

NSD OPERATION DAY02

1. [案例1：部署LNMP环境](#)
2. [案例2：构建LNMP平台](#)
3. [案例3：地址重写](#)

1 案例1：部署LNMP环境

1.1 问题

安装部署Nginx、MariaDB、PHP环境

- 安装部署Nginx、MariaDB、PHP、PHP-FPM；
- 启动Nginx、MariaDB、FPM服务；
- 并测试LNMP是否工作正常。

1.2 方案

在RHEL7系统中，源码安装Nginx，使用RPM包安装MariaDB、PHP、PHP-FPM软件。

操作过程中需要安装的软件列表如下：

- nginx
- mariadb、mariadb-server、mariadb-devel
- php、php-fpm、php-mysql

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装软件

1) 使用yum安装基础依赖包

```
01. [root@proxy ~]# yum -y install gcc openssl-devel pcre-devel zlib
```

2) 源码安装Nginx

```
01. [root@proxy ~]# useradd -s /sbin/nologin nginx
02. [root@proxy ~]# tar -xvf nginx-1.12.2.tar.gz
03. [root@proxy ~]# cd nginx-1.12.2
04. [root@proxy nginx-1.12.2]# ./configure \
05. > --user=nginx --group=nginx \
06. > --with-http_ssl_module
07. [root@proxy ~]# make && make install
08. .. ..
```

[Top](#)

3) 安装MariaDB

Mariadb在新版RHEL7光盘中包含有该软件，配置yum源后可以直接使用yum安装：

```
01. [root@proxy ~]# yum -y install mariadb mariadb-server mariadb-devel
```

4) php和php-fpm(该软件包在lnmp_soft中提供)

```
01. [root@proxy ~]# yum -y install php php-mysql
02. [root@proxy ~]# yum -y install php-fpm-5.4.16-42.el7.x86_64.rpm
```

步骤二：启动服务

1) 启动Nginx服务

这里需要注意的是，如果服务器上已经启动了其他监听80端口的服务软件（如httpd），则需要先关闭该服务，否则会出现冲突。

```
01. [root@proxy ~]# systemctl stop httpd //如果该服务存在
02. [root@proxy ~]# /usr/local/nginx/sbin/nginx //启动Nginx
03. [root@proxy ~]# netstat -tlnlp | grep :80
04. tcp      0      0 0.0.0.0:80          0.0.0.0:*        LISTEN
```

2) 启动MySQL服务

```
01. [root@proxy ~]# systemctl start mariadb //启动服务器
02. [root@proxy ~]# systemctl status mariadb //查看服务状态
03. [root@proxy ~]# systemctl enable mariadb //设置开机启动
```

3) 启动PHP-FPM服务

```
01. [root@proxy ~]# systemctl start php-fpm //启动服务
02. [root@proxy ~]# systemctl status php-fpm //查看服务状态
03. [root@proxy ~]# systemctl enable php-fpm //设置开机启动
```

[Top](#)

4) 设置防火墙与SELinux

```
01. [root@proxy ~]# firewall-cmd --set-default-zone=trusted
02. [root@proxy ~]# setenforce 0
```

2 案例2：构建LNMP平台

2.1 问题

沿用练习一，通过调整Nginx服务端配置，实现以下目标：

1. 配置Fast-CGI支持PHP网页
2. 创建PHP测试页面，测试使用PHP连接数据库的效果

2.2 方案

使用2台RHEL7虚拟机，其中一台作为LNMP服务器（192.168.4.5）、另外一台作为测试用的Linux客户机（192.168.4.100），如图-1所示。

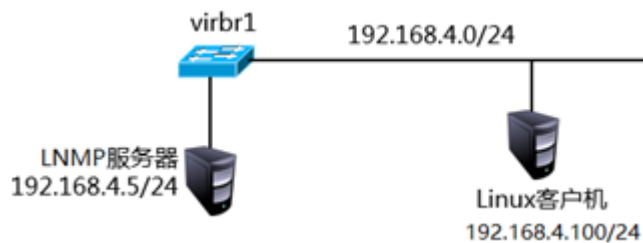


图-1

Nginx结合FastCGI技术即可支持PHP页面架构，因此本案例，需要延续练习一的实验内容，通过修改Nginx及php-fpm配置文件实现对PHP页面的支持。

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建并修改php-fpm配置文件

1) 查看php-fpm配置文件

```
01. [root@proxy etc]# vim /etc/php-fpm.d/www.conf
02. [www]
03. listen = 127.0.0.1:9000           //PHP端口号
04. pm.max_children = 32             //最大进程数量
05. pm.start_servers = 15            //最小进程数量
06. pm.min_spare_servers = 5         //最少需要几个空闲着的进程
07. pm.max_spare_servers = 32        //最多允许几个进程处于空闲状态
```

[Top](#)

2) 确认php-fpm服务已经启动

```
01. [root@proxy ~]# systemctl restart php-fpm //重启服务
02. [root@proxy ~]# systemctl status php-fpm
```

步骤二：修改Nginx配置文件并启动服务

```
01. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
02. location / {
03.     root    html;
04.     index  index.php index.html index.htm;
05. }
06. location ~ \.php$ {
07.     root            html;
08.     fastcgi_pass    127.0.0.1:9000;
09.     fastcgi_index   index.php;
10.     #fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;
11.     include         fastcgi.conf;
12. }
13. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

步骤三：创建PHP页面，测试LNMP架构能否解析PHP页面

1) 创建PHP测试页面1，可以参考lnmp_soft/php_scripts/test.php：

