Nginx\_Master: 192.168.1.103 提供负载均衡

Nginx\_BackUp: 192.168.1.104 负载均衡备机

Nginx\_VIP\_TP: 192.168.1.108 网站的 VIP 地址(虚拟 IP)

原理：

VIP 是外网访问的IP地址，通过 keepalived 设置，以及 VRRP 将 VIP 绑定到主机和备机上，通过权重实现控制。当主机挂掉后，keepalived 释放对主机的控制，备机接管VIP。

扩展：

主机和备机可进一步延伸，它们只通过 Nginx 提供负载均衡。再让其它的机器提供真实的 web 服务。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**安装 Nginx (省略)**

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

<http://blog.sina.com.cn/s/blog_5f54f0be0100yqm7.html>

**Keepalived 安装**

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

地址: <http://www.keepalived.org/download.html>

wget http://www.keepalived.org/software/keepalived-1.2.7.tar.gz

tar -zxf keepalived-1.2.7.tar.gz

cd keepalived-1.2.7

64 位系统：

./configure --sysconf=/etc --prefix=/usr/local/keepalived --with-kernel-dir=/usr/src/kernels/2.6.32-358.2.1.el6.x86\_64/

或 32 位系统:

./configure --sysconf=/etc --prefix=/usr/local/keepalived --with-kernel-dir=/usr/src/kernels/2.6.32-358.6.2.el6.i686/

可通过 getconf LONG\_BIT 得到系统位数。

**参数解释:**

--sysconf 指定了配置文件的地址.即：/etc/keepalived/keepalived.conf

--prefix 指定了安装目录

--with-kernel-dir 指定使用内核源码中的头文件,即 include 目录.只有使用 LVS 时才需要这个参数，其它的时候不需要。

如果报错:

configure: error: Popt libraries is required

解决:

yum install popt-devel

再 configue .成功后提示：

Keepalived configuration

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

Keepalived version       : 1.2.7

Compiler                 : gcc

Compiler flags           : -g -O2

Extra Lib                : -lpopt -lssl -lcrypto

Use IPVS Framework       : Yes

IPVS sync daemon support : Yes

IPVS use libnl           : No

Use VRRP Framework       : Yes

Use VRRP VMAC            : Yes

SNMP support             : No

Use Debug flags          : No

**安装：**

make

make install

**设置成为服务并开机启动:**

cp /usr/local/keepalived/sbin/keepalived /usr/sbin/

/etc/rc.d/init.d/keepalived status

chkconfig --add keepalived

chkconfig keepalived on

**设置主机上的配置文件内容：**

*vi /etc/keepalived/keepalived.conf*

! Configuration File for keepalived

global\_defs {

   notification\_email {

     sunyu@easymobi.cn

     wuxuegang.123@163.com

   }

   notification\_email\_from pub@easymobi.cn

   smtp\_server 127.0.0.1

   smtp\_connect\_timeout 30

   router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

    state MASTER

    interface eth0

    virtual\_router\_id 51

    # 此处是主 Nginx 的 IP 地址.

    mcast\_src\_ip 192.168.1.103

    # 该机的 priority(优先) 为 100

    priority 100

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111222

    }

    virtual\_ipaddress {

        192.168.1.108

    }

}

前面的结构那里已经规定好了 VIP 和 主备机的 IP, 所以这里按上面的填。

**备机的配置文件:**

! Configuration File for keepalived

global\_defs {

   notification\_email {

     sunyu@easymobi.cn

     wuxuegang.123@163.com

   }

   notification\_email\_from pub@easymobi.cn

   smtp\_server 127.0.0.1

   smtp\_connect\_timeout 30

   router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

    state SLAVER

    interface eth0

    virtual\_router\_id 51

    # 此处是备 Nginx 的 IP 地址.

    mcast\_src\_ip 192.168.1.104

    # 该机的 priority(优先) 为 99

    priority 99

    advert\_int 1

    authentication {

        auth\_type PASS

        auth\_pass 1111222

    }

    virtual\_ipaddress {

        192.168.1.108

    }

}

这时候 ping 192.168.1.108 是不通的.

然后在两台机器上分别启动 keepalived 服务

这时候再 ping 192.168.1.108 .通了.

实际上这时候 108 是被绑到主机上的。在主机上：

查看系统日志:

*tailf /var/log/messages*

May 29 18:32:16 localhost Keepalived\_vrrp[27731]: Opening file '/etc/keepalived/keepalived.conf'.

May 29 18:32:16 localhost Keepalived\_vrrp[27731]: Configuration is using : 62906 Bytes

May 29 18:32:16 localhost Keepalived\_vrrp[27731]: Using LinkWatch kernel netlink reflector...

May 29 18:32:16 localhost Keepalived\_healthcheckers[27729]: Using LinkWatch kernel netlink reflector...

