

web 平台搭建-LAMP（CentOS-7）

一. 准备工作

环境要求：

操作系统：CentOS 7.X 64 位

网络配置：nmtui 字符终端图形管理工具或者直接编辑配置文件

关闭 SELinux 和 firewalld 防火墙

防火墙：

临时关闭：systemctl stop firewalld

永久关闭：systemctl disable firewalld

1. 安装编译工具 gcc、gcc-c++等

注意解决依赖关系，推荐使用 yum 安装，若不能联网可使用安装光盘做为 yum 源

a. 编辑 yum 配置文件，启用本地光盘源（只有一张盘）

```
# mount /dev/sr0 /mnt
# vim /etc/yum.repos.d/CentOS-Media.repo
[c7-media]
name=CentOS-$releasever - Media
baseurl=file:///mnt
gpgcheck=0
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

b. 调整 yum 源配置文件引导优先级

```
# mv /etc/yum.repos.d/CentOS-Base.repo /backup
```

c. 安装 gcc、gcc-c++、make 等编译工具

```
# yum -y install gcc gcc-c++ make
```

2. 关闭系统 RPM 安装包的 Apache、MySQL 等服务

为了防止 rpm 安装的软件和接下来安装的源码软件包冲突

```
# systemctl stop httpd
# systemctl stop mysqld
# .....
```

确定 rpm 包安装的 httpd 和 mysqld 不能开机自启动

```
# systemctl disable httpd
# systemctl disable mysqld
# .....
```

3. 关闭 SELinux 和 firewalld

防止软件安装和调试过程被 firewalld 和 SELinux 所限制，无法实现效果

a. 关闭 SELinux（需重启）

```
# vim /etc/selinux/config
```

```
SELINUX=disabled
# reboot
```

b. 关闭 firewalld

```
# iptables -F
# systemctl disable firewalld
# reboot
```

4. 拷贝源码包，解包解压缩

建议将 LAMP 环境安装源码包统一存放在一个目录下，如 /lamp，可以使用解压脚本解压缩

注意：使用 Xshell 上传时传到 /tmp 下，/root 目录无法上传

```
# vim tar.sh
cd /lamp
/bin/ls *.tar.gz > ls.list
for TAR in `cat ls.list`
do
    /bin/tar -xf $TAR
done
/bin/rm ls.list
```

5. 查看安装软件的磁盘空间是否充足

保证软件能正常安装，空间不足时会导致软件安装失败

```
# df -h
```

6. 源码软件包安装报错确认与解决方案

```
echo $?          #安装软件过程中由于频繁刷屏，建议在每个步骤结束后执行此命令
./configure      #此步骤报错多是依赖关系没解决或是编译工具未安装（注意关键词提示）
make             #此步骤多是编译时选项参数书写错误、不存在、漏写等问题
                #一般需要检查上一个步骤：./configure --help
```

注意：若遇到报错，最简答的办法是，找到问题解决后重新解压软件，重新安装，步骤最简洁

二. 编译安装

注意：每个源码包配置编译安装完成后，确认安装目录下是否生成安装文件（并确定目录是否正确）

建议将安装路径指定为[--prefix=/usr/local/软件名]格式

1. 安装 libxml2

```
# yum install -y libxml2-devel python-devel
# cd /lamp/libxml2-2.9.1
# ./configure --prefix=/usr/local/libxml2/
# make
# make install
```

2. 安装 libmcrypt

```
# cd /lamp/libmcrypt-2.5.8
# ./configure --prefix=/usr/local/libmcrypt/
# make
# make install
```

安装 libltdl, 也在 libmcrypt 源码目录中, 非新软件

```
# cd /lamp/libmcrypt-2.5.8/libltdl
# ./configure --enable-ltdl-install
# make
# make install
```

3. 安装 mhash

```
# cd /lamp/mhash-0.9.9.9
# ./configure
# make
# make install
```

4. 安装 mcrypt

```
# cd /lamp/mcrypt-2.6.8
# export LD_LIBRARY_PATH=/usr/local/libmcrypt/lib:/usr/local/lib
变量: LD_LIBRARY_PATH 用于指定 libmcrypt 和 mhash 的库的位置
# ./configure --with-libmcrypt-prefix=/usr/local/libmcrypt
# make
# make install
```

5. 安装 zlib

