

LNMP 安装与配置

Nginx 与 Apache 性能综合对比，如下图：

Web 服务器	Nginx	Apache
反向代理	非常好	好
Rewrite 规则	非常好	好
FastCGI	好	差
热部署	支持	不支持
系统压力比较	很小	小
稳定性	非常好	好
安全性	一般	好
技术资料	很少	非常多
静态文件处理	非常好	一般
虚拟主机	支持	支持
内存消耗	非常小	很大

一. 系统需求：

- CentOS/RHEL/Fedora/Debian/Ubuntu 系统
- 需要 3GB 以上硬盘剩余空间
- **MySQL 5.6 及 MariaDB 10 必须 1G 以上内存。**
- Linux 下区分大小写，输入命令时请注意！
- 确定 yum 源正常使用！

二. 安装步骤：

1、下载并安装 LNMP 一键安装包：

```
#tar -zxvf lnmp1.2-full.tar.gz
```

```
#cd lnmp1.2-full
```

```
#./install.sh lnmp
```

安装 LNMP 执行：`wget -c http://soft.vpser.net/lnmp/lnmp1.2-full.tar.gz && tar zxf lnmp1.2-full.tar.gz && cd lnmp1.2-full && ./install.sh lnmp`

如需要安装 LNMPA 或 LAMP，将 `./install.sh` 后面的参数替换为 `lnmpa` 或 `lamp` 即可。

按上述命令执行后，会出现如下提示：

```

+-----+
|          LNMP V1.2 for Fedora Linux Server, Written by Licess          |
+-----+
|          A tool to auto-compile & install LNMP/LNMPA/LAMP on Linux      |
+-----+
|          For more information please visit http://www.lnmp.org          |
+-----+
Please setup root password of MySQL. (Default password: root)
Please enter: █

```

需要设置 MySQL 的 root 密码（不输入直接回车将会设置为 root），输入后回车进入下一步，如下图所示：

```

=====
Do you want to enable or disable the InnoDB Storage Engine?
Default enable, Enter your choice [Y/n]: █

```

这里需要确认是否启用 MySQL InnoDB，如果不确定是否启用可以输入 y，输入 y 表示启用，输入 n 表示不启用。默认为 y 启用，输入后回车进入下一步，选择 MySQL 版本：

```

=====
You have 5 options for your DataBase install.
1: Install MySQL 5.1.73
2: Install MySQL 5.5.42 (Default)
3: Install MySQL 5.6.23
4: Install MariaDB 5.5.42
5: Install MariaDB 10.0.17
Enter your choice (1, 2, 3, 4 or 5): █

```

输入 MySQL 或 MariaDB 版本的序号，回车进入下一步，选择 PHP 版本：

```

=====
You have 5 options for your PHP install.
1: Install PHP 5.2.17
2: Install PHP 5.3.29
3: Install PHP 5.4.40 (Default)
4: Install PHP 5.5.24
5: Install PHP 5.6.8
Enter your choice (1, 2, 3, 4 or 5): █

```

输入 PHP 版本的序号，回车进入下一步，选择是否安装内存优化：

```

=====
You have 3 options for your Memory Allocator install.
1: Don't install Memory Allocator. (Default)
2: Install Jemalloc
3: Install TCMalloc
Enter your choice (1, 2 or 3): █

```

可以选择不安装、Jemalloc 或 TCMalloc，输入对应序号回车。

如果是 LNMPA 或 LAMP 的话还需要设置管理员邮箱

```

=====
Please enter Administrator Email Address: █

```

再选择 Apache 版本

```
=====
You have 2 options for your Apache install.
1: Install Apache 2.2.29 (Default)
2: Install Apache 2.4.10
Enter your choice (1 or 2):
```

提示“Press any key to install...or Press Ctrl+c to cancel”后，按回车键确认开始安装。

LNMP 脚本就会自动安装编译 Nginx、MySQL、PHP、phpMyAdmin、Zend Optimizer 这几个软件。

3、安装完成

如果显示 Nginx: OK, MySQL: OK, PHP: OK

```
+-----+
|   Manager for LNMP, Written by Licens   |
+-----+
|           http://lnmp.org              |
+-----+
nginx (pid 1808 1806) is running...
php-fpm is runing!
• mysql.service - LSB: start and stop MySQL
  Loaded: loaded (/etc/init.d/mysql)
  Active: active (running) since Tue 2015-05-05 21:15:59 CST; 158ms ago
  CGroup: /system.slice/mysql.service
          └─1839 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/usr/loc
              2225 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --da
  ugin --user=mysql --log-error=/usr/local/mysql/var/lnmp.err --pid-file=/usr/l

May 05 21:15:57 lnmp mysql[1829]: Starting MySQL
May 05 21:15:59 lnmp mysql[1829]: ...
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address          State
tcp        0      0 0.0.0.0:111             0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:53072           0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:3306            0.0.0.0:*                LISTEN
tcp6       0      0 :::111                  :::*                      LISTEN
tcp6       0      0 :::22                   :::*                      LISTEN
tcp6       0      0 :::1:25                 :::*                      LISTEN
tcp6       0      0 :::47771                :::*                      LISTEN
Install lnmp V1.2 completed! enjoy it.
```

