**CentOS 7.3下配置 Apache2.4 + MySQL5.7 + PHP7.1.8**

源文档：<https://www.linuxidc.com/Linux/2017-08/146220.htm>

注：弘德誉曦

本篇文章讲解的是在[CentOS](http://www.linuxidc.com/topicnews.aspx?tid=14)7.3下配置 Apache2.4 + MySQL5.7 + PHP7.2.2

注:安装前升级yum源

#yum update

1. 安装Apache2.4
2. 查看httpd包是否可用

#yum list | grep httpd

1. 安装Apache

#yum install httpd

1. 配置servername

#vim /etc/httpd/conf/httpd.conf

修改这行：ServerName localhost:80

1. 启动

#systemctl start httpd

一般来说新安装的软件都会启动成功。如果启动失败，请注意错误信息。可用

#ps –aux | grep httpd #命令检查进程是否被占用。如果被占用

#kill -9 进程号 #把httpd的进程杀干净再启动

1. 设置开机启动(httpd服务需要先运行起)

#chkconfig httpd on

二. 安装MySQL

1. 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. 启动MySQL

#systemctl start mysqld

1. 获取密码

#grep ‘temporary password’ /var/log/mysqld.log

得到这行A temporary password is generated for root@localhost: Jqqskhz1Wr(? 冒号后面的就是密码)

1. 进入MySQL

#mysql –uroot –p

1. 修改密码

>ALTER USER ‘root’@’localhost’ IDENTIFIED BY ‘新密码’; #注意MySQL的密码必须复杂 不复杂会报错

1. 开启远程访权限

>use mysql;

>update user set host = ‘%’ where user = ‘root’; #百分号相当于\*号，意为全部放行，也可改为IP地址则只允许此IP连接，也可以设置为192.168.%.%或者192.168.0.1/9代表允许一个ip段进行连接，也可以多加几条数据设置不同ip允许连接。

1. MySQL权限管理

>create user ‘myqiutian’@’%’ IDENTIFIED BY ‘\*\*\*\*\*\*\*’; #创建一个用户为myqiutian，因为用的%所以任何IP都可以登录，但登录后却无法看到数据库，新增用户所有权限默认关闭。

1. 分配访问权限

>grant all on \*.\* to myqiutian; #设置用户myqiutian拥有所有数据库所有表的所有权限。

>grant select on A数据库.\* to myqiutian; #授权myqiutian这个用户可以查看A数据库里的所有表，但是仅限于查看。

>grant insert on A数据库.user to myqiutian; #授权 myqiutian这个用户可以对A数据库里的user表进行insert操作，也仅限于select 和 insert 操作。

撤销权限

>revoke insert on A数据库.user from myqiutian;

注意：撤销权限之前最好用show grants for myqiutian; 这条sql查一下该用户有哪些权限，增加的什么权限就撤销什么权限，你增加的insert就不能撤销all。

MySQL权限工作流程：

--->用户连接MySQL

--->查询user表 核对账号密码 检查host字段 是否允许你的ip进行连接

--->查询user表 其他权限字段 值若为Y代表用户对所有数据库所有表所有字段都拥有该权限，若为N则往下走

--->查询db表 库权限掏表 获取 该用户对哪 些库拥有哪些权限，这张表里一条记录代表一个库

--->查询tables\_priv表 获取该用户的表控制权限，同样，如果db表中该用户对A数据库拥有insert权限，那么不管tables\_priv表中如何设置，都不会影响该用户的insert权限，如果tables\_priv表中的单表权限不为all，则继续往下走。

--->查询columns\_priv表 字段控制 可以设置对表字段的控制权限

四个表依次为：user -> db -> tables\_priv -> columns\_riv

注意：

如果user表中全为Y，那么不会查询下面的表。

user表中的select为N时，可以在db表中指定哪些库可以被用户看到。

但是无论是表控制还是字段控制，上级权限表（user表除外）的select字段必须为Y。

而其他权限如果上级表给出了设置，那么不会采用下级表的设置。

具体权限字段请查阅官方手册：

[http://www.yesky.com/imagesnew/software/mysql/manual\_Privilege\_system.html#Privileges\_provided](http://www.yesky.com/imagesnew/software/mysql/manual_Privilege_system.html%23Privileges_provided)

1. 刷新权限 立即生效

>flush privileges

三．安装php7

1. yum源没有最新版本。可在线下载：

#wget <http://hk2.php.net/get/php-7.2.2.tar.gz/from/this/mirror> #官网下载php-7.2.2.tar.gz