```
# cd /lamp/zlib-1.2.3
# ./configure
然后修改配置文件, 否则无法正常安装此软件
# vi Makefile
CFLAGS=-O3 -DUSE_MMAP -fPIC
#找到 CFLAGS=-O3 -DUSE_MMAP, 在后面加入 -fPIC 变成 (注意: 小 f 大 PIC, 空格)
# make
# make install
```

6. 安装 libpng

```
# cd /lamp/libpng-1.2.31
# ./configure --prefix=/usr/local/libpng
```

```
# make
# make install
```

7. 安装 jpeg6

```
# mkdir /usr/local/jpeg6
# mkdir /usr/local/jpeg6/bin
# mkdir /usr/local/jpeg6/lib
# mkdir /usr/local/jpeg6/include
# mkdir -p /usr/local/jpeg6/man/man1
```

注意：此软件默认不会自动创建所需目录，所以目录必须手工建立

```
# yum -y install libtool*
# cd /lamp/jpeg-6b
# cp -a /usr/share/libtool/config/config.sub ./
# cp -a /usr/share/libtool/config/config.guess ./
复制 libtool 中的文件，覆盖 jpeg-6b 中的文件（64 位中的问题）
# ./configure --prefix=/usr/local/jpeg6/ --enable-shared --enable-static
# make
# make install
```

--enable-shared 与 --enable-static 参数分别为建立共享库和静态库使用的 libtool

8. 安装 freetype

```
# cd /lamp/freetype-2.3.5
# ./configure --prefix=/usr/local/freetype/
# make
# make install
```

9. 安装 Apache

a. 源码包 2.4.* 版本中默认没有集成 apr 的依赖包，所以需要提前解决依赖问题

```
# cp -a /lamp/apr-1.4.6 /lamp/httpd-2.4.7/src/lib/apr
# cp -a /lamp/apr-util-1.4.1 /lamp/httpd-2.4.7/src/lib/apr-util
```

解压 apr 和 apr-util，复制整个目录并取消目录上的版本号到指定位置，./configure 时会检测

b. Apache 默认需要依赖 pcre 软件，但由于 Apache 软件版本较高，则系统预安装的 pcre 无法使用，所以需要人为手动安装适合版本

```
# cd /lamp/pcre-8.34
# ./configure
# make
# make install
```

c. Apache 的加密传输模块 mod_ssl，需要安装此软件产生

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

d. httpd 软件安装

```
# cd /lamp/httpd-2.4.7
# ./configure --prefix=/usr/local/apache2 --sysconfdir=/usr/local/apache2/etc
  --with-included-apr --enable-so --enable-deflate=shared --enable-expire=shared
  --enable-rewrite=shared --enable-ssl
# make
# make install
```

若前面配置 zlib 时没有指定安装目录, Apache 配置时不要添加 `--with-z=/usr/local/zlib/` 参数, `--enable-ssl` 选项是为了后期实现 https 提前设置的参数

e. 启动 Apache 测试

```
# /usr/local/apache2/bin/apachectl start
# ps aux | grep httpd
使用进程查看命令确认 Apache 是否启动, 是否产生进程
# netstat -tln | grep :80
使用网络进程查看命令确认 Apache 是否启动, 是否开启了 80 监听端口
```

注意事项: 在 CentOS 7 操作系统上 Apache 默认监听了 Ipv6 地址的 80 端口, 监听 Ipv4 的地址。
Listen 80

报错提示: 若启动时提示 `/usr/local/apache2/modules/mod_deflate.so` 无权限, 可关闭 SELinux 解决, 类似此类 .so 文件不能载入或没有权限的问题, 都是 SELinux 问题, MySQL 和 Apache 都可能有类似问题。

警告提示: 发现启动服务提示: `AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally to suppress this message`

解决办法: 打开主配置文件 `httpd.conf`

搜索 `ServerName` (约在 200 行左右)

改为 `ServerName localhost:80` (并且去掉前面的#注释)

验证: 通过浏览器输入地址访问: `http://服务器 ip`, 若显示 “It works” 即表明 Apache 正常工作

10. 安装 ncurses

```
# yum -y install ncurses-devel
# cd /lamp/ncurses-5.9
# ./configure --with-shared --without-debug --without-ada --enable-overwrite
# make
# make install
```

若不安装 ncurses 编译 MySQL 时会报错

11. 安装 cmake 和 bison

```
yum -y install cmake bison
```

12. 安装 MySQL

```
# useradd -r -s /sbin/nologin mysql
```

为 MySQL 软件创建运行用户，创建为系统用户，并限制此用户登录操作系统

