operation day 03

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

**地址重写相关选项:**

last 不再读其他rewrite

break 不再读其他语句

redirect 临时重定向 状态码 302

permanent 永久重定向 状态码 301

**测试一: 重定向**

cd /usr/local/nginx

echo "nginx A~~" > html/a.html

echo "nginx B~~" > html/b.html

echo "nginx C~~" > html/c.html

cd conf

cp nginx.conf.default nginx.conf //把配置文件恢复默认

cd ..

vim conf/nginx.conf //修改配置

rewrite /a.html /b.html redirect; //在42行添加

然后开启或者重加载nginx服务

rewrite /a.html /b.html permanent; //再次修改配置

重加载nginx服务

curl 192.168.2.5/a.html //此时看到的页面也是b，说明redirect与permanent效果在客户机看来是一样的，但是状态码不一样，对于搜索引擎来说更关心301的

**测试二：**

**last 不再读其他rewrite**

打开主配置文件，在42行修改

rewrite /a.html /b.html last; //如果没有last,看a页面会得到c页面

的内容,加了last之后,就不会再读后面的rewrite了

rewrite /b.html /c.html;

sbin/nginx -s reload //重加载nginx服务

使用火狐访问192.168.2.5/a.html 看到的是b页面

**break 不再读其他语句**

**按下列方式修改配置文件,将默认的location中加入rewrite语句,然后**

**再创建一个新的location也加入rewrite语句**

location / {

rewrite /a.html /b.html break; //如果这里的选项依然用last,则仅仅会阻止

该location之内的rewrite语句,换成break就可以阻止之后的所有rewrite语句

root html;

index index.html index.htm;

}

location /b.html {

rewrite /b.html /c.html;

}

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

**用nginx实现网站代理功能 (组建网站集群)**

**一台服务器的能力是有限的，如果客户访问量比较大，可以利用nginx的代理功能组建集群，集群中的服务器越多集群整体性能就越强**

**1，集群中服务器的准备工作，这里使用web1 与web2**

yum -y install vim

yum -y install bash-completion //tab键补全软件包

yum -y install net-tools //网络工具软件包，支持ifconfig等命令

yum -y install httpd

echo "httpd\_web1~~~" > /var/www/html/index.html //这里

如果是web2主机要改成echo "httpd\_web2~~~"

systemctl restart httpd

systemctl stop firewalld

**之后到proxy主机测试:**

[root@proxy nginx]# curl 192.168.2.100

httpd\_web1~~~

[root@proxy nginx]# curl 192.168.2.200

httpd\_web2~~~

**2，在proxy主机添加创建集群配置**

首先是34~37行

upstream web { //创建nginx集群，名称是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

}

然后在47行

47 proxy\_pass http://web; //调用web集群

sbin/nginx -s reload //重加载nginx服务

使用火狐访问192.168.2.5 不断刷新，看到的页面内容是web1与web2之间切换

或者使用curl 192.168.2.5

**集群优化**

1. **任务量的分配，性能较强的服务器可以多分配工作量**

server 192.168.2.100:80 weight=2; //权重，值越大，工作量越大

**2, 健康检查功能**

server 192.168.2.100:80 max\_fails=2 fail\_timeout=30; //检测2次如

果失败，则认为集群中的服务器故障,认为集群中的服务器故障之

后等待30秒才会再次链接

**3，集群主机需要维护时**

server 192.168.2.100:80 down; 加down标记，使集群服务器暂时不参

与集群的任务轮询

**4，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 //重加载配置

再次访问192.168.2.5，就不会再轮询访问后台集群主机了

传输层 端口号

--with-stream 支持四层代理 可以让nginx组建其他业务(非web)的集群

--with-http\_stub\_status\_module 可以查看网站后台数据

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

make

make install

cd /usr/local/nginx/

sbin/nginx -V //检查安装的版本与模块

**2，创建集群**

打开nginx主配置文件，在16行左右(http上面)，添加以下内容

16 stream { //创建新业务

17 upstream backend { //创建名叫backend的集群

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

19 server 192.168.2.200:22;

20 }

21 server {

22 listen 12345; //监听端口号

23 proxy\_pass backend; //调用集群

24 }

25 }

写完后启动服务或者重新加载配置

[root@proxy nginx]# rm -rf ~/.ssh/known\_hosts //先删除远程连接的主机记录，之

后每次连接测试都先删除该文件

[root@proxy nginx]# ssh 192.168.2.5 -p 12345 //使用12345端口号连接代理服务器

此处使用自己连接自己测试，如果使用client主机就远程连接192.168.4.5

下课休息 15:10回