

## CentOS7配置LAMP环境（PHP7.2.1+MySQL5.6.39+Apache2.4.29）源码安装超详细步骤

最近出于对linux服务器的学习，在安装配置的过程中遇到了各种问题，也上网看了许多相关文章，现在终于弄好了，特写此教程向大家分享，也为自己留下学习笔记。

注：如果系统之前已配置过LAMP运行环境，记得先将之前的版本卸载，防止安装过程中出现各种莫名其妙的问题，本教程为全新的CentOS7最小化安装版。

### 【服务器配置】

配置网络参数，为服务器分配静态ip方便远程登录管理。



```
1 [root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-ens33
2 TYPE="Ethernet"
3 BOOTPROTO="static"
4 IPADDR="192.168.43.171"
5 NETMASK="255.255.255.0"
6 GATEWAY="192.168.43.1"
7 NM_CONTROLLED=no
8 DEFROUTE="yes"
9 IPV4_FAILURE_FATAL="no"
10 IPV6INIT="yes"
11 IPV6_AUTOCONF="yes"
12 IPV6_DEFROUTE="yes"
13 IPV6_FAILURE_FATAL="no"
14 IPV6_ADDR_GEN_MODE="stable-privacy"
15 NAME="ens33"
16 UUID="f52d1397-768e-426c-9a29-9b80600cb181"
17 DEVICE="ens33"
18 ONBOOT="yes"
```



重启网络服务

```
1 [root@localhost ~]# systemctl restart network.service
```

此时可通过 **ip addr** 命令查看当前网络配置，然后 **ping** 网关是通的, **ping** 百度域名www.baidu.com又不通，原因在于没有配置dns

修改配置文件，添加dns参数

```
1 [root@localhost ~]# vi /etc/NetworkManager/NetworkManager.conf
2 dns=none
```

修改resolv.conf,添加dns配置

```
1 [root@localhost ~]# vi /etc/resolv.conf
2 nameserver 192.168.43.1
```

此时再次 **ping** 百度域名www.baidu.com就能正常返回数据了。

开放防火墙端口（SSH远程连接22端口）

```
1 [root@localhost ~]# firewall-cmd --zone=public --add-port=22/tcp --permanent
```

重启防火墙服务

```
1 [root@localhost ~]# systemctl restart firewalld.service
```

安装 vim

```
1 [root@localhost ~]# yum install vim -y
```

安装 wget

```
1 [root@localhost ~]# yum install wget -y
```

## 【安装Apache】

下载 apache 及相关组件 apr、apr-util、pcre

```
1 [root@localhost ~]# wget http://mirrors.tuna.tsinghua.edu.cn/apache//httpd/httpd-2.4.29.tar.gz
2 [root@localhost ~]# wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-1.6.3.tar.gz
3 [root@localhost ~]# wget http://mirrors.tuna.tsinghua.edu.cn/apache//apr/apr-util-1.6.1.tar.gz
4 [root@localhost ~]# wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.41.tar.gz
```

安装编译所需工具及依赖库

```
1 [root@localhost ~]# yum install gcc gcc++ gcc-c++ libxml2 libxml2-devel expat-devel -y
```

解包并做相关配置



```
1 [root@localhost ~]# tar -zxvf httpd-2.4.29.tar.gz
2 [root@localhost ~]# tar -zxvf apr-1.6.3.tar.gz
3 [root@localhost ~]# tar -zxvf apr-util-1.6.1.tar.gz
4 [root@localhost ~]# tar -zxvf pcre-8.41.tar.gz
5 [root@localhost ~]# mv apr-1.6.3 apr
6 [root@localhost ~]# mv apr-util-1.6.1 apr-util
7 [root@localhost ~]# mv apr apr-util httpd-2.4.29/src/lib
```



安装 pcre

```
1 [root@localhost ~]# cd pcre-8.41
2 [root@localhost pcre-8.41]# ./configure --prefix=/usr/local/pcre8
```

```
3 [root@localhost pcre-8.41]# make
4 [root@localhost pcre-8.41]# make install
```

开始 **apache** 编译配置，完成后执行安装

```
1 [root@localhost ~]# cd httpd-2.4.29
2 [root@localhost httpd-2.4.29]# ./configure --prefix=/usr/local/apache2 --with-
pcre=/usr/local/pcre8/bin/pcre-config --with-included-apr
3 [root@localhost httpd-2.4.29]# make
4 [root@localhost httpd-2.4.29]# make install
```

开放防火墙端口并重启防火墙服务（**HTTP**协议**80**端口）

```
1 [root@localhost ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
2 [root@localhost ~]# systemctl restart firewalld.service
```

