operation 03

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

1. **使用nginx代理功能**

**该功能可以通过组建后台集群提高网站性能**

**1，环境准备**

proxy 2.5

web1 2.100

web2 2.200

检查ip，同网段要互通，检查yum

**在web1、web2主机安装常用软件包与网站服务httpd**

yum -y install vim net-tools bash-completion psmisc httpd

**然后继续准备网站页面**

echo "web1~~" > /var/www/html/index.html //创建测试页面，如果是web2

主机的话要写web2~~

systemctl start httpd //然后将web1、web2的服务开启

systemctl stop firewalld //关闭防火墙

**回到proxy主机**：

[root@proxy nginx]# curl 192.168.2.100 //到proxy主机测试

web1~~

[root@proxy nginx]# curl 192.168.2.200

web2~~

**接下来还原proxy主机的nginx**

[root@proxy nginx]# cd ~/lnmp\_soft/

[root@proxy lnmp\_soft]# killall nginx

[root@proxy lnmp\_soft]# rm -rf /usr/local/nginx/

[root@proxy lnmp\_soft]# rm -rf nginx-1.17.6

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

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

[root@proxy lnmp\_soft]#yum -y install gcc make pcre-devel openssl-devel

[root@proxy nginx-1.17.6]# ./configure

[root@proxy nginx-1.17.6]#make

[root@proxy nginx-1.17.6]#make install

**2，修改nginx配置文件实现集群功能**

[root@proxy nginx]# vim conf/nginx.conf

upstream web { //首先是34~37行，创建集群，集群名称叫web

server 192.168.2.100:80; //这里是集群中的服务器ip与端口

server 192.168.2.200:80; //第二台集群主机

}

server {

listen 80;

。。。。

location / {

proxy\_pass http://web; //在第47行，调用集群

root html;

index index.html index.htm;

}

之后开启nginx或者重加载nginx配置，用curl 192.168.2.5可以看到 web1

或web2页面轮询出现

**二，集群优化**

1. **调节权重定义集群主机任务的分配量**

server 192.168.2.200:80 weight=2; //为性能较强的集群主机配置权重

，权重越大任务的分配量就越大

[root@proxy nginx]# curl 192.168.2.5

web2~~

[root@proxy nginx]# curl 192.168.2.5

web2~~

[root@proxy nginx]# curl 192.168.2.5

web1~~

1. **配置健康检查**

server 192.168.2.200:80 max\_fails=2 fail\_timeout=30; //检测两次如果失败

，则认为集群中的主机故障，之后等待30秒再次测试

sbin/nginx -s reload

测试时，先将web2的httpd服务关闭，回到proxy访问集群页面curl 192.168.2.5

只会显示web1的页面，此时即使将web2的httpd服务开启也无效，因为要等

待30秒

**3，相同客户机访问相同服务器**

upstream web {

ip\_hash; //相同客户机访问相同服务器，让一个客户机访问集群时锁定

一个后台服务器，避免重复登陆的问题

server 192.168.2.100:80;

server 192.168.2.200:80;

}

sbin/nginx -s reload

curl 192.168.2.5 //重加载配置后访问2.5只会看到一个网站的页面

**4，添加down标记**

upstream web {

server 192.168.2.100:80;

server 192.168.2.200:80 down; //down标记可以让集群主机暂时不参与集群活动

}

sbin/nginx -s reload

curl 192.168.2.5

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

**nginx功能**

**网站服务，网站代理（网站业务），四层代理（其他业务）**

**三，使用nginx创建其他业务集群（四层代理）**

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

[root@proxy nginx-1.17.6]# killall nginx

[root@proxy nginx-1.17.6]# rm -rf /usr/local/nginx/

./configure --with-stream --with-http\_stub\_status\_module

//这里的--with-stream 是添加四层代理模块，可以用来创建其

他业务集群，--with-http\_stub\_status\_module是后面实验所需模块

make //编译

make install //安装

cd /usr/local/nginx/

sbin/nginx -V //查看安装模块情况

打开nginx主配置文件，在16行左右(http上面)，添加以下内容

stream { //创建新业务

upstream backend { //创建名叫backend的集群

server 192.168.2.100:22; //集群中的主机使用22端口对外提供服务

server 192.168.2.200:22;

}

server {

listen 12345; //监听端口号

proxy\_pass backend; //调用集群

}

}

sbin/nginx //配置写好之后开启服务或者重加载配置文件

ssh 192.168.2.5 -p 12345 //尝试远程登录,第一次可能会连接集群的某一台主机比如web1

exit //退出

**注意：如果配置没有错误，但无法反复登陆web1与web2，可以按下列方式解决**

[root@proxy nginx]# rm -rf ~/.ssh/known\_hosts //每登录一次之后在proxy中删除记录文件

ssh 192.168.2.5 -p 12345 //再次尝试登录，会连接到另外一台集群主机

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

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系统中可以使用ss命令替代netstat命令，功能一样，选

项一样。

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

**常见nginx问题处理**

**1，404报错**

error\_page 404 /test.jpg; //如果客户访问了不存在的页面就显

示test.jpg的内容

sbin/nginx -s reload //重加载配置

然后找一个图片扔到/usr/local/nginx/html里面，命名为test.jpg

使用浏览器虽随意访问不存在的页面192.168.2.5/XXXX.html 就可以看到之前

那张图片的内容

如果真机是windows可以用mobaXterm传test.jpg文件到nginx的html目录

如果真机是linux可以用scp命令

**2，查看网站后台数据**

--with-http\_stub\_status\_module //该功能利用的模块，之前配置nginx已经安装

打开配置文件

location /status { //在error\_page行上面添加此内容

stub\_status on; //显示后台的数据

allow 192.168.2.5; //只允许2.5查看

deny all; //拒绝其他

}

sbin/nginx -s reload //重加载配置

使用curl 192.168.2.5/status 查看

Active connections：当前活动的连接数量（当前有多少用户访问该网站）。

Accepts：已经接受客户端的连接总数量。

Handled：已经处理客户端的连接总数量。

Requests：客户端发送的请求数量。

Reading：当前服务器正在读取客户端请求头的数量。

Writing：当前服务器正在写响应信息的数量。

Waiting：当前多少客户端在等待服务器的响应。