```
01. [root@proxy ~]# vim /usr/local/nginx/html/test1.php
02. <?php
03. $i="This is a test Page";
04. echo $i;
05. ?>
```

2) 创建PHP测试页面,连接并查询MariaDB数据库。

可以参考lnmp_soft/php_scripts/mysql.php:

```
01. [root@proxy ~]# vim /usr/local/nginx/html/test2.php
02. <?php
03. $mysqli = new mysqli('localhost','root','密码','mysql');
04. //注意: root为mysql账户名称, 密码需要修改为实际mysql密码, 无密码则留空即可
05. if (mysqli_connect_errno()){
06.     die('Unable to connect!'). mysqli_connect_error();
    Top
```

```
07.     }
08.     $sql = "select * from user";
09.     $result = $mysqli->query($sql);
10.     while($row = $result->fetch_array()){
11.         printf("Host:%s", $row[0]);
12.         printf("</br>");
13.         printf("Name:%s", $row[1]);
14.         printf("</br>");
15.     }
16.     ?>
```

3) 客户端使用浏览器访问服务器PHP首页文档，检验是否成功：

```
01.     [root@client ~]# firefox http://192.168.4.5/test1.php
02.     [root@client ~]# firefox http://192.168.4.5/test2.php
```

4) LNMP常见问题

Nginx的默认访问日志文件为/usr/local/nginx/logs/access.log

Nginx的默认错误日志文件为/usr/local/nginx/logs/error.log

PHP默认错误日志文件为/var/log/php-fpm/www-error.log

如果动态网站访问失败，可用参考错误日志，查找错误信息。

3 案例3：地址重写

3.1 问题

沿用练习二，通过调整Nginx服务端配置，实现以下目标：

1. 所有访问a.html的请求，重定向到b.html;
2. 所有访问192.168.4.5的请求重定向至www.tmooc.cn;
3. 所有访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面;
4. 实现firefox与curl访问相同页面文件，返回不同的内容。

3.2 方案

关于Nginx服务器的地址重写，主要用到的配置参数是rewrite：

- rewrite regex replacement flag
- rewrite 旧地址 新地址 [选项]

3.3 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：修改配置文件(访问a.html重定向到b.html)

1) 修改Nginx服务配置:

```
01. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
02. ... ..
03. server {
04.     listen      80;
05.     server_name localhost;
06.     location / {
07.         root    html;
08.         index  index.html index.htm;
09.         rewrite /a.html /b.html;
10.     }
11. }
12.
13. [root@proxy ~]# echo "BB" > /usr/local/nginx/html/b.html
```

2) 重新加载配置文件

```
01. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

3) 客户端测试

```
01. [root@client ~]# firefox http://192.168.4.5/a.html
```

步骤二：访问a.html重定向到b.html（跳转地址栏）

1) 修改Nginx服务配置:

```
01. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
02. ... ..
03. server {
04.     listen      80;
05.     server_name localhost;
06.     location / {
07.         root    html;
08.         index  index.html index.htm;
09.         rewrite /a.html /b.html redirect;
```

[Top](#)

```
10.     }  
11.     }
```

2) 重新加载配置文件

```
01.     [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

3) 客户端测试 (仔细观察浏览器地址栏的变化)

```
01.     [root@client ~]# firefox http://192.168.4.5/a.html
```

步骤三：修改配置文件(访问192.168.4.5的请求重定向至www.tmooc.cn)

1) 修改Nginx服务配置

```
01.     [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf  
02.     ...  
03.     server {  
04.         listen      80;  
05.         server_name  localhost;  
06.         rewrite ^/ http://www.tmooc.cn/;  
07.         location / {  
08.             root     html;  
09.             index  index.html index.htm;  
10.             # rewrite /a.html /b.html redirect;  
11.         }  
12.     }
```

2) 重新加载配置文件

```
01.     [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

3) 客户端测试

```
01.     [root@client ~]# firefox http://192.168.4.5
```

[Top](#)

步骤四：修改配置文件(访问192.168.4.5/下面子页面，重定向至www.tmooc.cn/下相同的页面)

1) 修改Nginx服务配置

```
01. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
02.
03. ... ..
04. server {
05.     listen      80;
06.     server_name  localhost;
07.     rewrite ^/(.*) http://www.tmooc.cn/$1;
08.     location / {
09.         root    html;
10.         index  index.html index.htm;
11.         # rewrite /a.html /b.html redirect;
12.     }
13. }
```

2) 重新加载配置文件

```
01. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

3) 客户端测试

```
01. [root@client ~]# firefox http://192.168.4.5
```

步骤五：修改配置文件(实现curl和火狐访问相同链接返回的页面不同)

1) 创建网页目录以及对应的页面文件：

```
01. [root@proxy ~]# echo "I am Normal page" > /usr/local/nginx/html/
02.
03. [root@proxy ~]# mkdir -p /usr/local/nginx/html/firefox/
04. [root@proxy ~]# echo "firefox page" > /usr/local/nginx/html/fire
```

[Top](#)

2) 修改Nginx服务配置


```
01. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
02. ...
03. server {
04.     listen      80;
05.     server_name localhost;
06.     location / {
07.         root    html;
08.         index  index.html index.htm;
09.     }
10.     if ($http_user_agent ~* firefox) {           //识别客户端firefox
11.         rewrite ^(.*)$ /firefox/$1;
12.     }
13. }
```

3) 重新加载配置文件

```
01. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
```

4) 客户端测试

```
01. [root@client ~]# firefox http://192.168.4.5/test.html
02. [root@client ~]# curl http://192.168.4.5/test.html
```

5) 地址重写格式【总结】

rewrite 旧地址 新地址 [选项];

last 不再读其他rewrite

break 不再读其他语句，结束请求

redirect 临时重定向

permament 永久重定向

[Top](#)