HAProxy LB

1、它的工作原理是proxy代理。客户端把请求发送到HAProxy后，HAProxy再替用户发送请求到web服务器，web服务器响应HAProxy，把页面发给HAProxy，HAProxy再把页面发给客户端

2、配置

node4作为调度器

1. 装包

]# yum -y install haproxy

1. 修改配置文件

vim /etc/haproxy/haproxy.cfg

//把frontend和backend结合到一起的完整的声明用listen，删除frontend和backend配置

listen stats //后台统计页面 ,名字自定义

bind 0.0.0.0:1080 //监听端口

stats refresh 30s //统计页面30秒自动刷新

stats uri /monitor //统计页面url ，可以自定义

stats realm Haproxy Manager //统计页面密码框上提示的东西

stats auth admin:admin //统计页面用户名：密码

listen webservers 0.0.0.0:80 //监听后端的多台web服务器

cookie SERVERID rewrite

balance roundrobin //调度算法

server web1 192.168.4.2:80 cookie app1 check inter 2000 rise 2 fall 5

server web2 192.168.4.3:80 cookie app2 check inter 2000 rise 2 fall 5

//check inter 2000 每2000毫秒做一次健康检查

//rise 2 连续2次成功就认为是健康的 fall 5 连续5次失败就认为是故障

3）systemctl restart haproxy

4）测试访问

<http://192.168.4.4:1080/monitor> //访问后台管理界面

<http://192.168.4.4> //访问web调度器

1. 系统日志

vim /etc/rsyslog.conf

man rsyslog.conf

构成

Facility.priority 相当于是 服务.优先级

facility可以是这些关键字：

auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, security (same as auth), syslog, user, uucp 以及local0 到local7

local0 到local7是预留出来的接口，供第三方应用调用

priority可以是这些关键字：

debug, info, notice, warning, warn (same as warning), err, error (same as err), crit, alert, emerg, panic (same as emerg)

debug是最轻级别，panic 是最严重的级别，如果日志记录优先级是info，表示比info严重的日志都需要记录

6）配置本机接受通过网络发来的日志，默认是不接受的

vim /etc/rsyslog.conf //打开TCP／UDP注释

#Provides UDP syslog reception

$ModLoad imudp

$UDPServerRun 514

#Provides TCP syslog reception

$ModLoad imtcp

$InputTCPServerRun 514

systemctl restart syslog

1. 查看日志

tail -f /var/log/message //该日志记录系统大部分的日志

访问HAProxy,查看日志输出信息

keepalived HA

Keepalived是Linux下一个轻量级别的高可用解决方案,用来监控集群系统中各个服务节点的状态，它根据TCP/IP参考模型的第三、第四层、第五层交换机制检测每个服务节点的状态，如果某个服务器节点出现异常，或者工作出现故障，Keepalived将检测到，并将出现的故障的服务器节点从集群系统中剔除，这些工作全部是自动完成的，不需要人工干涉，需要人工完成的只是修复出现故障的服务节点，后来Keepalived又加入了VRRP的功能，VRRP（虚拟路由冗余协议)出现的目的是解决静态路由出现的单点故障问题，通过VRRP可以实现网络不间断稳定运行，因此Keepalvied 一方面具有服务器状态检测和故障隔离功能，另外一方面也有HA cluster功能

通过VRRP实现高可用性的工作原理

在网络层： Keepalived在网络层采用最常见的工作方式是通过ICMP协议向服务器集群中的每一个节点发送一个ICMP数据包(有点类似与Ping的功能)，如果某个节点没有返回响应数据包，那么认为该节点发生了故障，Keepalived将报告这个节点失效，并从服务器集群中剔除故障节点；

Keepalived在传输层里利用了TCP协议的端口连接和扫描技术来判断集群节点的端口是否正常，比如对于常见的WEB服务器80端口。或者SSH服务22端口，Keepalived一旦在传输层探测到这些端口号没有数据响应和数据返回，就认为这些端口发生异常，然后强制将这些端口所对应的节点从服务器集群中剔除掉；

用户可以通过自定义Keepalived工作方式，例如：可以通过编写程序或者脚本来运行Keepalived，而Keepalived将根据用户的设定参数检测各种程序或者服务是否允许正常，如果Keepalived的检测结果和用户设定的不一致时，Keepalived将把对应的服务器从服务器集群中剔除；

Keepalived 是运行在lvs 之上，它的主要功能是实现真实机的故障隔离及负载均衡器间的失败切换，提高系统的可用性；LVS提 供负载均衡，keepalived提供健康检查，故障转移，提高系统的可用性

