**operation 运维 day 04**

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

**继续优化nginx**

**1，优化Nginx数据包头缓存，支持更长的地址，默认情况下nginx无法**

**支持长地址栏，会报414错误**

打开配置文件，在默认的虚拟主机上面添加：

client\_header\_buffer\_size 200k; //存储地址栏等信息的空间大小

是200k

large\_client\_header\_buffers 4 200k; //如果不够再给4个200k

配置完毕后重加载nginx

cd /root/lnmp\_soft

vim buffer.sh //修改脚本，将里面的4.5修改为2.5(非必须操作)，该脚本的效果是可以产生一个超长的地址

./buffer.sh //执行测试脚本，可以支持超长地址栏并看到页面内容，而不是414报错

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

**2，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**

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

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

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

**临时修改限制**，修改为支持10万(如果已经配置过永久则无需再执行该步骤)

ulimit -Hn 100000

ulimit -Sn 100000

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

**注意：上述的文件访问限制也需要在测试主机(比如web1)上配置，另外所有主机的selinux也要都关闭**

[root@web1 ~]# ab -c 2000 -n 2000 http://192.168.2.5/ //最后再次使用测试主机web1去访问2.5，发现已经成功突破1024的限制

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

**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：**

yum -y install mariadb mariadb-server mariadb-devel

yum -y install php php-mysql php-fpm

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

[root@web1 ~]# systemctl start mariadb

ss -ntulp | grep 3306

ss -ntulp | grep 9000

[root@proxy ~]# scp lnmp\_soft.tar.gz 192.168.2.100: //在proxy主机拷贝文件到web1

[root@web1 ~]# tar -xf lnmp\_soft.tar.gz //回web1释放文件

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

[root@web1 lnmp\_soft]# tar -xf nginx-1.17.6.tar.gz //之后进行nginx部署工作

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

yum -y install gcc pcre-devel openssl-devel //安装依赖包

./configure

make

make install

**2，修改nginx配置文件65~71行，实现动静分离**

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;

}

保存退出，关闭防火墙，开启nginx服务，并使用火狐浏览器测试

**3，在web1与web2主机访问测试页**

cd /root/lnmp\_soft/php\_scripts

tar -xf php-memcached-demo.tar.gz

cp -r php-memcached-demo/\* /usr/local/nginx/html/

使用浏览器访问192.168.2.100/index.php

使用浏览器访问192.168.2.200/index.php

**排错**

**看不到动态页面，可能由于以下原因**

1. 关闭httpd服务
2. 防火墙
3. Php-fpm服务
4. 如果删除nginx重新安装需要杀死nginx进程

然后重新开启服务

1. 缺少依赖包 pcre-devel openssl-devel
2. 配置文件

**4，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 {

server 192.168.2.100:80;

server 192.168.2.200:80;

}

location / {

proxy\_pass http://web;

root html;

<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)

然后将web1与web2的/var/lib/php/session目录中的所有session文件删除，然后再删除浏览器的历史记录(Cookies)，再次登录，发现必须登录两次才能成功。

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

**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**

**1，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**

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 0 100 3 //测试创建或者覆盖abc变量，0是不压缩数据，数据存储时间是100秒，存储3个字符

get abc //获取变量abc的值

add abc 0 100 3 //仅仅创建，如果已经有abc变量则

无法覆盖

replace abc 0 100 3 //覆盖现有变量的数据

delete abc //删除变量

flush\_all //清空所有数据

quit //退出，通过以上命令可以证明memcached服务一切正常

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

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

vim /etc/php-fpm.d/www.conf //修改配置

php\_value[session.save\_handler] = memcache //这里改成memcache

php\_value[session.save\_path] = tcp://192.168.2.5:11211 //session存储位置

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

**所有服务器关闭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