**operation 运维 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，结束请求

redirect 临时重定向

permanent 永久重定向

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

**测试一：**

**redirect 临时重定向(http状态码302)**

**permanent 永久重定向(http状态码301)**

打开nginx主配置文件，在42行添加

rewrite /a.html /b.html redirect;

然后开启或者重加载nginx服务

curl 192.168.2.5/a.html //此时看到的页面是b

再次打开主配置文件，在42行修改

rewrite /a.html /b.html permanent;

sbin/nginx -s reload //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**

**break 不再读其他语句，结束请求**

[root@proxy nginx]# echo nginx\_web\_c~ > html/c.html //准备c页面

打开主配置文件，在42行修改

rewrite /a.html /b.html;

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c.页面

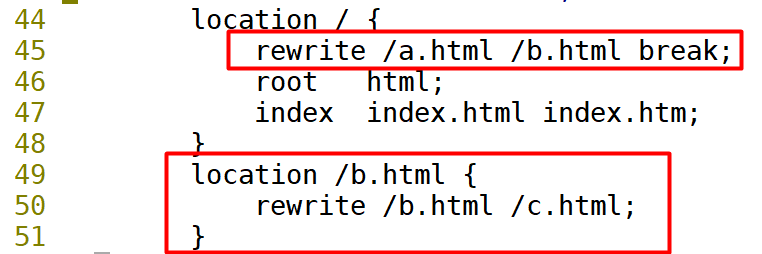
rewrite /a.html /b.html last; //然后再修改配置添加last可以实现不继续跳转

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.html

**再次按下图修改配置文件，删除原有rewrite语句，如果使用last无法阻止第**

**二个location中的rewrite语句，而break可以**



开启或者重加载服务

使用火狐访问192.168.2.5/a.html 看到的是b.html

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

**用nginx实现网站代理功能**

**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**

在web1与web2安装常用工具

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

在web1与web2安装网站服务

yum -y install httpd

echo web1 > /var/www/html/index.html //这里如果是web2主机要改成echo web2

systemctl start httpd

systemctl stop firewalld

使用proxy测试web1与web2

[root@proxy nginx]# curl 192.168.2.100

web1

[root@proxy nginx]# curl 192.168.2.200

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，在集群中的主机ip与端口后面加入以下参数，可**

**以实现不同效果。**

**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;

}

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

**使用nginx实现其他业务的集群**

**通常情况下nginx是搭建网站的工具，还可以组建网站集群，但如果后端的集群服务器跑的不是网站业务，就可以利用--with-stream模块创建非网站业务的集群。**

**1，准备工作**

在proxy主机：停止nginx服务，并删除nginx

回到家目录下的lnmp\_soft的nginx目录重新编译安装nginx，将上述两个模块都添加上，其中--with-stream是四层代理模块可以让nginx组建其他业务(非web)的集群，另外一个模块可以查看网站后台数据（后面的实验需要）。

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

make

make install

sbin/nginx -V //检查安装的版本与模块

sbin/nginx //开启服务

web1与web2的root密码要设置成一致

proxy的家目录下的.ssh目录中known\_hosts文件记录了都远程登录过哪些机器，可以先删除

**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

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

**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。**

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

**Nginx常见问题的处理**

**1，404报错优化：  
客户访问网站时，如果看到了不存在的页面会有404报错的英文提示，这种提示很不友好，可以通过自定义页面改善用户体验**

首先修改配置文件 59行

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

保存退出

找一张图片，内容随意，比如用中文标注"抱歉！您访问的页面不存在" 然后保存成test.jpg格式，然后拷贝到proxy主机

的/usr/local/nginx/html目录下

重新加载nginx配置

使用浏览器足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

location /status { //当用户输入的地址后面跟了/status之后

stub\_status on; //开启网站后台状态信息查看功能

allow 192.168.2.5; //仅仅允许2.5查看

deny all; //拒绝其他所有主机

}

error\_page 404 /test.jpg;

curl <http://192.168.2.5/status> //查看测试，仅仅2.5可以看

页面中各个字段的含义：

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。

**2，配置nginx缓存数据的功能：**

**客户在访问服务器时，有可能随时需要重复的文件，如果反复从服务器获取会造成资源浪费，还耽误时间，可以通过配置将常用的或者比较大的文件在用户访问一次之后就缓存在客户机中，下次客户访问时不用再找服务器而是从本机获取。**

修改配置文件 在默认的location的下面添加一个新的location

location ~\* \.(jpg|png|mp4|html)$ { //当用户访问的是这几种类型的

文件时

expires 30d; //都会缓存在客户机上30天

}

然后使用火狐浏览器，先清空历史记录，然后地址栏输入about:cache

查看disk文件的列表，找到被访问文件看最后倒数第2列信息显示多

久超时

