[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/OPERATION/DAY04/CASE/01/index.html" \l "page_top_case)

# NSD OPERATION DAY04

1. [案例1：PHP的本地Session信息](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/OPERATION/DAY04/CASE/01/index.html" \l "case1)
2. [案例2：构建memcached服务](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/OPERATION/DAY04/CASE/01/index.html" \l "case2)
3. [案例3：LNMP+memcached](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/OPERATION/DAY04/CASE/01/index.html" \l "case3)
4. [案例4：PHP实现session共享](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/OPERATION/DAY04/CASE/01/index.html" \l "case4)

## 1 案例1：PHP的本地Session信息

### 1.1 问题

通过Nginx调度器负载后端两台Web服务器，实现以下目标：

1. 部署Nginx为前台调度服务器
2. 调度算法设置为轮询
3. 后端为两台LNMP服务器
4. 部署测试页面，查看PHP本地的Session信息

### 1.2 方案

概念：

Session：存储在服务器端，保存用户名、登陆状态等信息。

Cookies：由服务器下发给客户端，保存在客户端的一个文件里。

保存的内容主要包括：SessionID。

实验拓扑环境：

使用4台RHEL7虚拟机，其中一台作为Nginx前端调度器服务器（eth0:192.168.4.5,eth1:192.168.2.5）、两台虚拟机部署为LNMP服务器，分别为Web1服务器（192.168.2.100）和Web2服务器（192.168.2.200），另外一台作为测试用的Linux客户机（192.168.4.10），拓扑如图-2所示。

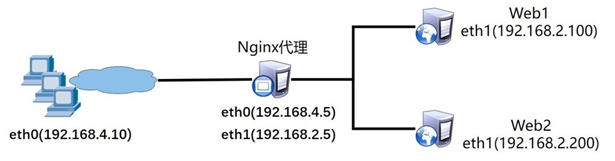


图-2

### 1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署后端LNMP服务器相关软件

注意:以下部署LNMP服务器的操作，需要在两台后端服务器做相同的操作，下面我们以一台Web2服务器（192.168.2.200）为例，对Web1服务器执行相同操作即可。

1）使用yum安装基础依赖包

1. [root@web2 ~]# yum -y install gcc openssl-devel pcre-devel
2. .. ..

2）源码安装Nginx

1. [root@web2 ~]# tar -xf nginx-1.12.2.tar.gz
2. [root@web2 ~]# cd nginx-1.12.2
3. [root@web2 nginx-1.12.2]# ./configure \
4. > --with-http\_ssl\_module
5. [root@web2 nginx-1.12.2]# make && make install

3）安装MariaDB数据库

1. [root@web2 ~]# yum -y install mariadb mariadb-server mariadb-devel

4）安装PHP

1. [root@web2 ~]# yum -y install php php-mysql
2. [root@web2 ~]# yum -y install php-fpm

5）修改Nginx配置文件（修改默认首页与动静分离）

1. [root@web2 ~]# vim /usr/local/nginx/conf/nginx.conf
2. location / {
3. root html;
4. index index.php index.html index.htm;
5. }
6. location ~ \.php$ {
7. root html;
8. fastcgi\_pass 127.0.0.1:9000;
9. fastcgi\_index index.php;
10. # fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;
11. include fastcgi.conf;
12. }

步骤二：启动LNMP服务器相关的服务

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。

1. [root@web2 ~]# systemctl stop httpd                //如果该服务存在，则关闭该服务
2. [root@web2 ~]# /usr/local/nginx/sbin/nginx
3. [root@web2 ~]# ss -utnlp | grep :80
4. tcp    0    0 0.0.0.0:80        0.0.0.0:\*        LISTEN        32428/nginx

2）启动MySQL服务

1. [root@web2 ~]# systemctl start mariadb
2. [root@web2 ~]# systemctl status mariadb

3）启动PHP-FPM服务

1. [root@web2 ~]# systemctl start php-fpm
2. [root@web2 ~]# systemctl status php-fpm