并且 Nginx、MySQL、PHP 都是 running，80 和 3306 端口都存在，并 Install lnmp V1.2 completed! enjoy it. 的话，说明已经安装成功。

4、安装失败

```
===== Check install =====
Checking ...
Nginx: OK
MySQL: OK
Error: PHP install failed.
Sorry,Failed to install LNMP!
Please visit http://bbs.vpser.net/forum-25-1.html feedback errors and logs.
You can download /root/lnmp-install.log from your server,and upload lnmp-install.log to LNMP Forum.
[root@li846-54 lnmp1.2]#
```

如果出现类似上图的提示，则表明安装失败，说明没有安装成功！！

二. LNMP 相关软件安装目录

Nginx 目录: /usr/local/nginx/

MySQL 目录 : /usr/local/mysql/

MySQL 数据库所在目录: /usr/local/mysql/var/

PHP 目录 : /usr/local/php/

PHPMyAdmin 目录 : /home/wwwroot/default/phpmyadmin/

默认网站目录 : /home/wwwroot/default/

Nginx 日志目录: /home/wwwlogs/

三. LNMP 相关配置文件位置

Nginx 主配置文件: /usr/local/nginx/conf/nginx.conf

MySQL 配置文件: /etc/my.cnf

PHP 配置文件: /usr/local/php/etc/php.ini

四. LNMP 状态管理命令

LNMP 状态管理: lnmp {start|stop|reload|restart|kill|status}

LNMP 各个程序状态管理: lnmp {nginx|mysql|mariadb|php-fpm|pureftpd}
{start|stop|reload|restart|kill|status}

五. 配置文件

```
#vi /usr/local/nginx/conf/nginx.conf
```

```
user www www;
```

```
worker_processes auto;
```

#启动进程

```
error_log /home/wwwlogs/nginx_error.log crit;
```

#错误日志

```
pid /usr/local/nginx/logs/nginx.pid;
```

#主进程 PID 保存文件

#Specifies the value for maximum file descriptors that can be opened by this

```

process.
worker_rlimit_nofile 51200;
#文件描述符数量

events
{
    use epoll;
    #网络 I/O 模型, 建议 linux 使用 epoll, FreeBSD 使用 kqueue
    #epoll 是多路复用 IO(I/O Multiplexing)中的一种方式, 但是仅用于 linux2.6 以
    上内 核, 可以大大提高 nginx 的性能
    worker_connections 51200;
    #单个工作进程最大允许连接数
    multi_accept on;

}

http
#整体环境配置
{
    include             mime.types;
    default_type        application/octet-stream;
    #设定 mime 类型, 文件传送类型由 mime.type 文件定义

    server_names_hash_bucket_size 128;    #保存服务器名字的 hash 表大小
    client_header_buffer_size 32k;        #客户端请求头部缓冲区大小
    large_client_header_buffers 4 32k;    #最大客户端头缓冲大小
    client_max_body_size 50m;             #客户端最大上传文件大小 (M)

    sendfile on;
    #sendfile 指令指定 nginx 是否调用 sendfile 函数 (zero copy 方式) 来输出文
    件, 对于普通应用, 必须设为 on。如果用来进行下载等应用磁盘 IO 重负载应
    设置为 off, 以平衡磁盘与网络 I/O 处理速度, 降低系统的 uptime.
    #高效文件传输
    tcp_nopush        on;
    #这个是默认的, 结果就是数据包不会马上传送出去, 等到数据包最大时, 一次性的
    传输出去, 这样有助于解决网络堵塞。(只在 sendfile on 时有效)

    keepalive_timeout 60;
    #连接超时时间

    tcp_nodelay on;
    #禁用 nagle 算法, 也即不缓存数据。有效解决网络阻塞

    fastcgi_connect_timeout 300;
    fastcgi_send_timeout 300;
    fastcgi_read_timeout 300;
    fastcgi_buffer_size 64k;
    fastcgi_buffers 4 64k;
    fastcgi_busy_buffers_size 128k;
    fastcgi_temp_file_write_size 256k;
    #fastcgi 设置

```

```

        gzip on;
        gzip_min_length 1k;
        gzip_buffers 4 16k;
        gzip_http_version 1.1;
        gzip_comp_level 2;
        gzip_types text/plain application/javascript
application/x-javascript text/javascript text/css application/xml
application/xml+rss;
        gzip_vary on;
        gzip_proxied expired no-cache no-store private auth;
        gzip_disable "MSIE [1-6]\.";

#limit_conn_zone $binary_remote_addr zone=perip:10m;
##If enable limit_conn_zone, add "limit_conn perip 10;" to server section.
        server_tokens off;
        #隐藏 nginx 版本号 (curl -I 192.168.4.154 可以查看, 更加安全)

        #log format
        log_format access '$remote_addr - $remote_user [$time_local]
"$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" $http_x_forwarded_for';
        #定义日志格式

server
{
    listen 80 default_server;
    #listen [::]:80 default_server ipv6only=on;
    #监听 80 端口
    server_name www.lnmp.org;
    #服务器名
    index index.html index.htm index.php;
    #默认网页文件
    root /home/wwwroot/default;
    #网页主目录

    #error_page 404 /404.html;
    include enable-php.conf;

    location /nginx_status
    {
        stub_status on;
        access_log off;
    }
    #开启 status 状态监测
    location ~ .*\. (gif|jpg|jpeg|png|bmp|swf)$
    {
        expires 30d;
    }
    #静态文件处理, 保存期 30 天
    location ~ .*\. (js|css)?$
    {

