operation day 04

================================================

**继续优化nginx**

**nginx并发访问的优化**

**并发：多数用户同时对网站发起访问，并发量支持的越高，说明网站性能越强。**

**默认情况下nginx并发仅仅支持1024个，需要修改配置才能增加**

yum -y install httpd-tools //安装支持压力测试命令的软件包

ab -c 1000 -n 1000 <http://192.168.2.5/> //使用压力测试工具模拟1000人，一共1000次, 相当于每人访问1次，看到100%的提示说明成功，但是增加到2000之后就不行了

-------------------------------

**打开nginx配置文件修改第3行，第13行**

worker\_processes 2; //开启的nginx进程数量，通常是随cpu的核心数一致

worker\_connections 50000; //每个nginx进程支持的并发访问量

之后重加载nginx

---------------------------------

**另外，除了nginx本身对并发量有限制，linux系统本身对文件的访问也有限制，默认情况下linux系统仅允许一个文件同时被打开1024次，普通情况下够用，但是作为网站服务器时，网站页面被N多用户同时访问时相当于同时打开，仅仅支持1024显然不够。**

**永久修改文件访问限制，要在proxy(被测主机)和web1(测试主机)上都配置**

vim /etc/security/limits.conf //修改53、54行，将下列两项内容修改为10万

\* soft nofile 100000

\* hard nofile 100000

**proxy和web1都重启虚拟机才能生效**

ulimit -n //检查系统对文件打开数量的值，默认1024，之后则显示10万

ab -c 2000 -n 2000 <http://192.168.2.5/> 再次测试,成功完成任务

防火墙和selinux都要关

-------------------------------------------

**解决集群主机过多而导致用户重复登陆网站的问题**

**在一个集群中，如果网站需要用户输入用户名和密码登陆之后才能继续访问，那么当用户登陆其中一台集群主机之后随着继续访问页面，请求可能被代理服务器轮询到另外一台服务器上，那么对于另外一台服务器来说用户并没有登陆，想查看登陆之后的页面还需要再次登陆，这样集群主机越多需要客户重复登陆的次数就越多，想要解决该问题就要从Session与Cookies入手**

**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。**

**1，在web1与web2主机部署lnmp环境：**

[root@web1 ~]# yum -y install gcc pcre-devel openssl-devel //安装依赖包

[root@proxy ~]# scp lnmp\_soft.tar.gz 192.168.2.100: //从proxy主机拷贝软件包到weib1的家目录

[root@web1 ~]# tar -xf lnmp\_soft.tar.gz

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

[root@web1 lnmp\_soft]# tar -xf nginx-1.17.6.tar.gz

[root@web1 lnmp\_soft]# cd nginx-1.17.6/

./configure

make

make install

[root@web1 nginx-1.17.6]# yum -y install php php-mysql php-fpm //安装php相关软件包

yum -y install mariadb mariadb-server mariadb-devel //安装数据库相关软件包

[root@web1 nginx-1.17.6]# systemctl start mariadb //启动数据库服务

[root@web1 nginx-1.17.6]# systemctl start php-fpm //启动php-fpm

ss -ntulp | grep :3306 //检查数据库

ss -ntulp | grep :9000 //检查php-fpm

cd /usr/local/nginx //进入nginx安装目录

vim conf/nginx.conf //修改配置文件,实现动静分离,修改65~71行为以下状态

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

include fastcgi.conf;

}

**web2也配置LNMP环境，并配置好nginx的动静分离**

总体思路：

* 拷贝lnmp\_soft.tar.gz软件包到web2
* 释放里面的nginx软件包，安装nginx软件依赖包并编译安装
* 安装LNMP其他软件包，如：数据库、php

