**实验环境：**

**主机tian1（192.168.67.100）即作为负载均衡设备，又作为数据库,此实验没有安装mysql数据库。**

**主机tian2（192.168.67.102）和tian3（192.168.67.101）作为web服务设备（安装的是tomcat），tian2在/usr/local/tomcat/webapps/ROOT/tian2路径下放的是my\_love.jpg图片文件，tian3在/usr/local/tomcat/webapps/ROOT/tian3访问是一个index.html。两台web服务器的域名都是www.tian2.com（本实验中的域名需要在hosts文件中绑定才能访问，但是在公网上能解析的域名不必绑定hosts，而且本机绑定hosts有个缺点，上面的会覆盖下面）。**

**实验目的：**

**需要根据访问文件的不同，tian1给调度到不同的web服务器之上。**

**第一步 ：安装及配置**

**1.1 :安装**

**tian1上面安装及配置haproxy，选择安装的目录是/usr/local：**

**haproxy的下载网址：**[**http://www.haproxy.org/download/1.8/src/**](http://www.haproxy.org/download/1.8/src/)

[**选择的版本是：haproxy-1.8.3.tar.gz**](http://www.haproxy.org/download/1.8/src/)

[**本人安装软件的习惯是安装在/usr/local目录下：**](http://www.haproxy.org/download/1.8/src/)

[**tar -zxvf haproxy-1.8.3.tar.gz**](http://www.haproxy.org/download/1.8/src/)

[**cd haproxy-1.8.3**](http://www.haproxy.org/download/1.8/src/)

[**make TARGET=linux26 PREFIX=/usr/local/haproxy**](http://www.haproxy.org/download/1.8/src/)

[**make install PREFIX=/usr/local/haproxy**](http://www.haproxy.org/download/1.8/src/)