```
# cd /lamp/mysql-5.5.48
```

```
# cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DMYSQL_UNIX_ADDR=/tmp/mysql.sock  
-DEXTRA_CHARSETS=all -DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci  
-DWITH_MYISAM_STORAGE_ENGINE=1 -DWITH_INNOBASE_STORAGE_ENGINE=1  
-DWITH_MEMORY_STORAGE_ENGINE=1 -DWITH_READLINE=1 -DENABLED_LOCAL_INFILE=1  
-DMYSQL_USER=mysql -DMYSQL_TCP_PORT=3306
```

```
# make
```

```
# make install
```

选项详解：

-DCMAKE_INSTALL_PREFIX=/usr/local/mysql	安装位置
-DMYSQL_UNIX_ADDR=/tmp/mysql.sock	指定 socket（套接字）文件位置
-DEXTRA_CHARSETS=all	扩展字符支持
-DDEFAULT_CHARSET=utf8	默认字符集
-DDEFAULT_COLLATION=utf8_general_ci	默认字符校对
-DWITH_MYISAM_STORAGE_ENGINE=1	安装 myisam 存储引擎
-DWITH_INNOBASE_STORAGE_ENGINE=1	安装 innodb 存储引擎
-DWITH_MEMORY_STORAGE_ENGINE=1	安装 memory 存储引擎
-DWITH_READLINE=1	支持 readline 库
-DENABLED_LOCAL_INFILE=1	启用加载本地数据
-DMYSQL_USER=mysql	指定 mysql 运行用户
-DMYSQL_TCP_PORT=3306	指定 mysql 端口

MySQL 安装后需要调整相应配置文件和参数才能正常运行

a. 修改 MySQL 目录的用户归属

```
# cd /usr/local/mysql/
```

```
# chown -R root .
```

```
# chown -R mysql data
```

b. 生成配置文件，并初始化授权表

```
# cp -a /lamp/mysql-5.5.48/support-files/my-medium.cnf /etc/my.cnf
```

复制 MySQL 配置文件到指定位置，覆盖掉系统自带文件

```
# cd /usr/local/mysql
```

```
# ./scripts/mysql_install_db --user=mysql
```

创建数据库授权表，初始化数据库，相当于安装完操作系统后的引导设置（添加第一个用户）

报错提示：FATAL ERROR: Could not find ./bin/my_print_defaults

原因：mysql_install_db 初始化所调用文件时使用的是相对路径，路径不在/usr/local/mysql 时，

是无法调用 `my_print_defaults` 文件并初始化成功的。

c. 启动 MySQL 服务

用原本源代码的方式去使用和启动 `mysql`

```
/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

d. 设定 MySQL 密码

```
/usr/local/mysql/bin/mysqladmin -uroot password 123456
```

e. 登录 MySQL

```
# /usr/local/mysql/bin/mysql -u root -p
mysql>show databases;
mysql>use test;
mysql>show tables;
mysql>exit
```

13. 安装 PHP

```
# cd /lamp/php-7.0.7
# ./configure --prefix=/usr/local/php/ --with-config-file-path=/usr/local/php/etc/
--with-apxs2=/usr/local/apache2/bin/apxs --with-libxml-dir=/usr/local/libxml2/
--with-jpeg-dir=/usr/local/jpeg6/ --with-png-dir=/usr/local/libpng/
--with-freetype-dir=/usr/local/freetype/ --with-mcrypt=/usr/local/libmcrypt/
--with-mysqli=/usr/local/mysql/bin/mysqli_config --enable-soap --enable-mbstring=all
--enable-sockets --with-pdo-mysql=/usr/local/mysql --with-gd --without-pear
# make
# make install
```

选项详解:

<code>--with-config-file-path=/usr/local/php/etc/</code>	指定配置文件目录
<code>--with-apxs2=/usr/local/apache2/bin/apxs</code>	指定 apache 动态模块位置
<code>--with-libxml-dir=/usr/local/libxml2/</code>	指定 libxml 位置
<code>--with-jpeg-dir=/usr/local/jpeg6/</code>	指定 jpeg 位置
<code>--with-png-dir=/usr/local/libpng/</code>	指定 libpng 位置
<code>--with-freetype-dir=/usr/local/freetype/</code>	指定 freetype 位置
<code>--with-mcrypt=/usr/local/libmcrypt/</code>	指定 libmcrypt 位置
<code>--with-mysqli=/usr/local/mysql/bin/mysqli_config</code>	指定 mysqli 位置
<code>--with-gd</code>	启用 gd 库
<code>--enable-soap</code>	支持 soap 服务
<code>--enable-mbstring=all</code>	支持多字节, 字符串
<code>--enable-sockets</code>	支持套接字
<code>--with-pdo-mysql=/usr/local/mysql</code>	启用 mysql 的 pdo 模块支持
<code>--without-pear</code>	不安装 pear (安装 pear 需要连接互联网)