4，修改nginx配置文件，实现动静分离

提醒，关闭防火墙，开启LNMP环境依赖的服务

[root@web1 nginx]# sbin/nginx //开启nginx服务

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

cp test.php /usr/local/nginx/html //拷贝测试页面,如果web2

也配置完毕,需要在web2主机做相同测试

使用火狐浏览器访问192.168.2.100/test.php

**2，再测试可以登录的网页**

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

[root@web1 php\_scripts]# tar -xf php-memcached-demo.tar.gz //释放带登录功能的网页

cp -r php-memcached-demo/\* /usr/local/nginx/html/ //拷贝测试页面到nginx中

查看有两个关键页面:

index.php 登录入口页面

home.php 登录后页面

[http://192.168.2.100/ HYPERLINK "http://192.168.2.100/home.php"home HYPERLINK "http://192.168.2.100/home.php".php](http://192.168.2.100/home.php) //直接无法访问,必须先登录

ls /var/lib/php/session/ //登录之后查看服务器session目录可以看到session

文件,以为有该文件,用户再次访问home页面无需重复登录,如果将该目录下的

session文件删除,则用户需要重新登录,用户那边也可以清空浏览器的历史记录

(其中包含Cookies),也需要再次登录.

yum -y reinstall php //如果不小心删除session目录,需要重装php软件包

**3，在proxy主机配置集群，注意该主机不能有动静分离的配置**

upstream web { //在第34行,也就是默认的虚拟主机上面添加集群

server 192.168.2.100:80;

server 192.168.2.200:80;

}

location / {

proxy\_pass [http://web](http://web/); //然后在默认的虚拟主机中的location里调用集群

sbin/nginx -s reload //重新加载配置文件

<http://192.168.2.5/index.php> //之后测试效果，不断刷新页面，会看到web1与web2的登录界面(需要提前在web1与web2的index.php与home.php页面进行标记)

**刷新火狐浏览器的页面,可以尝试下列4种方法**

1,按刷新按钮 2, f5 3, ctrl+f5 4, 鼠标点入地址栏按回车

删除web1与web2的session文件

rm /var/lib/php/session/\*

删除火狐浏览器的历史记录(主要是cookies文件)

然后尝试登录[http://192.168.2.5/index.php HYPERLINK "http://192.168.2.5/index.php，发现还是需要2"，发现还是需要 HYPERLINK "http://192.168.2.5/index.php，发现还是需要2"2](http://192.168.2.5/index.php，发现还是需要2)次才能成功

----------------------------------------------------------

**上述实验由于web1与web2都是在各自的/var/lib/php/session目录中存储session，所以造成客户需要重复登录，为了统一session存储的位置（该存储方式通常被称为session共享），需要安装专门的数据库工具**

**memcache 可以利用内存存储数据的高性能数据库服务**

**1，在proxy安装memcache数据库**

yum -y install memcached //安装数据库工具

systemctl start memcached //启动服务

yum -y install telnet //安装远程登录工具，为了测试memcache

**2，连接测试，验证memcache的增删改查是否正常：**

telnet 127.0.0.1 11211 //使用远程工具连接到本机的11211端口，该

端口就是memcached服务

set abc 0 200 3 //创建(覆盖)变量abc 0是不压缩，保存200秒，存3个字符

replace abc 0 200 3 //覆盖变量abc，如果变量不存在，则覆盖失败

add abc 0 200 3 //添加变量abc，如果变量已经存在，则添加失败

delete abc //删除变量abc

flush\_all //删除所有数据

quit //退出

**3，在web1与web2主机修改session存储的位置，实现session共享**

[root@web1 html]# vim /etc/php-fpm.d/www.conf //修改最后

两行为以下状态

php\_value[session.save\_handler] = memcache //这里改成memcache，表示session的存储不在是本地的普通文件，而是去找memcache

php\_value[session.save\_path] = tcp://192.168.2.5:11211 //这里的路径修改为安装了memcache服务的服务器地址与端口

yum -y install php-pecl-memcache //安装php与memcached服务关联的软件包

systemctl restart php-fpm //重启fpm服务，让上述配置生效

关闭所有服务器的seLinux

最后清空浏览器的历史记录，再访问<http://192.168.2.5/index.php>仅仅登录一次即

可成功

--------------------------------------------------------------------------------------------------

查看 课外练习-sudo.pdf 文档，根据文档内容实现普通用户提升权限