**1.2 添加启动脚本：**

**vim /etc/init.d/haproxy**

**#! /bin/sh**

**set -e**

**PATH=/sbin:/bin:/usr/sbin:/usr/bin:/home/ha/haproxy/sbin**

**PROGDIR=/home/ha/haproxy**

**PROGNAME=haproxy**

**DAEMON=$PROGDIR/sbin/$PROGNAME**

**CONFIG=$PROGDIR/conf/$PROGNAME.cfg**

**PIDFILE=$PROGDIR/conf/$PROGNAME.pid**

**DESC="HAProxy daemon"**

**SCRIPTNAME=/etc/init.d/$PROGNAME**

**# Gracefully exit if the package has been removed.**

**test -x $DAEMON || exit 0**

**start()**

**{**

**echo -e "Starting $DESC: $PROGNAME\n"**

**$DAEMON -f $CONFIG**

**echo "."**

**}**

**stop()**

**{**

**echo -e "Stopping $DESC: $PROGNAME\n"**

**haproxy\_pid="$(cat $PIDFILE)"**

**kill $haproxy\_pid**

**echo "."**

**}**

**restart()**

**{**

**echo -e "Restarting $DESC: $PROGNAME\n"**

**$DAEMON -f $CONFIG -p $PIDFILE -sf $(cat $PIDFILE)**

**echo "."**

**}**

**case "$1" in**

**start)**

**start**

**;;**

**stop)**

**stop**

**;;**

**restart)**

**restart**

**;;**

**\*)**

**echo "Usage: $SCRIPTNAME {start|stop|restart}" >&2**

**exit 1**

**;;**

**esac**

**exit 0**

**chmod 755 /etc/init.d/haproxy**

**如若脚本启动的时候出错，可以修改脚本**

**也可以指定指定配置文件自行启动./haproxy -f /usr/local/haproxy/conf/haproxy.cfg**

**1.3：修改haproxy配置文件：**

**vim /usr/local/haproxy/conf/haproxy.cfg**

**global #全局属 性**

**daemon #以daemon方式在后台运行**

**maxconn 256 #最大同时256连接**

**pidfile /usr/local/haproxy/conf/haproxy.pid #指定保存HAProxy进程号的文件**

**log 127.0.0.1 local0 notice**

**log 127.0.0.1 local1 warning**

**defaults #默认参数**

**mode http #http模式**

**log global**

**option http-keep-alive #使用keepAlive连接**

**option forwardfor #记录客户端IP在X-Forwarded-For头域中**

**option httplog #开启httplog，HAProxy会记录更丰富的请求信息**

**timeout connect 5000ms #连接server端超时5s**

**timeout client 50000ms #客户端响应超时50s**

**timeout server 50000ms #server端响应超时50s**

**timeout http-request 20000ms #从连接创建开始到从客户端读取完整HTTP请求的超时时间，用于避免类DoS攻击**

**option httpchk GET /healthCheck.html #定义默认的健康检查策略**

**frontend http-in #前端服务http-in**

**bind \*:9091 #监听9091端口**

**acl url\_tian2 path\_beg -i /tian2/ #定义ACL，当uri以/tian2/开头时，ACL[url\_tian2]为true**

**acl url\_tian3 path\_beg -i /tian3/ #同上，url\_tian3**

**use\_backend tian2 if url\_tian2 #当[url\_tian2]为true时，定向到后端服务群tian2中**

**use\_backend tian3 if url\_tian3 #当[url\_tian3]为true时，定向到后端服务群tian3中**

**default\_backend tian2 #请求转发至名为"tian2"的后端服务**

**default\_backend tian3 #请求转发至名为"tian3"的后端服务**

**backend tian2 #后端服务tian2**

**mode http**

**balance roundrobin #使用RR负载均衡算法**

**cookie HA\_STICKY\_tian2 insert indirect nocache #会话保持策略，insert名为"HA\_STICKY\_tian2"的cookie**

**server tian2 192.168.67.102:8080 cookie tian2 maxconn 300 check inter 2000ms rise 2 fall 3**

**#定义后端server[tian2.srv1]，请求定向到该server时会在响应中写入cookie值[tian2.srv1]**

**#针对此server的maxconn设置为300**

**#应用默认健康检查策略，健康检查间隔和超时时间为2000ms，两次成功视为节点UP，三次失败视为节点DOWN**

**#server tian3 192.168.67.101:8080 cookie tian3 maxconn 300 check**

**#同上，inter 2000ms rise 2 fall 3是默认值，可以省略**

**backend tian3 #定义后端服务群tian3**

**balance roundrobin**