1. 解压

#tar -zxvf php-7.2.2.tar.gz

1. 进入解压包安装一些属性的依赖

#cd php-7.2.2

# yum -y install libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel curl curl-devel openssl openssl-devel

1. 安装gcc

#yum install gcc

1. 安装php

#yum -y install libxslt-devel\*

#yum -y install perl\*

#yum -y install httpd-devel

#find / -name apxs #得到的路径是：/usr/bin/apxs

于是得到 –with-apxs2 是的路径是：/usr/bin/apxs

1. 配置

#./configure \

--prefix=/usr/local/php7 \

--with-config-file-path=/etc \

--enable-fpm \

--with-fpm-user=nginx \

--with-fpm-group=nginx \

--enable-inline-optimization \

--disable-debug \

--disable-rpath \

--enable-shared \

--enable-soap \

--with-libxml-dir \

--with-xmlrpc \

--with-openssl \

--with-mcrypt \

--with-mhash \

--with-pcre-regex \

--with-sqlite3 \

--with-zlib \

--enable-bcmath \

--with-iconv \

--with-bz2 \

--enable-calendar \

--with-curl \

--with-cdb \

--enable-dom \

--enable-exif \

--enable-fileinfo \

--enable-filter \

--with-pcre-dir \

--enable-ftp \

--with-gd \

--with-openssl-dir \

--with-jpeg-dir \

--with-png-dir \

--with-zlib-dir \

--with-freetype-dir \

--enable-gd-native-ttf \

--enable-gd-jis-conv \

--with-gettext \

--with-gmp \

--enable-json \

--enable-mbstring \

--enable-mbregex \

--enable-mbregex-backtrack \

--with-libmbfl \

--with-onig \

--enable-pdo \

--with-mysqli=mysqlnd \

--with-pdo-mysql=mysqlnd \

--with-pdo-sqlite \

--with-readline \

--enable-session \

--enable-shmop \

--enable-simplexml \

--enable-sockets \

--enable-sysvmsg \

--enable-sysvsem \

--enable-sysvshm \

--enable-wddx \

--with-xsl \

--enable-zip \

--enable-mysqlnd-compression-support \

--with-pear \

--enable-opcache \

--with-iconv-dir \

--with-kerberos \

--with-libdir=lib64 \

--with-mysqli \

--with-pdo-mysql \

--enable-pcntl \

--enable-xml \

--with-apxs2=/usr/bin/apxs

1. 编译

#make

1. 编译出现错误请查阅：

博客：<http://www.cnblogs.com/sweetXiaoma/p/5855732.html>

本地文档

1. 编译检查

#make test

注：这步可走可不走，因为它老是提示错误，但又不给错误信息。如果有问题，那么编译的时候一定不成功，既然编译成功了，那么可以直接进行安装。

1. 安装

#make install

1. 添加环境变量

#vim /etc/profile

在末尾加入：

PATH=$PATH:/usr/local/php7/bin

export PATH

1. 使用改动立即生效

#source /etc/profile

1. 查看php版本

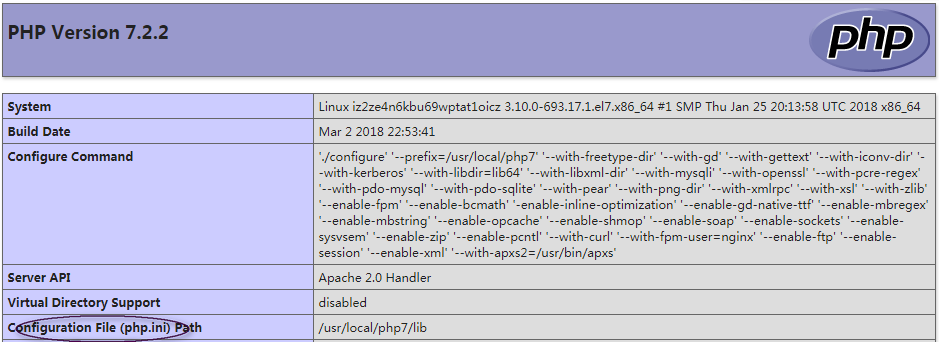
#php -v

#php –i #查看php安装信息

注：如果有问题谱表检查添加 的环境变量是否是php安装目录里的bin目录

1. 生成必要文件

查看



#cp php.ini-production /usr/local/php7/lib/php.ini

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

#cp /usr/local/php7/etc/php-fpm.conf.default /usr/local/php7/etc/php-fpm.conf

#cp /usr/local/php7/etc/php-fpm.d/www.conf.default /usr/local/php7/etc/php-fpm.d/www.conf

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

#/etc/init.d/php-fpm start #启动php-fpm

1. 安装扩展

方法一：用pecl安装

1.安装扩展之前先用pear设置php.ini的路径：

/usr/local/php7/bin/pear config-set php\_ini /usr/local/php7/etc/php.ini

这样做的好处就是用pecl安装php扩展的时候，会自动在php.ini中引用扩展包，不用自己手动

2.以安装redis扩展为例：

#/usr/local/php7/bin/pecl install channel://pecl.php.net/redis-3.1.4

#/usr/local/php7/bin/pecl install channel://pecl.php.net/oauth 2.0.0

方法二：源码编译安装

1. 安装oauth扩展

#wget http://pecl.php.net/get/oauth-2.0.0.tgz

#tar -zxf oauth-2.0.0.tgz

#cd oauth-2.0.2

#/usr/local/php/bin/phpize

#./configure --with-php-config=/usr/local/php7/bin/php-config

#make && make install

#/usr/local/php7/lib/php.ini # 添加 extension=oauth.so

#/etc/init.d/php-fpm restart #或 systemctl restart httpd

1. 如果报错，执行以下命令可以查到哪个文件第几行出错

#systemctl status httpd.service

1. 修改Apache默认欢迎页

#vim /etc/httpd/conf.d/welcome.conf

将/usr/share/httpd/noindex 修改为 /var/www

1. 修改Apache配置

#vim /etc/httpd/conf/httpd.conf

DocumentRoot “/var/www/”

(请注意，/var/www这个路径是自定义，在配置文件中有好几处这个路径，如果更改，请全局搜索一下都改掉)

找到

AddType application/x-compress .Z

AddType application/x-gzip .gz .tgz

在后面添加

AddType application/x-httpd-php .php

AddType application/x-httpd-php-source .php7

搜索<IfModule dir\_module>下面这一块添加上index.php

<IfModule dir\_module>

DirectoryIndex index.html index.php

</IfModule>

搜索有没有下面这一行：

LoadModule php7\_module modules/libphp7.so

如果没有 请手动添加 否则 会出现运行php文件变成下载

在最下面配置域名

<VirtualHost \*:80>

DocumentRoot /var/www

ServerName www.你的域名.com

ServerAlias 你的域名.com

<Directory /phpstudy/www>

Options +Indexes +FollowSymLinks +ExecCGI

AllowOverride All

Order Deny,Allow

Allow from all

</Directory>

</VirtualHost>

1. 测试

在www目录下创建index.php

添加<?php phpinfo();?>

访问:www.你的域名.com

四．安装redis

提示：安装Redis需要知道自己需要哪个版本，有针对性的安装，比如如果需要redis GEO这个地理集合的特性，那么redis版本就不能低于3.2版本，由于这个特性是3.2版本才有的。另外需要注意的是，Redis约定次版本号（即第一个小数点后的数字）为偶数的版本是稳定版（如2.8版、3.0版），奇数版本是非稳定版（如2.7版、2.9版），生产环境下一般需要使用稳定版本。