步骤三：部署前端Nginx调度服务器

1）使用源码安装nginx软件（如果Nginx软件包已安装可以忽略此步骤）

1. [root@proxy ~]# yum -y install gcc pcre-devel openssl-devel
2. [root@proxy ~]# tar -xf nginx-1.12.2.tar.gz
3. [root@proxy ~]# cd nginx-1.12.2
4. [root@proxy nginx-1.12.2]# ./configure
5. [root@proxy nginx-1.12.2]# make && make install

2）修改Nginx配置文件，实现代理服务器

Nginx配置文件中，通过upstream定义后端服务器地址池，默认调度策略为轮询，使用proxy\_pass调用upstream定义的服务器地址池：

1. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
2. .. ..
3. upstream webs {
4. server 192.168.2.100:80;
5. server 192.168.2.200:80;
6. }
7. server {
8. listen 80;
9. server\_name localhost;
10. location / {
11. proxy\_pass http://webs;
12. root html;
13. index index.php index.html index.htm;
14. }
15. }

3）重新加载配置文件

1. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload
2. #请先确保nginx是启动状态，否则运行该命令会报错,报错信息如下：
3. [error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)

步骤四：测试环境是否配置成功

1）浏览器访问测试页面验证。

1. [root@client ~]# curl http://192.168.4.5/index.html        //查看是否有数据

步骤五：部署测试页面

1）部署测试页面(Web1服务器）。

测试页面可以参考lnmp\_soft/php\_scripts/php-memcached-demo.tar.gz。

1. [root@web1 ~]# cd lnmp\_soft/php\_scripts/
2. [root@web1 php\_scripts]# tar -xf php-memcached-demo.tar.gz
3. [root@web1 php\_scripts]# cd php-memcached-demo
4. [root@web1 php-memcached-demo]# cp -r \* /usr/local/nginx/html/

2）浏览器直接访问后端服务器的测试页面（Web1服务器）。

1. [root@web1 ~]# firefox http://192.168.2.100            //填写账户信息
2. [root@web1 ~]# cd /var/lib/php/session/            //查看服务器本地的Session信息
3. [root@web1 ~]# ls
4. sess\_ahilcq9bguot0vqsjtd84k7244                        //注意这里的ID是随机的
5. [root@web1 ~]# cat sess\_ahilcq9bguot0vqsjtd84k7244

注意：可用修改index.php和home.php两个文件的内容，添加页面颜色属性，以区别后端两台不同的服务器:<body bgcolor=blue>。

3）部署测试页面(Web2服务器）。

测试页面可以参考lnmp\_soft/php\_scripts/php-memcached-demo.tar.gz。

1. [root@web2 ~]# cd lnmp\_soft/php\_scripts/
2. [root@web2 php\_scripts]# tar -xf php-memcached-demo.tar.gz
3. [root@web2 php\_scripts]# cd php-memcached-demo
4. [root@web2 php-memcached-demo]# cp -r \* /usr/local/nginx/html/

4）浏览器直接访问后端服务器的测试页面（Web2服务器）。

1. [root@web2 ~]# firefox http://192.168.2.100         //填写账户信息
2. [root@web2 ~]# cd /var/lib/php/session/            //查看服务器本地的Session信息
3. [root@web2 ~]# ls
4. sess\_qqek1tmel07br8f63d6v9ch401                        //注意这里的ID是随机的
5. [root@web2 ~]# cat sess\_qqek1tmel07br8f63d6v9ch401

注意：可用修改index.php和home.php两个文件的内容，添加页面颜色属性，以区别后端两台不同的服务器:<body bgcolor=green>。

5）浏览器访问前端调度器测试（不同后端服务器Session不一致）。

真实主机使用google浏览器测试。

1. [root@room9pc01 ~]# google-chrome http://192.168.4.5/index.php
2. //填写注册信息后，刷新，还需要再次注册，说明两台计算机使用的是本地Session
3. //第二台主机并不知道你再第一台主机已经登录，第一台主机的登录信息也没有传递给第二台主机