**cookie HA\_STICKY\_tian3 insert indirect nocache**

**#server tian2 192.168.67.102:8080 cookie tian2 maxconn 300 check**

**server tian3 192.168.67.101:8080 cookie tian3 maxconn 300 check**

**listen stats #定义监控页面**

**bind \*:1080 #绑定端口1080**

**stats refresh 30s #每30秒更新监控数据**

**stats uri /stats #访问监控页面的uri**

**stats realm HAProxy\ Stats #监控页面的认证提示**

**stats auth admin:admin #监控页面的用户名和密码**

**启动haproxy：**

**cd /usr/local/haproxy/sbin**

**./haproxy -f /usr/local/haproxy/conf/haproxy.cfg**

**1.5：用logrotate对haproxy进行日志切割：**

**通过rsyslog输出的日志是不会进行切分的，所以需要依靠Linux提供的logrotate来进行切分工作**

**1.5.1、修改haproxy.cfg**

**在global域和defaults域中添加红色部分的配置。**

**global**

**...**

**log 127.0.0.1 local0 notice**

**log 127.0.0.1 local1 warning**

**...**

**defaults**

**...**

**log global**

**...**

**意思是将notice（及以上）的日志推送到rsyslog的local1接口，将warn（及以上）的日志推送到rsyslog的local1接口，并且所有fronted都默认使用global中的日志配置**

**1.5.2.为rsyslog添加haproxy日志的配置**

**vim /etc/rsyslog.d/haproxy.conf**

**$ModLoad imudp**

**$UDPServerRun 514**

**$FileCreateMode 0644 #日志文件的权限**

**$FileOwner ha #日志文件的owner**

**local0.\* /var/log/haproxy.log #local0接口对应的日志输出文件**

**local1.\* /var/log/haproxy\_warn.log #local1接口对应的日志输出文件**

**1.5.3.修改rsyslog的启动参数**

**vim /etc/sysconfig/rsyslog**

**# Options for rsyslogd**

**# Syslogd options are deprecated since rsyslog v3.**

**# If you want to use them, switch to compatibility mode 2 by "-c 2"**

**# See rsyslogd(8) for more details**

**SYSLOGD\_OPTIONS="-c 2 -r -m 0"**

**1.5.4.重启rsyslog和haproxy**

**service rsyslog restart**

**service haproxy restart**

**此时就应该能在/var/log目录下看到haproxy的日志文件了**

**1.5.5.用logrotate进行日志切分**

**通过rsyslog输出的日志是不会进行切分的，所以需要依靠linux提供的logrotate来进行切分工作**

**mkdir /root/logrotate**

**vi /root/logrotate/haproxy**

**/usr/lcoal/haproxy/conf/haproxy.log /usr/local/haproxy/conf/haproxy\_warn.log { #切分的两个文件名**

**daily #按天切分**

**rotate 7 #保留7份**

**create 0644 ha ha #创建新文件的权限、用户、用户组**

**compress #压缩旧日志**

**delaycompress #延迟一天压缩**

**missingok #忽略文件不存在的错误**

**dateext #旧日志加上日志后缀**

**sharedscripts #切分后的重启脚本只运行一次**

**postrotate #切分后运行脚本重载rsyslog，让rsyslog向新的日志文件中输出日志**

**/bin/kill -HUP $(/bin/cat /var/run/syslogd.pid 2>/dev/null) &>/dev/null**

**endscript**

**}**

**并设置在crontab中运行**

**0 0 \* \* \* /usr/bin/logrorate /root/logrorate/haproxy**

**第二步：在tian2和tian3服务器上分别安装tomcat：**

**cd /usr/local**

**wget** [**http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz**](http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz)

[**tar -zxvf apache-tomcat-8.0.24.tar.gz**](http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz)

[**mv apache-tomcat-8.0.24 /usr/local/tomcat**](http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz)

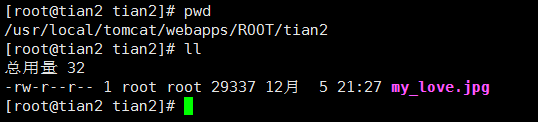
[**cd tomcat/**](http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz)

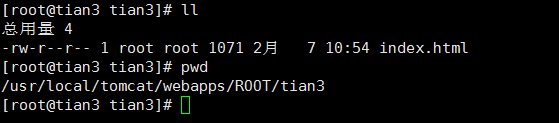
[**bin/startup.sh**](http://archive.apache.org/dist/tomcat/tomcat-8/v8.0.24/bin/apache-tomcat-8.0.24.tar.gz)