1. 下载安装包

官网：<https://redis.io/download>

#wget http://download.redis.io/releases/redis-4.0.8.tar.gz

1. 解压安装包并安装

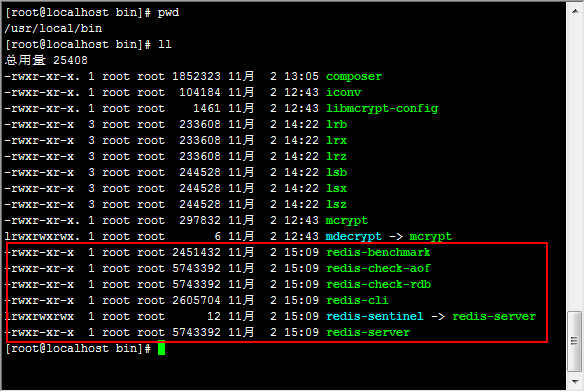
#tar –zxvf redis-4.0.8.tar.gz

#cd redis-4.0.8

#make

#make install

提示：Redis 没有其他自外部依赖。编译后在Redis源代码目录的src文件夹中可以找到若干个可执行的程序。安装赛后，在/usr/local/bin目录中可以找到刚刚安装的redis可执行文件。如：



1. 启动Redis
2. 直接运行redis-server即可启动Redis

#redis-server

1. 通过初始化脚本启动

在Redis源代码的utils文件夹中有一个名为redis\_init\_script的初始化脚本文件。需要配置Redis的运行方式和持久化文件、日志文件的存储位置。步骤如下：

1. 配置初始化脚本

首先将初始化脚本复制到/etc/init.d目录中，文件名为redis\_端口号，其中端口号表示要让Redis监听的端口号，客户端通过该端口连接Redis.然后修改脚本的REDISPORT变量为同样的端口号。

1. 建立以下需要的文件夹

#mkdir /etc/redis #用于存放Redis的配置文件

#mkdir /var/redis/端口号 #用于存放Redis的持久化文件

1. 修改配置文件

首先将配置文件模板（redis-4.0.8/redis.conf）复制到/etc/redis目录中，以端口号命名（如“6379.conf”），然后按照下表对其中的部分参数 进行编辑

参数 值 说明

daemonize yes 使Redis以守护进程模式运行

pidfile /var/run/redis\_端口号.pid 设置Redis的PID文件位置

port 端口号 设置Redis监听的端口号

dir /var/redis/端口号 设置持久化文件存放的位置

注：配置redis持久化，下次启动系统或redis时，缓存数据不会丢失。

1. 启动关闭Redis

通过以上配置，可以使用以下命令启动关闭Redis

#/etc/init.d/redis\_6379 start

#/etc/init.d/redis\_6379 stop

1. 设置开机启动
2. 需要对Redis初始化脚本进行简单修改，执行命令：

#vim /etc/init.d/redis\_6379

在打开的redis初始化脚本文件头部第四行的位置，追加下面两句，保存退出。

# chkconfig: 2345 90 10

# description: Redis is a persistent key-value database

1. 将redis加入系统启动项里

#chkconfig redis\_6379 on

1. 通过上面的操作后，以后也可以直接用下面的命令对redis进行启动和关闭：

#service redis\_6379 start

#service redis\_6379 stop

4. 停止Redis

考虑到 Redis 有可能正在将内存中的数据同步到硬盘中，强行终止 Redis 进程可能会导致数据丢失。正确停止Redis的方式应该是向Redis发送SHUTDOWN命令，方法为：

#redis-cli shutdown #或kill redis

当Redis收到SHUTDOWN命令后，会先断开所有客户端连接，然后根据配置执行持久化，最后完成退出。

Redis可以妥善处理 SIGTERM信号，所以使用 kill Redis 进程的 PID也可以正常结束Redis，效果与发送SHUTDOWN命令一样。

五． 安装MongoDB

1. 创建MongoDB安装目录

#mkdir –p /usr/local/mongodb

2. 下载与CentOS系统匹配的MongoDB数据库文件

官网：<https://www.mongodb.com/download-center#community>

#wget <https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-3.6.3.tgz>

3. 解压缩mongodb-linux-x86\_64-rhel70-3.6.3.tgz

#tar -zxvf mongodb-linux-x86\_64-rhel70-3.6.3

4. 移动并重命名

#mv -rf ./mongodb-linux-x86\_64-rhel70-3.6.3 /usr/local/mongodb/mongodb-3.6.3

5. 创建存在日志和数据文件的目录

#mkdir -p /usr/local/mongodb/data/logs

#mkdir -p /usr/local/mongodb/data/db

6. 创建配置文件

#vim /usr/local/mongodb/mongodb.conf

文件内容如下：

# idae - MongoDB config start - 2018-03-09

#设置数据文件的存放目录

dbpath=/usr/local/mongodb/data/db

#设置日志文件的存放目录及其日志文件名

logpath=/usr/local/mongodb/data/logs/mongodb.log

#设置端口号（默认端口号：27017）

port=27017

#设置以守护进程的方式运行

fork=true

# idae - MongoDB config end - 2018-03-09

7. 启动mongodb服务

1）以自定义的mongodb配置文件方式启动

# /usr/local/mongodb/mongodb-3.6.3/bin/mongod --config /usr/local/mongodb/mongodb.conf

2）以修复模式启动

# /usr/local/mongodb/mongodb-3.6.3/bin/mongod --repair -f /usr/local/mongodb/mongodb.conf

3）以参数式启动

# /usr/local/mongodb/mongodb-3.6.3/bin/mongod --dbpath=/usr/local/mongodb/data/db --logpath=/usr/local/mongodb/data/logs/mongodb.log --fork