添加 **apache** 开机启动服务



```
1 [root@localhost ~]# cp /usr/local/apache2/bin/apachectl /etc/init.d/httpd
2 [root@localhost ~]# vim /etc/init.d/httpd
3
4 在在#!/bin/sh后面加入下面两行
5 #chkconfig:345 85 15
6 #description: Start and stops the Apache HTTP Server.
7
8 [root@localhost ~]# chmod +x /etc/rc.d/init.d/httpd
9 [root@localhost ~]# chkconfig --add httpd
10 [root@localhost ~]# chkconfig httpd on
```



查看服务是否添加成功，启动 **apache** 服务

```
1 [root@localhost ~]# chkconfig --list httpd
2 [root@localhost ~]# systemctl start httpd.service
```

此时在浏览器中输入对应**ip**地址，将显示如下提示，表示**Apache**已经开始正常工作

## It works!

设置**Apache**支持**PHP**



```
1 [root@localhost ~]# vim /usr/local/apache2/conf/httpd.conf
2 找到<IfModule unixd_module> 在该行前面添加 LoadModule php7_module
modules/libphp7.so
3 按图片-1位置添加内容
4 AddType application/x-httpd-php .php
5 AddType application/x-httpd-php-source .php5
6
```

```
7 接着找到
8 <IfModule dir_module>
9     DirectoryIndex index.html
10 </IfModule>
11 修改为
12 <IfModule dir_module>
13     DirectoryIndex index.html index.php
14 </IfModule>
15
16 找到
17 DocumentRoot "/usr/local/apache2/htdocs"
18 <Directory "/usr/local/apache2/htdocs">
19 修改为
20 DocumentRoot "/home/www"
21 <Directory "/home/www">
22 此处 /home/www 为你自定义的网站根目录，设置效果如图片-2
```



```
# If the AddEncoding directives above are commented-out, then
# probably should define those extensions to indicate media t
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .php5
#
# AddHandler allows you to map certain file extensions to "han
# actions unrelated to filetype. These can be either built in
# or added with the Action directive (see below)
```

图片-1

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/home/www"
<Directory "/home/www">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
```

图片-2

保存退出后创建网站根目录

```
1 [root@localhost ~]# mkdir /home/www
```

重启Apache服务

```
1 [root@localhost ~]# systemctl restart httpd.service
```

注意：此时Apache重启后仍然无法解析php，因为 modules/libphp7.so 文件不存在，当安装完PHP后就能正常解析。

【安装PHP】

## 下载 php 并解包

```
1 [root@localhost ~]# wget http://am1.php.net/get/php-7.2.1.tar.gz/from/this/mirror
2 [root@localhost ~]# tar -zxvf mirror
```

切换至解包好的目录并执行编译配置（--enable-fpm 是为了配合能使用 nginx）

```
1 [root@localhost ~]# cd php-7.2.1/
2 [root@localhost php-7.2.1]# ./configure --prefix=/usr/local/php7 --enable-fpm --with-
apxs2=/usr/local/apache2/bin/apxs
```

编译配置完成后执行编译，最后执行安装

```
1 [root@localhost php-7.2.1]# make
2 [root@localhost php-7.2.1]# make install
```

此时重启Apache后即可在 /home/www 目录中自行创建php文件进行测试访问

## 【安装MySQL】

查找并卸载 mariadb

```
1 [root@localhost ~]# rpm -qa | grep mariadb
2 [root@localhost ~]# rpm -e --nodeps 包名
```

进行安装前准备工作

```
1 [root@localhost ~]# groupadd mysql
2 [root@localhost ~]# useradd -r -g mysql mysql
3 [root@localhost ~]# mkdir -p /data/mysql
4 [root@localhost ~]# mkdir /usr/local/mysql
5 [root@localhost ~]# mkdir /var/lib/mysql
```

下载 mysql 并解包( 5.7.5以上版本需安装对应 boost)

```
1 [root@localhost ~]# wget https://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-
5.6.39.tar.gz
2 [root@localhost ~]# tar -zxvf mysql-5.6.39.tar.gz
```

安装编译工具及依赖

```
1 [root@localhost ~]# yum install cmake gcc-c++ ncurses-devel perl-Data-Dumper -y
```

执行编译及安装



```
1 [root@localhost ~]# cd mysql-5.6.39/
2 [root@localhost mysql-5.6.39]# cmake \
3 -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \
4 -DMYSQL_DATADIR=/data/mysql \
5 -DMYSQL_USER=mysql \
6 -DSYSCONFDIR=/etc \
```

```
7 -DWITH_MYISAM_STORAGE_ENGINE=1 \  
8 -DWITH_INNOBASE_STORAGE_ENGINE=1 \  
9 -DWITH_MEMORY_STORAGE_ENGINE=1 \  
10 -DWITH_READLINE=1 \  
11 -DMYSQL_UNIX_ADDR=/var/lib/mysql/mysql.sock \  
12 -DMYSQL_TCP_PORT=3306 \  
13 -DENABLED_LOCAL_INFILE=1 \  
14 -DWITH_PARTITION_STORAGE_ENGINE=1 \  
15 -DEXTRA_CHARSETS=all \  
16 -DDEFAULT_CHARSET=utf8 \  
17 -DDEFAULT_COLLATION=utf8_general_ci  
18 [root@localhost mysql-5.6.39]# make  
19 [root@localhost mysql-5.6.39]# make install
```



ps: 编译安装的步骤耗时会比较长，耐心等待一会 ^\_^

设置目录所有者和组