May 29 18:32:16 localhost Keepalived\_vrrp[27731]: VRRP sockpool: [ifindex(2), proto(112), fd(11,12)]

May 29 18:32:17 localhost Keepalived\_vrrp[27731]: VRRP\_Instance(VI\_1) Transition to MASTER STATE

May 29 18:32:18 localhost Keepalived\_vrrp[27731]: VRRP\_Instance(VI\_1) Entering MASTER STATE

May 29 18:32:18 localhost Keepalived\_vrrp[27731]: VRRP\_Instance(VI\_1) setting protocol VIPs.

May 29 18:32:18 localhost Keepalived\_vrrp[27731]: VRRP\_Instance(VI\_1) Sending gratuitous ARPs on eth0 for 192.168.1.108

May 29 18:32:18 localhost Keepalived\_healthcheckers[27729]: Netlink reflector reports IP 192.168.1.108 added

可以看到.**VRRP**(虚拟路由冗余协议)已经启动.我们可以通过命令 ip addr 来检查主 Nginx 上的 IP 分配情况.

*[root@localhost ~]# ip addr*

1: lo: mtu 16436 qdisc noqueue state UNKNOWN

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    inet 127.0.0.1/8 scope host lo

    inet6 ::1/128 scope host

       valid\_lft forever preferred\_lft forever

2: eth0: mtu 1500 qdisc mq state UP qlen 1000

    link/ether 00:15:c5:ef:53:8c brd ff:ff:ff:ff:ff:ff

    inet 192.168.1.103/25 brd 192.168.1.255 scope global eth0

    inet 192.168.1.108/32 scope global eth0

    inet6 fe80::215:c5ff:feef:538c/64 scope link

       valid\_lft forever preferred\_lft forever

3: eth1: mtu 1500 qdisc noop state DOWN qlen 1000

    link/ether 00:15:c5:ef:53:8e brd ff:ff:ff:ff:ff:ff

可以看到 VIP 地址已经绑定到主 Nginx 机器上: inet 192.168.1.108/32 scope global eth0

**我们通过 tcpdump 抓包：**

*[root@localhost ~]# tcpdump vrrp*

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes

13:38:27.797982 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:28.794693 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:29.794518 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:30.798581 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:31.795902 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:32.804050 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:33.801191 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

13:38:34.798793 IP htuidc.bgp.ip > vrrp.mcast.net: VRRPv2, Advertisement, vrid 51, prio 100, authtype simple, intvl 1s, length 20

这样，一个 Nginx + Keepalived 的架构就完成了。

**监控和主备切换**

接下来可以完善一下，加上实时监控，如果发现负载均衡的 Nginx 出现问题，就将该机器上的 Keepalived 服务停掉。

nginx\_check.sh:

#!/bin/bash

while :

do

nginxpid = 'ps -C nginx --no-header | wc -l'

if[ $nginxpid -eq 0 ];then

service nginx start

sleep 3

nginxpid = 'ps -C nginx --no-header | wc -l'

echo $nginxpid

if[ $nginxpid -eq 0 ];then

service keepalived stop

fi

fi

sleep 3

done

然后让该脚本一直在后台运行:

nohup /etc/nginx\_check.sh

或者将它添加成服务，让它开机自启动:

<http://blog.sina.com.cn/s/blog_5f54f0be0101b3bs.html>

**测试:**

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

在两台机器的 web 服务器上分别放一个 index.html, 里面内容分别是自己机器的IP.

通过VIP访问：

http://192.168.1.108/index.html 发现显示的是主机的IP.

此时，关掉主机的 nginx, 这时候由于上面的监控脚本。主机的 keepalived 也会关闭。这时候再访问上面地址，发现显示的是备机的IP。可见，切换成功。

**负载均衡**

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

<http://blog.sina.com.cn/s/blog_5f54f0be0100zvf5.html>

通过 ip\_hash 实现会话保持

双主keepalived+nginx

nginx 地址1：192.168.137.131

nginx地址2：192.168.137.133

VIP地址1：192.168.137.10

VIP地址2：192.168.137.11

安装keepalived

keepalived需要openssl-devel组件包

yum install openssl-devel yum安装openssl-devel

tar -zxvf keepalived-1.2.23.tar.gz 解压keepalived

. /configure --prefix=/usr/local/keepalived 编译安装

make

make install

整理文件

cp  /usr/local/keepalived/sbin/keepalived   /usr/sbin/

cp /usr/local/keepalived/etc/sysconfig/keepalived   /etc/sysconfig/

cp /usr/local/keepalived/etc/rc.d/init.d/keepalived   /etc/init.d/

mkdir -p /etc/keepalived

在192.168.137.131机器上创建keepalived.conf文件

[root@localhost tmp]# vi /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

notification\_email {

}

}

vrrp\_script chk\_nginx {

script "/tmp/check\_http.sh" #检测nginx状态目录

interval 2

weight 2

}

vrrp\_instance VI\_1 {

state BACKUP #主写为MASTER

interface eno16777736 #网卡

virtual\_router\_id 11

priority 99 #主是100备99

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

track\_script {

chk\_nginx

}

virtual\_ipaddress {

192.168.137.11 dev eno16777736 label eno16777736:1 #VIP地址2

}

}

vrrp\_instance VI\_2 {

state MASTER

interface eno16777736

virtual\_router\_id 10

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.137.10 dev eno16777736 label eno16777736:2 #VIP地址1