**netstat -lntp| grep 8080**

**tomcat默认的发布路径是：/usr/local/tomcat/webapps/ROOT**

**在tian2上创建/usr/local/tomcat/webapps/ROOT/tian2，在tian3上创建/usr/local/tomcat/webapps/ROOT/tian3以此来区别访问的文件不同**





**修改tomcat的访问端口和添加域名，两个的配置文件相同，我只复制了一份：**

**....**

**Define a non-SSL/TLS HTTP/1.1 Connector on port 8080**

**-->**

**<Connector port="8080" protocol="HTTP/1.1"**

**connectionTimeout="20000"**

**redirectPort="8443" />**

**<!-- A "Connector" using the shared thread pool-->**

**.....**

**resourceName="UserDatabase"/>**

**</Realm>**

**<Host name="www.tian2.com" appBase="webapps"**

**unpackWARs="true" autoDeploy="true">**

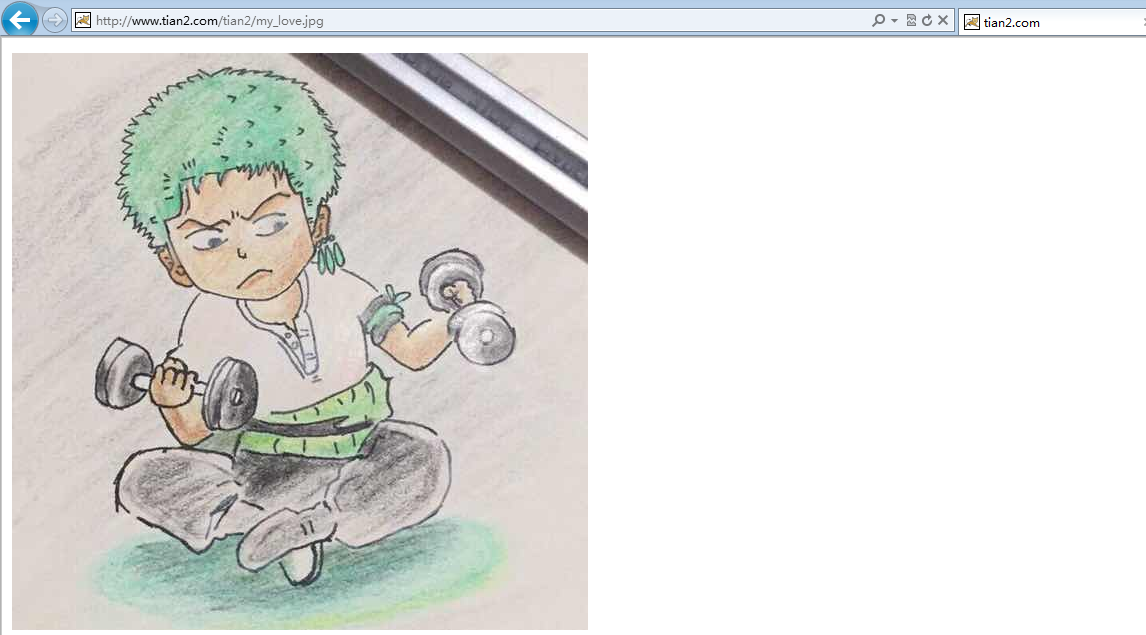
**<!-- SingleSignOn valve, share authentication between web applications**

**.....**

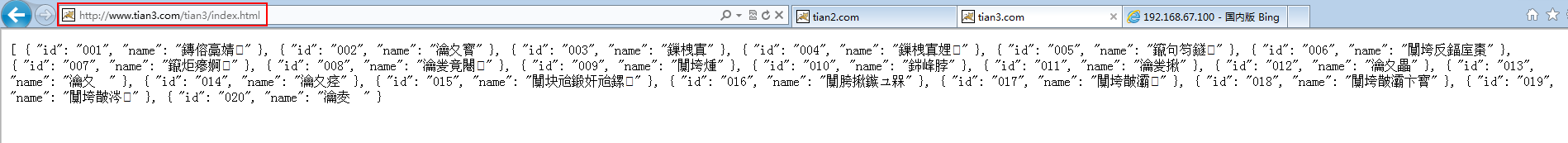
**因为自己的域名未添加备案，不能得到解析，因为需要添加hosts**



[**访问tian2上面的图片文件：**](http://www.haproxy.org/download/1.8/src/)



**访问tian3上面的html文件：**



**需要在tian2和tian3上面分别配置相同的域名:www.tian2.com，在访问到不同路径的时候分别访问到tian2或者tian3**

**这个实验还是存在问题，本实验的域名www.tian2.com需要在hosts文件中绑定才能访问域名，但是在公网上能解析的域名不必绑定hosts，而且本机绑定hosts有个缺点，上面的会覆盖下面。所以只能访问到192.168.67.102这台设备，目前还没有想到较好的办法。但是如果是公网上的域名就不牵扯，配置在哪台服务器上面，就会解析到哪台。**