```
1 [root@localhost ~]# chown -R mysql:mysql /usr/local/mysql  
2 [root@localhost ~]# chown -R mysql:mysql /data/mysql  
3 [root@localhost ~]# chown -R mysql:mysql /var/lib/mysql
```

初始化 mysql 数据库

```
1 [root@localhost ~]# cd /usr/local/mysql  
2 [root@localhost mysql]# scripts/mysql_install_db --user=mysql --datadir=/data/mysql
```

此时会在 /usr/local/mysql 目录下生成一个 my.cnf 的配置文件

设置 mysql 配置文件



```
1 [root@localhost ~]# cp /usr/local/mysql/my.cnf /etc/my.cnf  
2 [root@localhost ~]# vim /etc/my.cnf  
3 如果直接使用这份配置文件，也可不执行第一条复制命令，直接执行第二条命令新建 my.cnf 配置文件  
4 以下为 my.cnf 配置文件的内容  
5  
6 # For advice on how to change settings please see  
7 # http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html  
8  
9 [client]  
10 port = 3306  
11 socket = /var/lib/mysql/mysql.sock  
12  
13 [mysqld]  
14  
15 # Remove leading # and set to the amount of RAM for the most important data  
16 # cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.  
17 # innodb_buffer_pool_size = 128M  
18
```

```
19 # Remove leading # to turn on a very important data integrity option: logging
20 # changes to the binary log between backups.
21 # log_bin
22
23 # These are commonly set, remove the # and set as required.
24 basedir = /usr/local/mysql
25 datadir = /data/mysql
26 port = 3306
27 server_id = 1
28 socket = /var/lib/mysql/mysql.sock
29
30 # Remove leading # to set options mainly useful for reporting servers.
31 # The server defaults are faster for transactions and fast SELECTs.
32 # Adjust sizes as needed, experiment to find the optimal values.
33 join_buffer_size = 128M
34 sort_buffer_size = 2M
35 read_rnd_buffer_size = 2M
36
37 sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```



将 mysql 执行命令添加到PATH

```
1 [root@localhost ~]# vim /etc/profile
2 在文件末尾添加以下内容
3 PATH=$PATH:/usr/local/mysql/bin
4 export PATH
5 保存退出后执行以下命令
6 [root@localhost ~]# source /etc/profile
```

将 mysql 添加为系统服务并设置开机启动

```
1 [root@localhost ~]# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysqld
2 [root@localhost ~]# chkconfig --add mysqld
3 [root@localhost ~]# chkconfig mysqld on
4 [root@localhost ~]# chkconfig --list mysqld
```

启动 mysql 服务

```
1 [root@localhost ~]# service mysqld start
```

此时如果出现以下错误

```
1 Starting MySQL... ERROR! The server quit without updating PID file
(/data/mysql/localhost.localdomain.pid).
```

处理错误状况



```
1 [root@localhost ~]# vim /etc/init.d/mysqld
2 找到以下位置
```

```
3 basedir=  
4 datadir=  
5 修改为对应的路径  
6 basedir='/usr/local/mysql'  
7 datadir='/data/mysql'  
8 保存退出后执行以下命令  
9 [root@localhost ~]# chown -R mysql:mysql /var/lib/mysql
```



重新启动 **mysql** 服务

```
1 [root@localhost ~]# service mysqld start  
2 Starting MySQL. SUCCESS!
```

此时已提示 **mysql** 启动成功

设置 **mysql** 密码

```
1 [root@localhost ~]# mysqladmin -u root -p password  
2 Enter password:此处因为没原密码，直接回车  
3 New password:输入新密码  
4 Confirm new password:再次输入确认密码
```

登录 **mysql**

```
1 [root@localhost ~]# mysql -uroot -p  
2 Enter password:输入密码，回车
```

设置远程登录用户

```
1 远程登录用户名为root，密码为1234  
2 mysql>GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '1234' WITH GRANT OPTION;  
3 mysql>flush privileges;
```

开放防火墙端口并重启防火墙服务（MySQL服务3306端口）

```
1 [root@localhost ~]# firewall-cmd --zone=public --add-port=3306/tcp --permanent  
2 [root@localhost ~]# systemctl restart firewalld.service
```

此时就可以通过其他计算机远程连接到此服务器的MySQL数据库啦~

源码安装就是比较复杂点，现在整个服务器的LAMP运行环境已全部安装完毕 ^\_^

值得注意的是MySQL的安装比较复杂易错，注意目录所属用户和组的设置

分类: [Linux](#)