```

```

        expires      12h;
    }
    #js 和 css 文件处理, 保存期 12 小时
    location ~ /\. {
        deny all;
    }

    access_log /home/wwwlogs/access.log access;
    #正确访问日志
}
include vhost/*.conf;
#vhost/下子配置文件生效
}

```

检查 nginx 配置文件语句错误

/usr/local/nginx/sbin/nginx -t

平滑重启 nginx 进程

kill -HUP nginx

六.配置实验

1.nginx 虚拟主机

sina 和 sohu 域名事先解析

Vi /usr/local/nginx/conf/nginx.conf

==>www.sina.com 公司网站

```

server
{
    listen 80 ;
    #listen [::]:80 default_server ipv6only=on;
    server_name www.sina.com;
    index index.html index.htm index.php;
    root /home/wwwroot/sina;
    #error_page 404 /404.html;
    include enable-php.conf;
}

```

==>www.sohu.com 公司网站

```

server
{
    listen 80 ;
    #listen [::]:80 default_server ipv6only=on;
    server_name www.sohu.com;
}

```

```

        index index.html index.htm index.php;
        root /home/wwwroot/sohu;
        #error_page 404 /404.html;
        include enable-php.conf;
    }

```

重启 nginx

最后在客户端测试虚拟主机 www.baidu.com 和 www.sina.com 两家公司网站

2.列表页显示

```

server
{
    listen      80;
    server_name www.sina.com;
    index index.html index.htm index.php;
    root /home/wwwroot/sina;
    autoindex on;
}

```

3.nginx 状态监控

```

location /nginx_status{
    stub_status on;
    access_log off;
}
#客户端访问网址:http://IP/nginx_status

```

4.rewrite 正则过滤

```

location ~ /\.php$ {
    proxy_pass http://127.0.0.1;
}

```

Rewrite 指令最后一项参数为 flag 标记，支持的 flag 标记如下：

Last

停止执行当前这一轮的 ngx_http_rewrite_module 指令集，然后查找匹配改变后 URI 的新 location；

Break

停止执行当前这一轮的 ngx_http_rewrite_module 指令集；

Redirect

在 replacement 字符串未以“http://”或“https://”开头时，使用返回状态码为 302 的临时重定向；

Permanent

返回状态码为 301 的永久重定向。

Last 和 break 用来实现 uri 重写，浏览器地址栏的 url 地址不变，但在服务器访问的路径发生了变化，redirect 和 permanent 用来实现 url 跳转，浏览器地址栏会显示跳转后的 url 地址，使用 alias 指令时必须使用 last 标记，使用 proxy_pass 指令时要使用 break 标记，last 标记在

本条 rewrite 规则执行完毕后，会对其所在的 server{} 标签重新发起请求，而 break 标记则在本条规则匹配完成后，终止匹配，不再匹配后面的规则。

例 1：域名跳转

输入 www.sina.com，跳转到 www.sohu.com

```
server
{
    listen      80;
    server_name www.sina.com;
    index index.html index.htm index.php;
    root /home/wwwroot/sina;

    if ($http_host = www.sina.com) {
        rewrite (.*?) http://www.sohu.com permanent;
    }
}

server
{
    listen      80;
    server_name www.sohu.com;
    index index.html index.htm index.php;
    root /home/wwwroot/sohu;
}
```

例 2：文件跳转

```
server
{
    listen      80;
    server_name www.sina.com;
    index index.html index.htm index.php;
    root /home/wwwroot/sina;
    rewrite index(\d+).html /index.php?id=$1 last;
}
```

5. 代理负载均衡技术（反向代理）

http

```
upstream myweb1 {
    #定义地址池
    server 192.168.242.100:80;
```

```

server 192.168.242.111:80;
}

server {
    listen      80;
    server_name www.sohu.com;
    #使用 www.sohu.com 访问
    location / {
        proxy_pass http://myweb1;
        #使用地址池
        proxy_next_upstream    http_500    http_502    http_503    error    timeout
        invalid_header;
        #定义故障转移。后端服务器节点返回 500、502、503、504 和超时等错误时，自动把
        请求转发到另一台服务器，转移故障。
        proxy_set_header Host $host;
        #利用 HOST 变量向后端服务器传递需要解析的客户端访问的域名（传递域名）
        proxy_set_header X-Forwarded-For $remote_addr;
        #remote_addr 把客户端真实 IP 赋予 X-Forwarded-For。后端服务器才能获取真
        实的客户端 IP。以便记录日志，要不日志中记录的访问信息都是负载服务器，而不是客户
        端（传递 IP）
    }
}

```