**6\_Operation04Session与cookie部署memcachedSession共享**

**一 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：

部署Nginx为前台调度服务器

调度算法设置为轮询

后端为两台LNMP服务器

部署测试页面，查看PHP本地的Session信息

**1.2 方案**

概念：

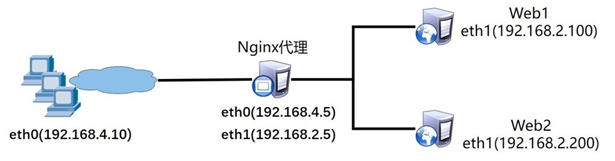
Session：存储在服务器端，保存用户名、登陆状态等信息。

Cookies：保存文件名,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），拓扑如图所示。



**1.3 步骤**

实现此案例需要按照如下步骤进行。

**1.3.1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**

web2 ~]# yum -y install gcc openssl-devel pcre-devel

**2）源码安装Nginx**

web2 ~]# tar -xf nginx-1.12.2.tar.gz

web2 ~]# cd nginx-1.12.2

web2 nginx-1.12.2]# ./configure \

> --with-http\_ssl\_module

web2 nginx-1.12.2]# make && make install

**3）安装MariaDB数据库**

web2 ~]# yum -y install mariadb mariadb-server mariadb-devel

**4）安装PHP**

web2 ~]# yum -y install php php-mysql

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）**

web2 ~]# vim /usr/local/nginx/conf/nginx.conf

location / {

root html;

index index.php index.html index.htm;

}

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

#fastcgi\_param...\_script\_name;

include  **fastcgi.conf**;

}

**1.3.2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。

web2 ~]# systemctl stop httpd #如果该服务存在，则关闭该服务

web2 ~]# /usr/local/nginx/sbin/nginx #启动nginx

web2 ~]# netstat -utnlp | grep :80

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 32428/nginx

**2）启动MySQL服务**

web2 ~]# systemctl start mariadb

web2 ~]# systemctl status mariadb

**3）启动PHP-FPM服务**

web2 ~]# systemctl start php-fpm

web2 ~]# systemctl status php-fpm

**1.3.3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）**

proxy ~]# yum -y install gcc pcre-devel openssl-devel

proxy ~]# tar -xf nginx-1.12.2.tar.gz

proxy ~]# cd nginx-1.12.2

proxy nginx-1.12.2]# ./configure

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：

proxy ~]# vim /usr/local/nginx/conf/nginx.conf

.. ..

**upstream webs {**

**server 192.168.2.100:80;**

**server 192.168.2.200:80;**

**}**  #写在#gzip on;语句上

server {

listen 80;

server\_name localhost;

location / {

**proxy\_pass http://webs;**

root html;

index index.php index.html index.htm;

}

}

**3）重新加载配置文件**

proxy ~]# /usr/local/nginx/sbin/nginx -s reload

**1.3.4 测试环境是否配置成功**

**1）浏览器访问测试页面验证。**

client ~]# curl <http://192.168.4.5/index.html> #查看是否有数据

**1.3.5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。

web1 ~]# cd lnmp\_soft/php\_scripts/

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

web1 php\_scripts]# cd php-memcached-demo

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）**

web1 ~]# firefox <http://192.168.2.100>/index.php #填写账户信息

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

web1 ~]# ls

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

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。

web2 ~]# cd lnmp\_soft/php\_scripts/

web2 php\_scripts]# tar -xf php-memcached-demo.tar.gz

web2 php\_scripts]# cd php-memcached-demo

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）**

web2 ~]# firefox [http://192.168.2.200](http://192.168.2.100) #填写账户信息

web2 ~]# cd /var/lib/php/session/ #查看服务器本地的Session信息

web2 ~]# ls

sess\_qqek1tmel07br8f63d6v9ch401 #注意这里的ID是随机的

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）**

推荐使用google浏览器测试。

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

**二 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(秒)，10为需要存储的数据字节数量。

**2.3 步骤**

**2.3.1构建memcached服务**

**1）使用yum安装软件包memcached(缓存数据库软件,数据放在内存,重启全丢失)**

proxy ~]# yum -y install memcached

proxy ~]# rpm -qa memcached

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）**

proxy ~]# vim /**usr/lib/systemd/system/memcached.service**

ExecStart=/usr/bin/memcached -u $USER -p $PORT -m $CACHESIZE -c $MAXCONN $OPTIONS

proxy ~]# vim /etc/sysconfig/memcached

PORT="11211"

USER="memcached"

MAXCONN="1024"

CACHESIZE="64" #默认M为单位

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。

proxy ~]# systemctl start memcached

proxy ~]# systemctl status memcached

proxy ~]# netstat -anptu | grep memcached

tcp 0 0 0.0.0.0:**11211** 0.0.0.0:\* LISTEN 2839/memcached

tcp 0 0 :::11211 :::\* LISTEN 2839/memcached

udp 0 0 0.0.0.0:11211 0.0.0.0:\* 2839/memcached

udp  0 0 :::11211 :::\* 2839/memcached

proxy ~]# setenforce 0

proxy ~]# firewall-cmd --set-default-zone=trusted

**2.3.2使用telnet访问memcached服务器**

**1）使用yum安装telnet**

proxy ~]# yum -y install 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**

proxy ~]# telnet 192.168.4.5 11211

Trying 192.168.4.5...

……

##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。

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

STORED

get name #获取变量的值

VALUE name 0 3 #输出结果

plj

END

##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。

add myname 0 180 10 #新建，myname不存在则添加，存在则报错,不能修改值

set myname 0 180 10 #添加或替换变量,能新建变量和替换变量的值

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

flush\_all #清空所有

quit #退出登录

**三 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。

**3.3.1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。

web1 ~]# vim /usr/local/nginx/html/mem.php

<?php

$memcache=new Memcache; #创建memcache对象

$memcache->connect('**192.168.2.5**',11211) or die ('could not connect!!');

$memcache->set('key','test');          #定义变量

$get\_values=$memcache->get('key');     #获取变量值

echo $get\_values;

?>

**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：

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PHP添加memcache扩展**

web1 ~]# yum -y install php-pecl-memcache

web1 ~]# systemctl restart php-fpm

**4）客户端再次测试（结果会成功显示数据结果**）

web1 ~]# firefox http://192.168.2.100/mem.php

**四 案例4：PHP实现session共享**

**4.1 问题**

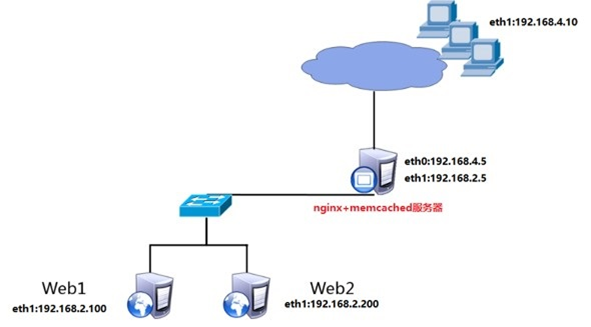
沿用练习三，通过修改PHP-FPM配置文件，实现session会话共享：

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。拓扑结构如图所示。



**4.3 步骤**

**4.3.1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)。

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)。

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/ses

sion）

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）

web1 ~]# systemctl restart php-fpm

**4.3.2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。