VRRP：虚拟冗余路由协议，IETF公共标准

HSRP：热备份路由协议，思科私有

搭建高可用的服务器集群

1. node2、node3提供web服务，由于没有过多的用户访问，只有一台服务器工作，另一台起到备份的作用
2. 在node2、node3上安装服务

]# yum -y install keepalived

1. 修改配置文件

7.2以前用slave，之后用backup

两台服务器优先级最好不要挨在一起，方便增减服务器

1）node2操作：

]# vim /etc/keepalived/keepalived.conf

#vrrp\_strict //防止开防火墙

vrrp\_instance VI\_1 {

state MASTER //辅助写BACKUP

interface eth0

virtual\_router\_id 51 //虚拟路由器ID号主从都是一样，代表在同一组

priority 100 //优先级

advert\_int 1 //每隔一秒发送心跳

authentication { //集群成员共享密码

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress { //VIP

192.168.4.200

}

}

]# systemctl restart keepalived

]# ip address show eth0 ip a s eth0 多出192.168.4.200

1. node3操作：

#vrrp\_strict

vrrp\_instance VI\_1 {

state BACKUP //辅助写BACKUP

interface eth0

virtual\_router\_id 51 //虚拟路由器ID号主从都是一样，代表在同一组

priority 90 //优先级

advert\_int 1 //每隔一秒发送心跳

authentication { //集群成员共享密码

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress { //VIP

192.168.4.200

}

}

]# systemctl restart keepalived

]# ip address show eth0

双主配置：两台主机，两个服务，每个主机是一个服务的主,配置文件配置互为主从

搭建高可用、负载均衡的web集群

1. 新建node5虚拟机作为额外的lvs服务器配置。虽然有两台调度器，也是DR模式。
2. 在web服务器lo上配置VIP
3. 调整web服务器的内核参数
4. 调度器上不要再手工的为eth0配置VIP了，因为VIP出现在活跃的调度器上，活跃设备由keepalived决定。

5、在node4、node5上安装lvs

]# yum -y install ipvsadm

1. 在node4、node5上安装keepalived

]# yum -y install keepalived

1. 在node4上配置keepalived

global\_defs {

notification\_email {

root@localhost //收件人

}

notification\_email\_from [admin@tedu.cn](mailto:admin@tedu.cn) //发件人

smtp\_server 127.0.0.1 //邮件服务器地址

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

vrrp\_skip\_check\_adv\_addr

#vrrp\_strict //注释

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

----------------------配置高可用----------------------

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 51

priority 150

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.4.100 //VIP在这里指定

}

}

----------------------配置负载均衡----------------------

virtual\_server 192.168.4.100 80{ //相当于lvs -A

delay\_loop 6

lb\_algo rr

lb\_kind DR //LVS-DR模式

persistence\_timeout 50

//50秒之内相同客户端总是访问同一台服务器，减少TCP连接，提高效率

protocol TCP

real\_server 192.168.4.2 80 { //相当于lvs -a

weight 1

TCP\_CHECK { //健康检查TCP\_CHECK后面一定要有空格

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3 //等3秒之后重试连接

}

}

real\_server 192.168.4.3 80 { //相当于lvs -a

weight 1

TCP\_CHECK { //健康检查TCP\_CHECK后面一定要有空格

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3 //等3秒之后重试连接

}

}

}

剩余配置删除

1. 启用服务、验证

]# systemctl start keepalived

]# ipvsadm -Ln //重启服务就能查看到规则

]# ip a s eth0 //查看192.168.4.100

LVS:基于四层的转发，在软件转发效率是最高的，可以达到F5的60％的性能,不支持正则，不能做动静分离，支持Session的直接保持

HAProxy其次，支持两种代理模式：TCP（四层）和HTTP（七层），不能做Web服务器即Cache

nginx：是基本七层的转发，可以识别URL，支持正则表达式过滤URL，Nginx还能做Web服务器即Cache功能。不支持Session的直接保持

三大主流软件负载均衡器适用业务场景：  
1、网站建设初期，可以选用Nigix/HAproxy作为反向代理负载均衡（或者流量不大都可以不选用负载均衡），因为其配置简单，性能也能满足一般的业务场景。如果考虑到负载均衡器是有单点问题，可以采用Nginx+Keepalived/HAproxy+Keepalived避免负载均衡器自身的单点问题。  
2、网站并发达到一定程度之后，为了提高稳定性和转发效率，可以使用LVS、毕竟LVS比Nginx/HAproxy要更稳定，转发效率也更高。不过维护LVS对维护人员的要求也会更高，投入成本也更大。

注：Niginx与Haproxy比较：Niginx支持七层、用户量最大，稳定性比较可靠。Haproxy支持四层和七层，支持更多的负载均衡算法，支持session保存等。具体选型看使用场景，目前来说Haproxy由于弥补了一些Niginx的缺点用户量也不断在提升。

vrrp\_instance VI\_1 {

state MASTER

interface eth1

virtual\_router\_id 51

priority 150

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.2.200/24 dev eth1

}

}

keepalived + haproxy keepalived监控两台haproxy，设置VIP192.168.2.200，该VIP自动创建