root@localhost mysql]# service mysqld start
```

如果启动不了，请到 **/data/mysql/** 下查看错误日志，这个日志通常是主机名**.err**。检查**mysql**是否启动的命令为：

```
[root@localhost mysql]# ps aux |grep mysqld
```

安装Apache

同样**apache**也需要到官网下载合适的版本，目前使用较多的版本为**2.0**或者**2.2**阿铭建议下载**2.2**版本。**apache**官网下载地址：<http://www.apache.org/dyn/closer.cgi> 你也可以使用阿铭提供的地址下载。

```
[root@localhost mysql]# cd /usr/local/src/
[root@localhost src]# wget http://syslab.comsenz.com/downloads/linux/httpd-2.2.16.tar.gz
```

解压：

```
[root@localhost src]# tar zxvf httpd-2.2.16.tar.gz
```

配置编译参数：

```
[root@localhost src]# cd httpd-2.2.16
[root@localhost httpd-2.2.16]# ./configure \
--prefix=/usr/local/apache2 \
--with-included-apr \
--enable-so \
--enable-deflate=shared \
--enable-expire=shared \
--enable-rewrite=shared \
--with-pcre
```

--prefix 指定安装到哪里，**--enable-so** 表示启用**DSO** [\[1\]](#) **--enable-deflate=shared** 表示共享的方式编译**deflate**，后面的参数同理。如果这一步你出现了这样的错误：

```
error: mod_deflate has been requested but can not be built due to prerequisite failures
```

解决办法是：

```
yum install -y zlib-devel
```

为了避免在**make**的时候出现错误，所以最好是提前先安装好一些库文件：

```
yum install -y pcre pcre-devel apr apr-devel
```

编译：

```
[root@localhost httpd-2.2.16]# make
```

安装：

```
[root@localhost httpd-2.2.16]# make install
```

以上两个步骤都可以使用 **echo \$?** 来检查是否正确执行，否则需要根据错误提示去解决问题。

安装PHP

阿铭写这本教程时，**php**当前最新版本为**5.5**，相信大多网站还在跑着**5.2**甚至更老的版本，其实**5.2**版本的**php**很经典也很稳定，因为阿铭的公司一直在使用**5.2**版本，但是考虑到版本太老，难免会有些漏洞，所以建议你使用**5.3**或者**5.4**版本，**php**官方下载地址：<http://www.php.net/downloads.php>

下载**php**：

```
[rot@localhost httpd-2.2.16]# cd /usr/local/src
[root@localhost src]# wget http://am1.php.net/distributions/php-5.3.27.tar.gz
```

解压：

```
[root@localhost src]# tar zxf php-5.3.27.tar.gz
```

SEARCH

Go

Enter search terms or a module, class or function name.

配置编译参数:

```
[root@localhost src]# cd php-5.3.27
[root@localhost php-5.3.27]# ./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 \
--enable-soap \
--enable-gd-native-ttf \
--enable-mbstring \
--enable-sockets \
--enable-exif \
--disable-ipv6
```

在这一步, 阿铭遇到如下错误:

```
configure: error: xml2-config not found. Please check your libxml2 installation.
```

解决办法是:

```
yum install -y libxml2-devel
```

还有错误:

```
configure: error: Cannot find OpenSSL's <evp.h>
```

解决办法是:

```
yum install -y openssl openssl-devel
```

错误:

```
checking for BZip2 in default path... not found
configure: error: Please reinstall the BZip2 distribution
```

解决办法:

```
yum install -y bzip2 bzip2-devel
```

错误:

```
configure: error: png.h not found.
```

解决办法:

```
yum install -y libpng libpng-devel
```

错误:

```
configure: error: freetype.h not found.
```

解决办法:

```
yum install -y freetype freetype-devel
```

错误:

```
configure: error: mcrypt.h not found. Please reinstall libmcrypt.
```

解决办法:

```
rpm -ivh "http://www.aminglinux.com/bbs/data/attachment/forum/month_1211/epel-release-6-7.noarch.rpm"
yum install -y libmcrypt-devel
```

因为centos6.x 默认的yum源没有libmcrypt-devel 这个包, 只能借助第三方yum源。

编译:

```
[root@localhost php-5.3.27]# make
```

在这一步，你也许还会遇到诸多错误，没有关系，请仔细查看报错信息，解决办法很简单，就是装缺少的库。你可以把错误信息复制到google上搜一下，如果实在是解决不了，请到阿铭论坛(<http://www.aminglinux.com/bbs/forum-40-1.html>)发帖请教阿铭吧。

安装：

```
[root@localhost php-5.3.27]# make install
```

拷贝配置文件：

```
[root@localhost php-5.3.27]# cp php.ini-production /usr/local/php/etc/php.ini
```

apache结合php

Apache主配置文件为：/usr/local/apache2/conf/httpd.conf

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

找到：

```
AddType application/x-gzip .gz .tgz
```

在该行下面添加：

```
AddType application/x-httpd-php .php
```

找到：

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

将该行改为：

```
<IfModule dir_module>
    DirectoryIndex index.html index.htm index.php
</IfModule>
```

找到：

```
#ServerName www.example.com:80
```

修改为：

```
ServerName localhost:80
```

测试LAMP是否成功

启动apache之前先检验配置文件是否正确：

```
/usr/local/apache2/bin/apachectl -t
```

如果有错误，请继续修改httpd.conf，如果是正确的则显示为“Syntax OK”，启动apache的命令为：

```
/usr/local/apache2/bin/apachectl start
```

查看是否启动：

```
[root@localhost ~]# netstat -lnp |grep httpd
tcp        0      0 :::80          :::*   LISTEN    7667/httpd
```

如果有显示这行，则启动了。也可以使用curl命令简单测试：

```
[root@localhost ~]# curl localhost
<html><body><h1>It works!</h1></body></html>
```

只有显示这样才正确。

测试是否正确解析php：

```
vim /usr/local/apache2/htdocs/1.php
```

写入：

```
<?php
    echo "php解析正常";
?>
```

保存后，继续测试：

```
curl localhost/1.php
```

看是否能看到如下信息：

```
[root@localhost ~]# curl localhost/1.php
php解析正常[root@localhost ~]#
```

只有显示如阿铭这样才正确。

初次使用浏览器访问我们的web服务的时候，你可能无法访问，这是因为防火墙的缘故。请运行下面的命令：

```
[root@localhost ~]# iptables -F
```

这样就可以清除系统默认的防火墙规则，放行80端口。

LAMP环境是搭建好了，这其实仅仅是安装上了软件而已，而具体的配置还是有很多工作要做的呢？也就是说，你虽然搭建出来了环境，但是如果不会配置细节的东西，相当于没有任何工作经验，所以还是多配置配置apache或者php吧，具体参考资料可以到阿铭论坛的相应版本中找到，大多帖子为阿铭工作中所配置过的，阿铭真心希望你能够按照阿铭的帖子配置一下，这样对你有很大的好处。论坛地址：

<http://www.aminglinux.com/bbs/forum.php>

阿铭建议你最好再扩展学习一下：<http://www.aminglinux.com/bbs/thread-5441-1-1.html>

教程答疑：[请移步这里](#)。

欢迎你加入 [阿铭学院](#) 和阿铭一起学习Linux，让阿铭成为你Linux生涯中永远的朋友吧！

[1]DSO是Dynamic Shared Objects（动态共享目标）的缩写，它提供了一种在运行时将特殊格式的代码在程序运行需要时，将需要的部分从外存调入内存执行的方法。Apache 支持动态共享模块，也支持静态模块，静态的话，会把需要的目标直接编译进apache的可执行文件中，相比较动态，虽然省去了加载共享模块的步骤，但是也加大了二进制执行文件的空间，变得臃肿。

[PREVIOUS](#) | [NEXT](#) | [INDEX](#)

© Copyright 2013, lishiming.net. Created using [Sphinx](#) 1.3b1[网站统计](#) .