1. **构建memcached服务**

**memcached是高性能的分布式缓存服务器，用来集中缓存数据库查询结果，减少数据库访问次数，以提高动态Web应用的响应速度**

1. 使用yum安装软件包memcached

[root@proxy ~]# yum -y install memcached

[root@proxy ~]# rpm -qa memcached

1. memcached配置文件（查看即可，不需要修改）

[root@proxy ~]# vim /usr/lib/systemd/system/memcached.service

[root@proxy ~]# vim /etc/sysconfig/memcached

1. 启动服务并查看网络连接状态验证是否开启成功

[root@proxy ~]# systemctl start memcached

[root@proxy ~]# systemctl status memcached

[root@proxy ~]# netstat -anptu | grep memcached

步骤二：使用telnet访问memcached服务器

1. 使用yum安装telnet

2)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。

[root@proxy ~]# telnet 192.168.4.5 11211

##0表示不压缩，180为数据缓存时间妙，3为需要存储的数据字节数量

set name 0 180 3 //定义变量，变量名称为name

plj //输入变量的值，值为plj，空格也算

STORED

get name //获取变量的值

VALUE name 0 3 //输出结果

plj

END

add myname 0 180 10 //新建，myname不存在则添加，存在则报错

set myname 0 180 10 //添加或替换变量

replace myname 0 180 10 //替换，如果myname不存在则报错

get myname //读取变量

append myname 0 180 10 //向变量中追加数据

delete myname //删除变量

stats //查看状态

flush\_all //清空所有

quit //退出登录

**案例2：LNMP+memcached**

部署LNMP实现PHP动态网站架构

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

1. 部署LNMP环境
2. 1）使用yum安装基础依赖包
3. [root@web1 ~]# yum -y install gcc openssl-devel pcre-devel zlib-devel
4. 2）源码安装Nginx
5. [root@web1 ~]# tar -xf nginx-1.12.2.tar.gz
6. [root@web1 ~]# cd nginx-1.12.2
7. [root@web1 nginx-1.12.2]# ./configure \
8. > --with-http\_ssl\_module
9. [root@web1 ~]# yum -y install mariadb mariadb-server mariadb-devel
10. 4）安装PHP（php-fpm软件包在lnmp\_soft中有提供）
11. [root@web1 ~]# yum -y install php php-mysql
12. 3）安装MariaDB数据库
13. 修改Nginx配置文件
14. location ~ \.php$ {
15. root html;
16. fastcgi\_pass 127.0.0.1:9000;
17. fastcgi\_index index.php;
18. # fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;
19. include fastcgi.conf;
20. 启动服务（如果所有服务已经启动，也可以忽略这一步骤）
21. **步骤三：部署前端Nginx调度服务器**
22. **1）使用源码安装nginx软件**
23. 创建PHP页面，使用PHP语言测试memcached服务
24. [root@proxy ~]# yum -y install pcre-devel openssl-devel
25. [root@proxy ~]# tar -xf nginx-1.12.2.tar.gz
26. [root@proxy ~]# cd nginx-1.12.2
27. [root@proxy nginx-1.12.2]# ./configure
28. [root@proxy nginx-1.12.2]# make && make install
29. 2）修改Nginx配置文件
30. 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：
31. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
32. .. ..
33. upstream webs {
34. server 192.168.2.100:80;
35. server 192.168.2.200:80;
36. }
37. server {
38. listen 80;
39. server\_name localhost;
40. location / {
41. proxy\_pass http://webs;
42. root html;
43. index index.php index.html index.htm;
44. }

yum list | grep memcache

yum -y install php-pecl-memcache.x86\_64 #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。

cd lnm\_soft/php\_scripts/

cp mem.php /usr/local/nginx/html/test.php

客户端再次测试

[root@client ~]# firefox http://192.168.4.5/test.php

**PHP的本地Session信息**

1. **在web1和web2部署LNMP环境，实现动静分离功能**

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。

[root@web1 ~]# cd lnmp\_soft/php\_scripts/

[root@web1 php\_scripts]# tar -xf php-memcached-demo.tar.gz

[root@web1 php\_scripts]# cd php-memcached-demo

[root@web1 php-memcached-demo]# cp -a \* /usr/local/nginx/html/ #把向目录下的所有复制到html下

2）浏览器直接访问后端服务器的测试页面（Web1服务器）。

[root@web1 ~]# firefox http://192.168.2.100 //填写账户信息

[root@web1 ~]# cd /var/lib/php/session/ //查看服务器本地的Session信息

[root@web1 ~]# ls

sess\_ahilcq9bguot0vqsjtd84k7244 //注意这里的ID是随机的

[root@web1 ~]# cat sess\_ahilcq9bguot0vqsjtd84k7244

3)web2也部署同样的测试页面

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）。

推荐使用google浏览器测试。

[root@client ~]# google-chrome http://192.168.4.5

//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

//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

#有几台后端服务器访问时就会要填写几次账户和密码信息，用户体验不好

**案例4：PHP实现session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

**步骤一：构建memcached服务**

1. **安装Memcached服务**

[root@proxy ~]# yum -y install memcached

**2）启动服务并查看网络连接状态验证是否开启成功：**

[root@proxy ~]# systemctl restart memcached

[root@proxy ~]# netstat -anptu | grep memcached

**步骤二：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）为PHP添加memcache扩展

注意，因为后端两台web服务器(web1,web2)都需要连接memcached数据库，所以两台主机都需要安装PHP扩展模块(下面也web1为例)。

[root@web1 ~]# yum -y install php-pecl-memcache

**2）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)。

[root@web1 ~]# vim /etc/php-fpm.d/www.conf //修改该配置文件的两个参数

//文件的最后2行

修改前效果如下:

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

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

//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）

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

修改后效果如下:

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

php\_value[session.save\_path] = "tcp://192.168.2.5:11211"

//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）

//通过path参数定义公共的memcached服务器在哪（服务器的IP和端口）

[root@web1 ~]# systemctl restart php-fpm

客户端测试

客户端使用浏览器访问两台不同的Web服务器。

操作步骤与练习三一致，最终可以获得相关的Session ID信息。

#填写一次账户信息和密码就可以访问其它的服务端信息页面