



## Nginx + php-fpm

采用 Linux + Nginx + MySQL + PHP 搭建服务器环境：

Nginx	nginx-1.25.3.tar.gz	<a href="http://nginx.org/en/download.html">http://nginx.org/en/download.html</a>
MySQL	mysql-5.7.28.tar.gz	<a href="http://www.mysql.com/downloads/mysql/">http://www.mysql.com/downloads/mysql/</a>
PHP	php-8.3.1.tar.gz	<a href="http://www.php.net/downloads.php">http://www.php.net/downloads.php</a>
libjpeg	jpegsrc.v9e.tar.gz	<a href="http://www.iij.org/">http://www.iij.org/</a>
libpng	libpng-1.6.40.tar.gz	<a href="http://www.libpng.org/">http://www.libpng.org/</a>
gettext	gettext-0.20.1.tar.gz	<a href="http://www.gnu.org/s/gettext/">http://www.gnu.org/s/gettext/</a>
gd	libgd-2.3.3.tar.gz	<a href="http://www.libgd.org/">http://www.libgd.org/</a>
zlib	zlib-1.2.13.tar.gz	<a href="http://zlib.net/">http://zlib.net/</a>
pcre	pcre-8.45.tar.gz	<a href="http://www.pcre.org/">http://www.pcre.org/</a>
libevent	libevent-2.1.12-stable.tar.gz	<a href="http://monkey.org/~provos/libevent/">http://monkey.org/~provos/libevent/</a>

### 一. 安装环境（需要先升级 gcc (GCC) 至 13.2.0）

```
# export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH

# 安装 zlib
export CFLAGS="-O3 -fPIC"
tar zxvf zlib-1.2.13.tar.gz
cd zlib-1.2.13
./configure --prefix=/usr/local --static
make
make install

# 安装 libtool automake autoconf
tar zxvf libtool-2.4.6.tar.gz
cd libtool-2.4.6
./configure --prefix=/usr
make
make install

tar zxvf automake-1.16.5.tar.gz
cd automake-1.16.5
./configure --prefix=/usr/local
make
make install

tar zxvf autoconf-2.72.tar.gz
```

```
cd autoconf-2.72
./configure --prefix=/usr/local
make
make install

tar zxvf m4-1.4.19.tar.gz
cd m4-1.4.19
./configure --prefix=/usr
make
make install

# 安装 openssl
tar zxvf openssl-1.1.1w.tar.gz
cd openssl-1.1.1w
./config --prefix=/usr/local/ssl -fPIC enable-shared
make
make install

# 安装 libevent
tar zxvf libevent-2.1.12-stable.tar.gz
cd libevent-2.1.12
./configure --prefix=/usr/local PKG_CONFIG_PATH=/usr/local/ssl/lib/pkgconfig
make
make install

# 安装 iconv
tar zxvf libiconv-1.17.tar.gz
cd libiconv-1.17
./configure --prefix=/usr/local
make
make install
```

```
# export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH

# 安装 pcre2
tar zxvf pcre2-10.42.tar.gz
cd pcre2-10.42
./configure --prefix=/usr/local --enable-jit
make
make install

# PHP8 要求 libxml2-2.9.14 , libxml2-2.9.14 要求 python3
# 安装 libxml2 ( 复制 aclocal 为 bugfix 或设置 ./autogen.sh ACLOCAL_PATH=/usr/share/aclocal )
tar zxvf libxml2-2.9.14.tar.gz
cd libxml2-2.9.14
cp /usr/share/aclocal/*.m4 /usr/local/share/aclocal/
```

```
./autogen.sh --prefix=/usr/local/libxml2 --enable-shared --enable-static
--with-python=/usr/local
make
make install

# 安装 onig
tar zxvf onig-6.9.7.tar.gz
cd onig-6.9.7
./configure --prefix=/usr/local
make
make install

# 安装 jpeg
tar zxvf jpegsrc.v9e.tar.gz
cd jpeg-9e
./configure --prefix=/usr/local
make
make install

# 安装 png
tar zxvf libpng-1.6.40.tar.gz
cd libpng-1.6.40
./configure --prefix=/usr/local
make
make install

# 安装 gettext
tar zxvf gettext-0.20.1.tar.gz
cd gettext-0.20.1
./configure --prefix=/usr/local
make
make install

# 安装 freetype
tar zxvf freetype-2.5.5.tar.gz
cd freetype-2.5.3
./configure --prefix=/usr/local --enable-shared --enable-static
make
make install

# 安装 gd
tar zxvf libgd-2.3.3.tar.gz
cd libgd-2.3.3
./configure --prefix=/usr/local --with-zlib=/usr/local --with-png=/usr/local --with-
jpeg=/usr/local --with-freetype=/usr/local
make
make install
```