8. 查看mobgodb进程

# ps aux | grep mongodb

9. 相看mongodb服务的运行日志

# tail -200f /usr/local/mongodb/data/logs/mongodb.log

10. 杀死mongodb进程，即可关闭mongodb服务

# kill -15 PID

PID可以通过步骤9查看到

11. 将mongodb服务加入到自启动文件中

# /etc/re.local

在文件末尾追加如下命令：

/usr/local/mongodb/mongodb-3.6.3/mongod --config /usr/local/mongodb/mongodb.conf

mongodb.conf 参数解释:

--dbpath 数据库路径(数据文件)

--logpath 日志文件路径

--master 指定为主机器

--slave 指定为从机器

--source 指定主机器的IP地址

--pologSize 指定日志文件大小不超过64M.因为resync操作量大且耗时，最好通过设置一个足够大的oplogSize来避免resync(默认的 oplog大小是空闲磁盘大小的5%)。

--logappend 日志文件末尾添加，即以追加的方式写日志

--journal 启用日志

--port 启用端口号

--fork 在后台运行

--only 指定只复制哪一个数据库

--slavedelay 指从复制检测的时间间隔

--auth 是否需要验证权限登录(用户名和密码)

--syncdelay 数据写入硬盘的时间（秒），0是不等待，直接写入

--notablescan 不允许表扫描

--maxConns 最大的并发连接数，默认2000

--pidfilepath 指定进程文件，不指定则不产生进程文件

--bind\_ip 绑定IP，绑定后只能绑定的IP访问服务

**linux环境apache的启停**

如果当前用户的paache没有安装为linux的服务，可以使用以下命令进行以上操作。

1. 查看Apache2的安装信息

# httpd -V

其中HTTPD\_ROOT和SERVER\_CONFIG\_FILE 就可以确定httpd.conf的路径了

假设当前Linux用户的apahce安装目录为：/usr/local/apache2

1. 启动 apache的命令：

#/usr/local/apache2/bin/apachectl start apache

1. 停止 apache的命令：

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

1. 重启apache的命令：

#/usr/local/apache2/bin/atachectl restart

1. 平滑启动apache的命令

#/usr/local/sbin/apachectl graceful

如果当前用户的paache已经安装为linux的服务，可以使用以下命令进行以上操作。

service httpd start|stop|restart

**linux编译安装时常见错误解决办法**

This article is post on https://coderwall.com/p/ggmpfa

原文链接：http://www.bkjia.com/PHPjc/1008013.html

configure: error: xslt-config not found. Please reinstall the libxslt >= 1.1.0 distribution

解决方案如下:

yum -y install libxslt-devel

configure: error: Could not find net-snmp-config binary. Please check your net-snmp installation.

解决方案如下:

yum -y install net-snmp-devel

configure: error: Please reinstall readline - I cannot find readline.h

解决方案如下:

yum -y install readline-devel

configure: error: Cannot find pspell

解决方案如下:

yum -y install aspell-devel

checking for unixODBC support... configure: error: ODBC header file '/usr/include/sqlext.h' not found!

解决方案如下:

yum -y install unixODBC-devel

configure: error: Unable to detect ICU prefix or /usr/bin/icu-config failed. Please verify ICU install prefix and make sure icu-config works.

解决方案如下:

yum -y install libicu-devel

configure: error: utf8mime2text() has new signature, but U8TCANONICAL is missing. This should not happen. Check config.log for additional information.

解决方案如下:

yum -y install libc-client-devel

configure: error: freetype.h not found.

解决方案如下:

yum -y install freetype-devel

configure: error: xpm.h not found.

解决方案如下:

yum -y install libXpm-devel

configure: error: png.h not found.

解决方案如下:

yum -y install libpng-devel

configure: error: vpx\_codec.h not found.

解决方案如下:

yum -y install libvpx-devel

configure: error: Cannot find enchant

解决方案如下:

yum -y install enchant-devel

configure: error: Please reinstall the libcurl distribution - easy.h should be in /include/curl/

解决方案如下:

yum -y install libcurl-devel

LAOGAO added 20140907：

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

解决方案如下:

wget ftp://mcrypt.hellug.gr/pub/crypto/mcrypt/libmcrypt/libmcrypt-2.5.7.tar.gz

tar zxf libmcrypt-2.5.7.tar.gz

cd libmcrypt-2.5.7

./configure

make && make install

added 20141003：

Cannot find imap

解决方案如下:

ln -s /usr/lib64/libc-client.so /usr/lib/libc-client.so

configure: error: utf8\_mime2text() has new signature, but U8T\_CANONICAL is missing.

解决方案如下:

yum -y install libc-client-devel

yum install libc-client-devel.x86\_64

ln -s /usr/lib64/libc-client.so /usr/lib/libc-client.so

Cannot find ldap.h

解决方案如下:

yum -y install openldap

yum -y install openldap-devel

configure: error: Cannot find ldap libraries in /usr/lib

解决方案如下:

cp -frp /usr/lib64/libldap\* /usr/lib/

configure: error: Cannot find libpq-fe.h. Please specify correct PostgreSQL installation path

解决方案如下:

yum -y install postgresql-devel

configure: error: Please reinstall the lib curl distribution

解决方案如下:

yum -y install curl-devel

configure: error: Could not find net-snmp-config binary. Please check your net-snmp installation.

解决方案如下:

yum -y install net-snmp-devel

configure: error: xslt-config not found. Please reinstall the libxslt >= 1.1.0 distribution

解决方案如下:

yum -y install libxslt-devel

checking for BZip2 support… yes checking for BZip2 in default path… not found configure: error: Please reinstall the BZip2 distribution

解决方案如下:

yum -y install bzip2-devel

checking for cURL support… yes checking if we should use cURL for url streams… no checking for cURL in default path… not found configure: error: Please reinstall the libcurl distribution – easy.h should be in/include/curl/

解决方案如下:

yum -y install curl-devel

checking for curl\_multi\_strerror in -lcurl… yes checking for QDBM support… no checking for GDBM support… no checking for NDBM support… no configure: error: DBA: Could not find necessary header file(s).