PHP 安装后需要调整相应配置文件和参数才能正常运行

a. 生成 php 配置文件

```
# mkdir /usr/local/php/etc
# cp /lamp/php-7.0.7/php.ini-production /usr/local/php/etc/php.ini
```

b. 修改 Apache 配置文件，使其识别*.php 文件，并能通过 php 模块调用 php 进行页面解析

```
# vim /usr/local/apache2/etc/httpd.conf
AddType application/x-httpd-php .php .html
AddType application/x-httpd-php-source .phps
```

重启 Apache 服务

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

c. 测试 php 页面是否能正常解析（即 apache 和 php 连通性）

```
# vim /usr/local/apache2/htdocs/test.php
<?php
    phpinfo();
?>
```

通过浏览器输入地址访问：http://Apache 服务器地址/test.php

14. 为 PHP 安装 openssl 模块

```
# cd /lamp/php-7.0.7/ext/openssl
# mv config0.m4 config.m4
# /usr/local/php/bin/phpize
# ./configure --with-openssl --with-php-config=/usr/local/php/bin/php-config
# make
# make install
```

15. 为 PHP 安装 memcache 模块

```
# unzip pecl-memcache-php7.zip
# cd /lamp/pecl-memcache-php7
# /usr/local/php/bin/phpize
# ./configure --with-php-config=/usr/local/php/bin/php-config
# make
# make install
```

16. 修改 php 配置文件，使其识别并调用 openssl 和 memcache 两个模块

```
# vi /usr/local/php/etc/php.ini
extension_dir="/usr/local/php/lib/php/extensions/no-debug-zts-20151012/"
取消分号注释，并添加以上路径（此路径来自于模块安装命令的结果）
extension="openssl.so";
```



```
extension="memcache.so";
```

添加以上两个库文件的调用

重启 apache，刷新 phpinfo 页面，并查看是否有两个新增的模块

17. 安装 memcached 服务

```
# wget
ftp://ftp.pbone.net/mirror/ftp.centos.org/7.6.1810/os/x86_64/Packages/libevent-devel-2.0.21-4.el7.x86_64.rpm
# yum -y install libevent-devel
# cd /lamp/memcached-1.4.17
# ./configure --prefix=/usr/local/memcache
# make
# make install
```

```
# useradd -r -s /sbin/nologin memcache
添加 memcache 用户，此用户不用登录，不设置密码
# /usr/local/memcache/bin/memcached -umemcache &
启动 memcache 服务，并设置为后台运行
# netstat -an | grep :11211
检查 memcache 是否正常启动，并监听了 11211 端口
```

18. 安装 phpMyAdmin

```
# cp -a /lamp/phpMyAdmin-4.1.4-all-languages /usr/local/apache2/htdocs/phpmyadmin
# cd /usr/local/apache2/htdocs/phpmyadmin
# cp -a config.sample.inc.php config.inc.php
# vim config.inc.php
$config['Servers'][$i]['auth_type'] = 'cookie';
$config['Servers'][$i]['auth_type'] = 'http';
设置 auth_type 为 http，即设置为 HTTP 身份认证模式（新增即可）
```

通过浏览器输入地址访问：[http://Apache 服务器地址/phpmyadmin/index.php](http://Apache服务器地址/phpmyadmin/index.php)

用户名为 root，密码为 MySQL 设置时指定的 root 密码 123456

19. 设置 Apache、MySQL、Memcache 开机自启

借助系统自带脚本/etc/rc.local，此脚本开机后会自动加载，我们可以将源码安装的服务启动命令写入该脚本，间接实现开机自启动

```
# vi /etc/rc.local
/usr/local/apache2/bin/apachectl start
/usr/local/mysql/bin/mysqld_safe --user=mysql &
/usr/local/memcache/bin/memcached -umemcache &
```

20. 项目迁移：

- 1、把 php 项目拷贝到网站默认目录下：/usr/local/apache2/htdocs/**
- 2、使用 phpMyAdmin 创建网站所需数据库

注意事项：注意目录权限和归属，防止权限过大或者权限过小

切记：做完 LAMP 环境后保存一个快照，后面讲 Apache 要使用！