## 2 案例2：构建memcached服务

### 2.1 问题

本案例要求先快速搭建好一台memcached服务器，并对memcached进行简单的增、删、改、查操作：

* 安装memcached软件，并启动服务
* 使用telnet测试memcached服务
* 对memcached进行增、删、改、查等操作

### 2.2 方案

使用1台RHEL7虚拟机作为memcached服务器（192.168.4.5）。

在RHEL7系统光盘中包含有memcached，因此需要提前配置yum源，即可直接使用yum安装，客户端测试时需要提前安装telnet远程工具。

验证时需要客户端主机安装telnet，远程memcached来验证服务器的功能：

* add name 0 180 10 //变量不存在则添加
* set name 0 180 10 //添加或替换变量
* replace name 0 180 10 //替换
* get name //读取变量
* append name 0 180 10 //向变量中追加数据
* delete name //删除变量
* flush\_all //清空所有
* 提示：0表示不压缩，180为数据缓存时间，10为需要存储的数据字节数量。

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：构建memcached服务

1）使用yum安装软件包memcached

1. [root@proxy ~]# yum -y install memcached telnet
2. [root@proxy ~]# rpm -qa memcached
3. memcached-1.4.15-10.el7\_3.1.x86\_64

2) memcached配置文件（查看即可，不需要修改）

1. [root@proxy ~]# vim /usr/lib/systemd/system/memcached.service
2. ExecStart=/usr/bin/memcached -u $USER -p $PORT -m $CACHESIZE -c $MAXCONN $OPTIONS
3. [root@proxy ~]# vim /etc/sysconfig/memcached
4. PORT="11211"
5. USER="memcached"
6. MAXCONN="1024"
7. CACHESIZE="64"
8. OPTIONS=""

3）启动服务并查看网络连接状态验证是否开启成功：

ss命令可以查看系统中启动的端口信息，该命令常用选项如下：

-a显示所有端口的信息

-n以数字格式显示端口号

-t显示TCP连接的端口

-u显示UDP连接的端口

-l显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p显示监听端口的服务名称是什么（也就是程序名称）

注意：在RHEL7系统中，使用ss命令可以替代netstat，功能与选项一样。

1. [root@proxy ~]# systemctl start memcached
2. [root@proxy ~]# systemctl status memcached
3. [root@proxy ~]# ss -anptu | grep Memcached
4. udp UNCONN 0 0 \*:11211 \*:\* users:(("memcached",pid=12068,fd=28))
5. udp UNCONN 0 0 :::11211 :::\* users:(("memcached",pid=12068,fd=29))
6. tcp LISTEN 0 128 \*:11211 \*:\* users:(("memcached",pid=12068,fd=26))
7. tcp LISTEN 0 128 :::11211 :::\* users:(("memcached",pid=12068,fd=27))

步骤二：使用telnet访问memcached服务器

1)使用telnet连接服务器测试memcached服务器功能，包括增、删、改、查等操作。

1. [root@proxy ~]# telnet 192.168.4.5 11211
2. Trying 192.168.4.5...
3. ……
4. ##提示：0表示不压缩，180为数据缓存时间，3为需要存储的数据字节数量。
5. set name 0 180 3                //定义变量，变量名称为name
6. plj                            //输入变量的值，值为plj
7. STORED
8. get name                        //获取变量的值
9. VALUE name 0 3             //输出结果
10. plj
11. END
12. ##提示：0表示不压缩，180为数据缓存时间，3为需要存储的数据字节数量。
13. add myname 0 180 10            //新建，myname不存在则添加，存在则报错
14. set myname 0 180 10            //添加或替换变量
15. replace myname 0 180 10        //替换，如果myname不存在则报错
16. get myname                    //读取变量
17. append myname 0 180 10        //向变量中追加数据
18. delete myname                    //删除变量
19. flush\_all                        //清空所有
20. quit                            //退出登录

## 3 案例3：LNMP+memcached

### 3.1 问题

沿用练习一和练习二，部署LNMP+memcached网站平台,通过PHP页面实现对memcached服务器的数据操作，实现以下目标：

* 为PHP安装memcache扩展
* 创建PHP页面，并编写PHP代码，实现对memcached的数据操作

### 3.2 方案

如果希望使用PHP来操作memcached，注意必须要为PHP安装memcache扩展（php-pecl-memcache），否则PHP无法解析连接memcached的指令。客户端测试时需要提前安装telnet远程工具。