}

}

在192.168.137.133机器上创建keepalived.conf文件

[root@localhost sbin]# vi /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

notification\_email {

}

}

vrrp\_script chk\_nginx {

script "/tmp/check\_http.sh"

interval 2

weight 2

}

vrrp\_instance VI\_1 {

state MASTER

interface eno16777736

virtual\_router\_id 11

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.137.11 dev eno16777736 label eno16777736:1 # VIP地址2

}

}

vrrp\_instance VI\_2 {

state BACKUP

interface eno16777736

virtual\_router\_id 10

priority 99

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

track\_script {

chk\_nginx

}

virtual\_ipaddress {

192.168.137.10 dev eno16777736 label eno16777736:2 #VIP地址1

}

}

配置nginx状态检查 （2台机器都需要执行）

vi /tmp/check\_http.sh

#!/bin/bash

url="http://192.168.137.131/"      # 另一台机器填192.168.137.133

status=$(/usr/bin/curl -s --head "$url" | awk '/HTTP/ {print $2}')

if [ "$status" != "200" ]; then

 /etc/init.d/keepalived stop

fi

给keepalived.conf文件和check\_http.sh文件附权 （2台机器都需要执行）

chmod -R 755 /tmp/check\_http.sh

chmod -R 755 /etc/keepalived/keepalived.conf

验证：

kill 一台机器上nginx 查看IP地址是否自动切换。

负载均衡实现很简单，只需要配置DNS轮询就可以实现负载均衡功能。

#!/bin/bash

#version 0.0.1

#

A=`ps -C nginx --no-header |wc -l`

if [ $A -eq 0 ];then

# /usr/sbin/nginx

service nginx start

sleep 3

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

# killall keepalived

service keepalived stop

fi

fi

#!/bin/bash

while :

do

nginxpid=`ps -C nginx --no-header | wc -l`

if [ $nginxpid -eq 0 ];then

/usr/local/nginx/sbin/nginx

sleep 5

if [ $nginxpid -eq 0 ];then

/etc/init.d/keepalived stop

fi

fi

sleep 5

done

EOF

vim /opt/nginx\_pid.sh

#!/bin/bash

# varsion 0.0.2

A=`ps -C nginx --no-header |wc -l` ## 查看是否有 nginx进程 把值赋给变量A

if [ $A -eq 0 ];then ## 如果没有进程值得为 零

/usr/local/nginx/sbin/nginx

sleep 3

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

killall keepalived ## 则结束 keepalived 进程

fi

fi

#!/bin/bash

while :

do

nginxpid=`ps -C nginx --no-header | wc -l`

if [ $nginxpid -eq 0 ];then

/usr/local/nginx/sbin/nginx

sleep 5

nginxpid=`ps -C nginx --no-header | wc -l`

echo $nginxpid

if [ $nginxpid -eq 0 ];then

/etc/init.d/keepalived stop

fi

fi

sleep 5

done

#!/bin/sh

# check nginx server status

NGINX=/usr/local/nginx/sbin/nginx

PORT=80

nmap localhost -p $PORT | grep "$PORT/tcp open"

#echo $?

if [ $? -ne 0 ];then

$NGINX -s stop

$NGINX

sleep 3

nmap localhost -p $PORT | grep "$PORT/tcp open"

[ $? -ne 0 ] && /etc/init.d/keepalived stop

fi

vim /root/nginx\_pid.sh

#!/bin/bash

while  :

do

nginxpid=`ps -C nginx --no-header | wc -l`

if [ $nginxpid -eq 0 ];then

 /usr/local/nginx/sbin/nginx

 sleep 5

nginxpid=`ps -C nginx --no-header | wc -l`

  if [ $nginxpid -eq 0 ];then

  /etc/init.d/keepalived stop

fi

 fi

 sleep 5

done

然后置于后台运行 sh /root/nginx\_pid.sh &，这种写法是错误的，这样你用root用户logout后，此进程会消失；正确写法为nohup/bin/bash /root/nginx\_pid.sh &,附带下注释:如果你正在运行一个进程，而且你觉得在退出帐户时该进程还不会结束，那么可以使用nohup命令。该命令可以在你退出root帐户之后继续运行相应的进程。nohup就是不挂起的意思( no hang up)，哈哈，差点老马失蹄了。

<http://network.51cto.com/art/201007/209823_3.htm>