## 二. 安装 Nginx

```
# 解压 Nginx
tar zxvf openssl-1.1.1w.tar.gz

# 安装 Nginx ( 需要 openssl 源码 )
tar zxvf nginx-1.25.3.tar.gz
cd nginx-1.25.3
./configure --prefix=/usr/local/nginx --with-poll_module --with-pcre --with-threads
--with-http_sub_module --with-http_ssl_module --with-openssl=../openssl-1.1.1w
make
make install
```

## 三. 安装 Mysql

```
# 安装 Mysql

groupadd mysql
useradd mysql -g mysql

tar zxvf mysql-5.7.28.tar.gz
cd mysql-5.7.28
cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DWITH_INNOBASE_STORAGE_ENGINE=1
-DMYSQL_TCP_PORT=3306 -DDEFAULT_CHARSET=utf8mb4 -DDEFAULT_COLLATION=utf8mb4_general_ci
-DMYSQL_USER=mysql -DWITH_DEBUG=0
make
make install

# 初始化

cd /usr/local/mysql/bin
./mysqld --initialize --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
chown mysql:mysql /usr/local/mysql -R

# 启动

cd /usr/local/mysql/bin
./mysqld_safe --defaults-file=/etc/my.cnf --user=mysql &
```

## 四. 安装 PHP

```
# 开始编译安装 PHP 以 FPM 方式安装, 如找不到需要组件则按需定义如下环境变量

# export ONIG_LIBS="-L/usr/local/lib -lonig"
# export ONIG_CFLAGS="-I/usr/local/include"

# export SQLITE_LIBS="-L/usr/local/lib"
# export SQLITE_CFLAGS="-I/usr/local/include"

# export PCRE2_LIBS="-L/usr/local/lib"
# export PCRE2_CFLAGS="-I/usr/local/include"
```

```

# export PNG_LIBS="-L/usr/local/lib"
# export PNG_CFLAGS="-I/usr/local/include"

# export JPEG_LIBS="-L/usr/local/lib"
# export JPEG_CFLAGS="-I/usr/local/include"

# export LIBZIP_LIBS="-L/usr/local/lib"
# export LIBZIP_CFLAGS="-I/usr/local/include"

# export LIBXML_LIBS="-L/usr/local/libxml2/lib"
# export LIBXML_CFLAGS="-I/usr/local/libxml2/include/libxml2"

export PKG_CONFIG_PATH=/usr/lib/pkgconfig
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH
export PKG_CONFIG_PATH=/usr/local/libxml2/lib/pkgconfig:$PKG_CONFIG_PATH

tar xzf php-8.3.1.tar.gz
cd php-8.3.1
./configure --prefix=/usr/local/php --enable-fpm --enable-opcache --enable-mbstring
--with-config-file-path=/usr/local/php/lib --with-pdo-mysql=/usr/local/mysql/ --with-curl
--enable-calendar --with-pcre-jit --with-jpeg --enable-gd --with-gettext --with-freetype
--without-sqlite3 --without-pdo-sqlite --with-iconv=/usr/local
make
make install

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

```

## 五. 配置 PHP php-fpm

```

# 配置 php-fpm : /usr/local/php/etc/php-fpm.conf

[global]

[www]

listen = 127.0.0.1:9000

user = nobody
group = nobody
pm = dynamic
pm.max_children = 50
pm.start_servers = 10
pm.min_spare_servers = 5
pm.max_spare_servers = 40

# 最开始是 pm.start_servers 指定的数量，如果请求较多，则会自动增加，保证空闲的进程数不小于
pm.min_spare_servers，如果进程数较多，也会进行相应清理，保证多余的进程数不多于
pm.max_spare_servers