解决方案如下:

yum -y install db4-devel

checking for fabsf… yes checking for floorf… yes configure: error: jpeglib.h not found.

解决方案如下:

yum -y install libjpeg-devel

checking for fabsf… yes checking for floorf… yes checking for jpeg\_read\_header in -ljpeg… yes configure: error: png.h not found.

解决方案如下:

yum -y install libpng-devel

checking for png\_write\_image in -lpng… yes If configure fails try –with-xpm-dir=

configure: error: freetype.h not found.

解决方案如下:

Reconfigure your PHP with the following option. --with-xpm-dir=/usr

checking for png\_write\_image in -lpng… yes configure: error: libXpm.(a|so) not found.

解决方案如下:

yum -y install libXpm-devel

checking for bind\_textdomain\_codeset in -lc… yes checking for GNU MP support… yes configure: error: Unable to locate gmp.h

解决方案如下:

yum -y install gmp-devel

checking for utf8\_mime2text signature… new checking for U8T\_DECOMPOSE… configure: error: utf8\_mime2text() has new signature, but U8T\_CANONICAL is missing. This should not happen. Check config.log for additional information.

解决方案如下:

yum -y install libc-client-devel

checking for LDAP support… yes, shared checking for LDAP Cyrus SASL support… yes configure: error: Cannot find ldap.h

解决方案如下:

yum -y install openldap-devel

checking for mysql\_set\_character\_set in -lmysqlclient… yes checking for mysql\_stmt\_next\_result in -lmysqlclient… no checking for Oracle Database OCI8 support… no checking for unixODBC support… configure: error: ODBC header file ‘/usr/include/sqlext.h' not found!

解决方案如下:

yum -y install unixODBC-devel

checking for PostgreSQL support for PDO… yes, shared checking for pg\_config… not found configure: error: Cannot find libpq-fe.h. Please specify correct PostgreSQL installation path

解决方案如下:

yum -y install postgresql-devel

checking for sqlite 3 support for PDO… yes, shared checking for PDO includes… (cached) /usr/local/src/php-5.3.7/ext checking for sqlite3 files in default path… not found configure: error: Please reinstall the sqlite3 distribution

解决方案如下:

yum -y install sqlite-devel

checking for utsname.domainname… yes checking for PSPELL support… yes configure: error: Cannot find pspell

解决方案如下:

yum -y install aspell-devel

checking whether to enable UCD SNMP hack… yes checking for default\_store.h… no

checking for kstat\_read in -lkstat… no checking for snmp\_parse\_oid in -lsnmp… no checking for init\_snmp in -lsnmp… no configure: error: SNMP sanity check failed. Please check config.log for more information.

解决方案如下:

yum -y install net-snmp-devel

checking whether to enable XMLWriter support… yes, shared checking for xml2-config path… (cached) /usr/bin/xml2-config checking whether libxml build works… (cached) yes checking for XSL support… yes, shared configure: error: xslt-config not found. Please reinstall the libxslt >= 1.1.0 distribution

解决方案如下:

yum -y install libxslt-devel

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

解决方案如下:

yum -y install libxml2-devel

checking for PCRE headers location… configure: error: Could not find pcre.h in /usr

解决方案如下:

yum -y install pcre-devel

configure: error: Cannot find MySQL header files under yes. Note that the MySQL client library is not bundled anymore!

解决方案如下:

yum -y install mysql-devel

checking for unixODBC support… configure: error: ODBC header file ‘/usr/include/sqlext.h' not found!

解决方案如下:

yum -y install unixODBC-devel

checking for pg\_config… not found configure: error: Cannot find libpq-fe.h. Please specify correct PostgreSQL installation path

解决方案如下:

yum -y install postgresql-devel

configure: error: Cannot find pspell

解决方案如下:

yum -y install pspell-devel

configure: error: Could not find net-snmp-config binary. Please check your net-snmp installation.

解决方案如下:

yum -y install net-snmp-devel

configure: error: xslt-config not found. Please reinstall the libxslt >= 1.1.0 distribution

解决方案如下:

yum -y install libxslt-devel

error:virtual memory exhausted: Cannot allocate memory

当安装虚拟机时系统没有设置swap大小或设置内存太小

解决方案如下:

用swap扩展内存：

# free –m #查看cannot allocate memory信息

# mkdir /opt/images/

# rm -rf /opt/images/swap

# dd if=/dev/zero of=/opt/images/swap bs=1024 count=4096000

# mkswap /opt/images/swap

# swapon /opt/images/swap

# free –m

# swapoff swap #使用完毕后可以关掉（可选）

# rm –f /opt/images/swap # swap文件也可以不删除，留着以后使用，关键是你的虚拟机硬盘够用

当执行：sudo /etc/init.d/php-fpm start 出现以下错误：ALERT: [pool www] user has not been defined

解决办法：

vim /usr/local/php7/etc/php-fpm.d/www.conf 添加:

user=username

group=groupname

注：其中username需要是系统中存在的用户，否则RROR: [pool www] cannot get uid for user ' username '

当执行：/etc/init.d/php-fpm 出现以下错误：

Starting php-fpm [07-Mar-2018 08:33:28] NOTICE: PHP message: PHP Warning: PHP Startup: Unable to load dynamic library 'imap' (tried: /usr/local/php7/lib/php/extensions/no-debug-non-zts-20170718/imap (/usr/local/php7/lib/php/extensions/no-debug-non-zts-20170718/imap: cannot open shared object file: No such file or directory), /usr/local/php7/lib/php/extensions/no-debug-non-zts-20170718/imap.so (/usr/local/php7/lib/php/extensions/no-debug-non-zts-20170718/imap.so: cannot open shared object file: No such file or directory)) in Unknown on line 0