### 3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建PHP页面，使用PHP语言测试memcached服务

1）部署测试页面

创建PHP首页文档/usr/local/nginx/html/mem.php，测试页面可以参考lnmp\_soft/php\_scripts/mem.php。

注意：192.168.2.5是memcached数据库。

1. [root@web1 ~]# vim /usr/local/nginx/html/mem.php
2. <?php
3. $memcache=new Memcache;                //创建memcache对象
4. $memcache->connect('192.168.2.5',11211) or die ('could not connect!!');
5. $memcache->set('key','test');          //定义变量
6. $get\_values=$memcache->get('key');     //获取变量值
7. echo $get\_values;
8. ?>

2）客户端测试（结果会失败）

客户端使用浏览器访问服务器PHP首页文档，检验对memcached的操作是否成功：

1. [root@web1 ~]# firefox http://192.168.2.100/mem.php

注意：这里因为没有给PHP安装扩展包，默认PHP无法连接memcached数据库，需要给PHP安装扩展模块才可以连接memcached数据库。

3）为web1主机的PHP添加memcache扩展

1. [root@web1 ~]# yum -y install php-pecl-memcache
2. [root@web1 ~]# systemctl restart php-fpm

4）客户端再次测试（结果会成功显示数据结果）

1. [root@web1 ~]# firefox http://192.168.2.100/mem.php

5）为web2主机的PHP添加memcache扩展

1. [root@web2 ~]# yum -y install php-pecl-memcache
2. [root@web2 ~]# systemctl restart php-fpm

## 4 案例4：PHP实现session共享

### 4.1 问题

沿用练习三，通过修改PHP-FPM配置文件，实现session会话共享：

* 配置PHP使用memcached服务器共享Session信息
* 客户端访问两台不同的后端Web服务器时，Session 信息一致

### 4.2 方案

在练习三拓扑的基础上，Nginx服务器除了承担调度器外，还需要担任memcached数据库的角色，并在两台后端LNMP服务器上实现PHP的session会话共享。拓扑结构如图-4所示。

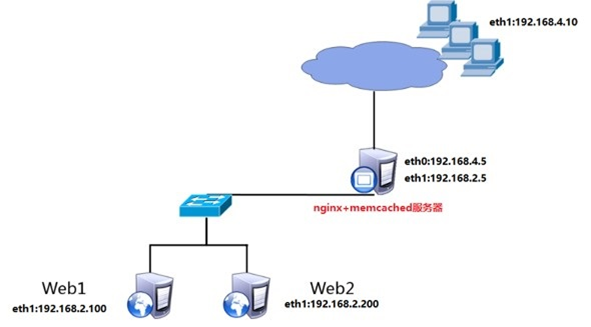


图-4

### 4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：在后端LNMP服务器上部署Session共享

注意：这些操作在两台后端Web服务器上均需要执行，以下操作以Web1（192.168.2.100）服务器为例。

1）修改PHP-FPM配置文件，并重启服务

注意，因为后端两台web服务器(web1,web2)都需要修改配置文件(下面也web1为例)。

1. [root@web1 ~]# vim /etc/php-fpm.d/www.conf            //修改该配置文件的两个参数
2. //文件的最后2行
3. 修改前效果如下:
4. php\_value[session.save\_handler] = files
5. php\_value[session.save\_path] = /var/lib/php/session
6. //原始文件，默认定义Sessoin会话信息本地计算机（默认在/var/lib/php/session）
7. +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
8. 修改后效果如下:
9. php\_value[session.save\_handler] = memcache
10. php\_value[session.save\_path] = "tcp://192.168.2.5:11211"
11. //定义Session信息存储在公共的memcached服务器上，主机参数中为memcache（没有d）
12. //通过path参数定义公共的memcached服务器在哪（服务器的IP和端口）
13. [root@web1 ~]# systemctl restart php-fpm

步骤三：客户端测试

客户端使用浏览器访问两台不同的Web服务器。

操作步骤参考练习一，最终可以获得相关的Session ID信息。

＃进入到ｍemcached

[root@v8 lib]# telnet 201.1.2.5 11211

Trying 201.1.2.5...

Connected to 201.1.2.5.

Escape character is '^]'.

get tl7cmrpddloibagp8e5g8k8mq6 #Session IＤ　进行查询