```

## 六. 配置 Nginx FastCGI

```

# fastcgi.conf 文件 : vim /usr/local/nginx/conf/fastcgi.conf

fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

```

```

fastcgi_param QUERY_STRING      $query_string;
fastcgi_param REQUEST_METHOD    $request_method;
fastcgi_param CONTENT_TYPE      $content_type;
fastcgi_param CONTENT_LENGTH    $content_length;

fastcgi_param SCRIPT_NAME        $fastcgi_script_name;
fastcgi_param REQUEST_URI        $request_uri;
fastcgi_param DOCUMENT_URI      $document_uri;
fastcgi_param DOCUMENT_ROOT     $document_root;
fastcgi_param SERVER_PROTOCOL    $server_protocol;

fastcgi_param GATEWAY_INTERFACE CGI/1.1;
fastcgi_param SERVER_SOFTWARE    nginx/$nginx_version;

fastcgi_param REMOTE_ADDR        $remote_addr;
fastcgi_param REMOTE_PORT        $remote_port;
fastcgi_param SERVER_ADDR        $server_addr;
fastcgi_param SERVER_PORT        $server_port;
fastcgi_param SERVER_NAME        $server_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param REDIRECT_STATUS    200;

```

## 七. 配置 Nginx

```

# 配置 Nginx vim /usr/local/nginx/conf/nginx.conf

worker_processes 4;          -- 进程数四核填4
worker_rlimit_nofile 10240;

events {
    use epoll;                -- 使用 I/O 复用模式，提高磁盘读取性能
    worker_connections 10240; -- 每个进程最大连接数，总 = 1024 x 2
}

http {

    include mime.types;
    default_type application/octet-stream;

    sendfile on;                -- 优化性能
    server_tokens off;          -- 隐藏版本号

    gzip on;                    -- 启用 Gzip 压缩
    gzip_proxied any;
    gzip_types text/plain text/css text/javascript application/x-javascript
               application/javascript application/json;

    keepalive_timeout 65;
    client_max_body_size 15m;    -- 最大页面 15 Mb
    index index.html index.htm index.php; -- 全局默认首页

    fastcgi_cache_path cache levels=1:2 keys_zone=arimac:100m inactive=30m max_size=2g;

    server {

        listen 80;
        server_name localhost;
        access_log off;
        charset utf-8;
        root /var/www;

        location / {
            root /var/www;        -- 多加 root 少出问题
            rewrite ^/$ /arimac/ permanent; -- permanent 代表永久重定向，跳转
            autoindex on;

```

```

        index index.php index.html index.htm;
    }

    location /mysql {
        alias /var/www/mysql;
    }

    location ~ /mysql/.*\.php {
        root /var/www;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        include fastcgi.conf;
    }

    location ~ \.php {
        fastcgi_cache arimac;
        fastcgi_cache_valid 200 304 1m;
        fastcgi_cache_key $host$request_uri;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        include fastcgi.conf;
    }
}
}

```

## 八. 启动服务器

```

# 启动 Nginx : /usr/local/nginx/sbin/nginx
# 重载 Nginx : /usr/local/nginx/sbin/nginx -s reload
# 重开 Nginx : /usr/local/nginx/sbin/nginx -s reopen
# 关闭 Nginx : /usr/local/nginx/sbin/nginx -s stop

# 启动 PHP-fpm : /usr/local/php/sbin/php-fpm

master 进程可以理解以下信号:
INT, TERM 立刻终止
QUIT      平滑终止
USR1      重新打开日志文件
USR2      平滑重载所有 worker 进程并重新载入配置和二进制模块

kill -INT `ps aux|grep "php-fpm: master"|grep -v grep|awk '{print $2}` -- 关闭
kill -USR2 `ps aux|grep "php-fpm: master"|grep -v grep|awk '{print $2}` -- 重启

```