[07-Mar-2018 08:33:28] NOTICE: PHP message: PHP Warning: PHP Startup: Invalid library (appears to be a Zend Extension, try loading using zend\_extension=opcache.so from php.ini) in Unknown on line 0

问题分析：/usr/local/php7/lib/php/extensions/no-debug-non-zts-20170718/imap: cannot open shared object file: No such file or directory可知模块加载路径(这是php软件安装后用户添加模块的php扩展文件的存在路径)。

utf8\_mime2text() has new signature

解决方案如下:

yum install -y libc-client-devel

wget下载百度网盘资源出错

解决方案如下：

wget -c --referer=百度云分享链接 -O 保存的文件名 "百度云实际下载地址"

此处-c 为断点续传，--referer为百度云分享地址，-O为指定输出文件，后面接浏览器下载文件的下载地址

**用yum查询想安装的软件**

1.使用YUM查找软件包

命令：yum search~

2.列出所有可安装的软件包

命令：yum list

3.列出所有可更新的软件包

命令：yum list updates

4.列出所有已安装的软件包

命令：yum list installed

5.列出所有已安装但不在Yum Repository 內的软件包

命令：yum list extras

6.列出所指定软件包

命令：yum list ～

7.使用YUM获取软件包信息

命令：yum info ～

8.列出所有软件包的信息

命令：yum info

9.列出所有可更新的软件包信息

命令：yum info updates

10.列出所有已安裝的软件包信息

命令：yum info installed

11.列出所有已安裝但不在Yum Repository 內的软件包信息

命令：yum info extras

12.列出软件包提供哪些文件

命令：yum provides~

## [2015博客升级记(五)：CentOS 7.1编译安装PHP7](https://typecodes.com/web/centos7compilephp7.html)

* 作者：vfhky | 时间：2015-06-26 19:16 | 分类：[web](https://typecodes.com/web/index.html)

这是[《2015年博客升级记》](https://typecodes.com/mix/2015updateblog.html)系列文章的第五篇，主要记录在Linux系统中如何编译安装PHP7。

##### 1 创建php用户和用户组，并在github下载php7源码

首先创建一个名为php且没有登录权限的用户和一个名为php的用户组，然后去GitHub下载php7源码包。

#######新建php用户和php组

[root@typecodes ~]# groupadd -r php && useradd -r -g php -s /bin/false -d /usr/local/php7 -M php

######从GitHub下载php7安装包

[root@typecodes ~]# wget -c --no-check-certificate -O php7-src-master.zip https://github.com/php/php-src/archive/master.zip

######开始解压php7包

[root@typecodes ~]# unzip -q php7-src-master.zip && cd php-src-master

#####安装编译php7时需要的依赖包

[root@typecodes php-src-master]# yum -y install libxml2 libxml2-devel openssl openssl-devel curl-devel libjpeg-devel libpng-devel freetype-devel libmcrypt-devel

##### 2 PHP7编译参数的配置

准备工作做好后，就开始正式配置php7的安装明细了。注意，操作时一定要先把下面反斜杠“\”后面添加的注释文字去掉！！！

######开始生成配置文件

[root@typecodes php-src-master]# ./buildconf

buildconf: checking installation...

buildconf: autoconf version 2.69 (ok)

rebuilding aclocal.m4

rebuilding configure

rebuilding main/php\_config.h.in

######开始配置

[root@typecodes php-src-master]# ./configure \

--prefix=/usr/local/php7 \ [PHP7安装的根目录]

--exec-prefix=/usr/local/php7 \

--bindir=/usr/local/php7/bin \

--sbindir=/usr/local/php7/sbin \

--includedir=/usr/local/php7/include \

--libdir=/usr/local/php7/lib/php \

--mandir=/usr/local/php7/php/man \

--with-config-file-path=/usr/local/php7/etc \ [PHP7的配置目录]

--with-mysql-sock=/var/run/mysql/mysql.sock \ [PHP7的Unix socket通信文件]

--with-mcrypt=/usr/include \

--with-mhash \

--with-openssl \

--with-mysql=shared,mysqlnd \ [PHP7依赖mysql库]

--with-mysqli=shared,mysqlnd \ [PHP7依赖mysql库]

--with-pdo-mysql=shared,mysqlnd \ [PHP7依赖mysql库]

--with-gd \

--with-iconv \

--with-zlib \

--enable-zip \

--enable-inline-optimization \

--disable-debug \

--disable-rpath \

--enable-shared \

--enable-xml \

--enable-bcmath \

--enable-shmop \

--enable-sysvsem \

--enable-mbregex \

--enable-mbstring \

--enable-ftp \

--enable-gd-native-ttf \

--enable-pcntl \

--enable-sockets \

--with-xmlrpc \

--enable-soap \

--without-pear \

--with-gettext \

--enable-session \ [允许php会话session]

--with-curl \ [允许curl扩展]

--with-jpeg-dir \

--with-freetype-dir \

--enable-opcache \ [使用opcache缓存]

--enable-fpm \

--enable-fastcgi \

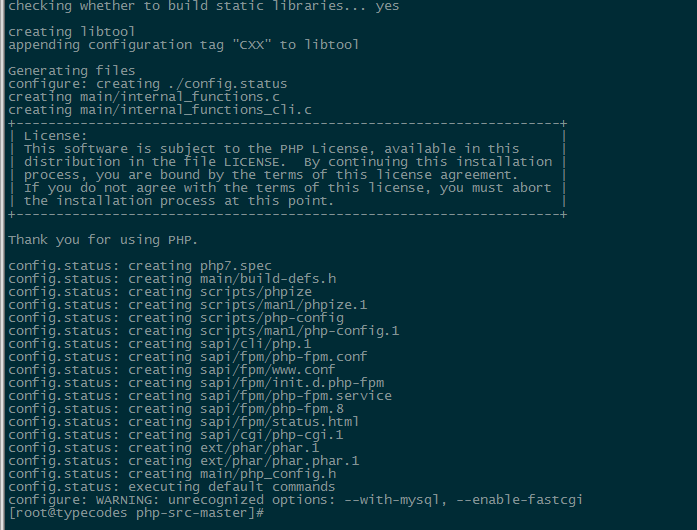
--with-fpm-user=nginx \ [php-fpm的用户]

--with-fpm-group=nginx \ [php-fpm的用户组]

--without-gdbm \

--disable-fileinfo

执行上面的配置命令的结果如下图所示：

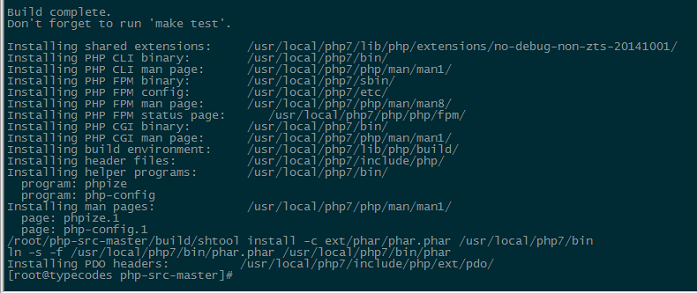


##### 3 开始编译和安装PHP7

相对编译安装MySQL的大量CPU和内存消耗，PHP7的编译安装轻松多了，整个过程大约1个小时左右。

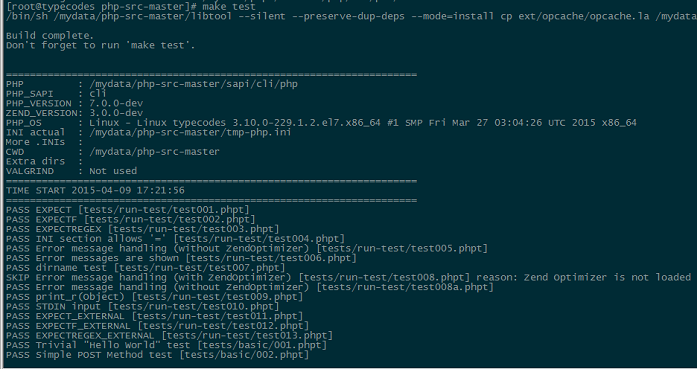
[root@typecodes php-src-master]# make clean && make && make install

看到下图就代表已经编译安装好了PHP7！

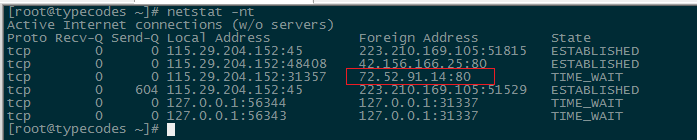


##### 4 选做步骤：执行make test命令进行测试

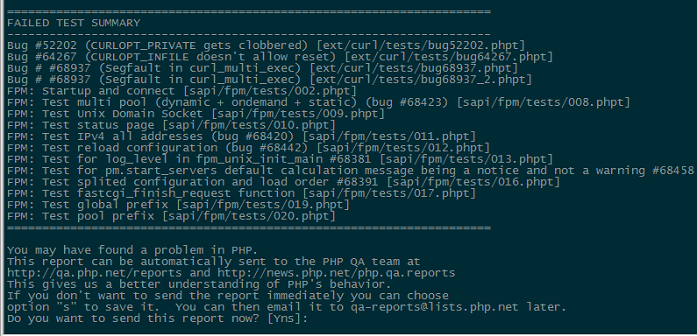
这是一个非必需的操作步骤，执行make test命令。



有趣的一件事就是：在test期间，会和一个IP地址72.52.91.14建立tcp连接，它对应着php官网http://www.php.net。

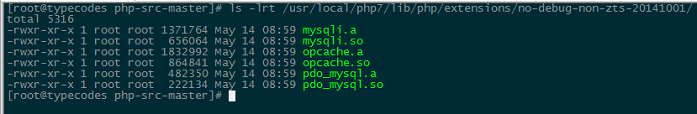


执行完make test命令后，最终的结果如下：



##### 5 查看编译成功后的PHP7安装目录

由于需要和MySQL进行通信，所以需要特别查看PHP7安装后的lib扩展库目录（/usr/local/php7/lib/php/extensions/no-debug-non-zts-20141001/）。需要确保至少存在mysqli.so、pdo\_mysql.so这两个动态库文件，如下图所示。



##### 6 开始设置PHP7的配置文件php.ini、php-fpm.conf、www.conf和php-fpm脚本

可以用编译后的配置文件复制到PHP7的配置目录（/usr/local/php7/etc/），推荐使用[《PHP7中php.ini、php-fpm和www.conf的配置》](https://typecodes.com/web/php7configure.html)文中整理的3个PHP7配置文件和php-fpm服务控制脚本。

#######方法一：直接使用编译后未经优化处理的配置

[root@typecodes php-src-master]# cp php.ini-production /usr/local/php7/etc/php.ini

[root@typecodes php-src-master]# cp /root/php-src-master/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm

[root@typecodes php-src-master]# cp /usr/local/php7/etc/php-fpm.conf.default /usr/local/php7/etc/php-fpm.conf

[root@typecodes php-src-master]# cp /usr/local/php7/etc/php-fpm.d/www.conf.default /usr/local/php7/etc/php-fpm.d/www.conf

#######方法二：使用https://typecodes.com/web/php7configure.html文中的配置

[root@typecodes php-src-master]# mv ~/php.ini /usr/local/php7/etc/php.ini && mv ~/php-fpm /etc/init.d/php-fpm

[root@typecodes php-src-master]# mv ~/php-fpm.conf /usr/local/php7/etc/php-fpm.conf && mv ~/www.conf /usr/local/php7/etc/php-fpm.d/www.conf

##### 7 添加php的环境变量

将php编译生成的bin目录添加到当前Linux系统的环境变量中

[root@typecodes ~]# echo -e '\nexport PATH=/usr/local/php7/bin:/usr/local/php7/sbin:$PATH\n' >> /etc/profile && source /etc/profile

##### 8 设置PHP日志目录和php-fpm进程文件（php-fpm.sock）目录

其中，设置php-fpm进程目录的用户和用户组为nginx，并创建php会话session目录。

#######设置PHP日志目录和php-fpm的运行进程ID文件（php-fpm.sock）目录

[root@typecodes ~]# mkdir -p /var/log/php-fpm/ && mkdir -p /var/run/php-fpm && cd /var/run/ && chown -R nginx:nginx php-fpm

#######修改session的目录配置

[root@typecodes etc]# mkdir -p /var/lib/php/session

[root@typecodes etc]# chown -R nginx:nginx /var/lib/php

##### 9 设置PHP开机启动以及测试配置文件是否正确

######配置开机自启动，增加到主机sysV服务

[root@typecodes php-src-master]# chmod +x /etc/init.d/php-fpm

[root@typecodes php-src-master]# chkconfig --add php-fpm

[root@typecodes php-src-master]# chkconfig php-fpm on

######测试PHP的配置文件是否正确合法

[root@typecodes sbin]# php-fpm -t

[03-May-2015 17:50:04] NOTICE: configuration file /usr/local/php7/etc/php-fpm.conf test is successful

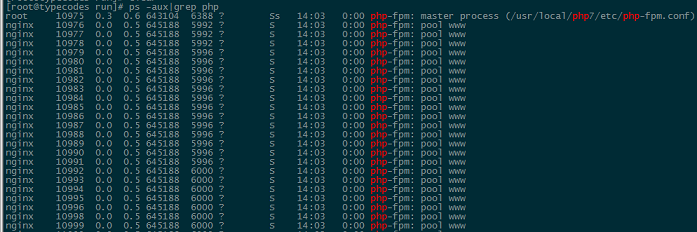
##### 10 启动php服务

在完成上面的操作后，就可以正式使用php服务了。启动php进程服务的命令如下：

[root@typecodes sbin]# service php-fpm start

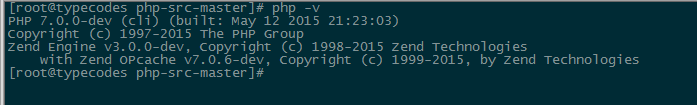
Starting php-fpm done

然后可以通过命令ps -aux|grep php查看是否成功（图中的php-fpm进程数和进程用户nginx都是由www.conf中pm.start\_servers和user的值分别决定的）：



##### 11 查看PHP7版本信息

最后，可以通过命令php -v查看当前PHP版本信息，图中可以看到当前PHP7还使用了Zend OPcache缓存，是因为在php.ini文件中添加了zend\_extension=opcache.so配置。



## [PHP7中php.ini、php-fpm和www.conf的配置](https://typecodes.com/web/php7configure.html)

* 作者：vfhky | 时间：2015-06-27 19:05 | 分类：[web](https://typecodes.com/web/index.html)

根据前文[《2015博客升级记(五)：CentOS 7.1编译安装PHP7》](https://typecodes.com/web/centos7compilephp7.html)的configure编译参数设定，安装后的PHP7配置文件所在路径是/usr/local/php7/etc。该目录下会有两个配置文件：php-fpm.conf和php.ini，以及一个文件夹php-fpm.d（包含www.conf文件）。

因此，这篇文章主要对这三个PHP配置文件的进行简单说明，具体的配置过程见本文末尾的附录部分。

##### 1 配置php.ini

php.ini是php运行核心配置文件：

######避免PHP信息暴露在http头中

expose\_php = Off

######避免暴露php调用mysql的错误信息

display\_errors = Off

######在关闭display\_errors后开启PHP错误日志（路径在php-fpm.conf中配置）

log\_errors = On

######设置PHP的扩展库路径

extension\_dir = "/usr/local/php7/lib/php/extensions/no-debug-non-zts-20141001/"

######设置PHP的opcache和mysql动态库

zend\_extension=opcache.so

extension=mysqli.so

extension=pdo\_mysql.so

######设置PHP的时区

date.timezone = PRC

######开启opcache

[opcache]

; Determines if Zend OPCache is enabled

opcache.enable=1

######设置PHP脚本允许访问的目录（需要根据实际情况配置）

;open\_basedir = /usr/share/nginx/html;

##### 2 配置php-fpm.conf

php-fpm.conf是php-fpm进程服务的配置文件：

######设置错误日志的路径

error\_log = /var/log/php-fpm/error.log

######引入www.conf文件中的配置

include=/usr/local/php7/etc/php-fpm.d/\*.conf

##### 3 配置www.conf（在php-fpm.d目录下）

www.conf这是php-fpm进程服务的扩展配置文件：

######设置用户和用户组

user = nginx

group = nginx

######根据nginx.conf中的配置fastcgi\_pass unix:/var/run/php-fpm/php-fpm.sock;设置PHP监听

; listen = 127.0.0.1:9000 #####不建议使用

listen = /var/run/php-fpm/php-fpm.sock

######开启慢日志

slowlog = /var/log/php-fpm/$pool-slow.log

request\_slowlog\_timeout = 10s

######设置php的session目录（所属用户和用户组都是nginx）

php\_value[session.save\_handler] = files

php\_value[session.save\_path] = /var/lib/php/session

##### 4 附录

在前文[《2015博客升级记(六)：Nginx配置HTTPS和SPDY实战》](https://typecodes.com/web/centos7nginxhttpsspdy.html)和[《CentOS系统MySQL的配置文件my.cnf》](https://typecodes.com/web/centosmysqlconfig.html)已经说明了，目前已经把这个关于Nginx、MySQL和PHP配置的全部托管在GitHub和coding.net上，地址如下：

GitHub地址：<https://github.com/vfhky/mylnmp>；

Coding地址：<https://coding.net/u/vfhky/p/mylnmp/git>。

最后，附上工程的截图：

