**建立高可用的负载均衡**

以下操作，需要在2个节点上执行。

# 一、环境

1、编辑/etc/hosts

192.168.203.20 haproxy

192.168.203.21 haproxy1

192.168.203.22 haproxy2

192.168.203.23 controller1

192.168.203.25 controller2

# 二、修改虚拟IP配置

1、编辑/etc/sysctl.conf

增加一行： net.ipv4.ip\_nonlocal\_bind=1 //启动haproxy的时候，允许忽视VIP的存在

2、使修改生效

$sudo sysctl -p

# 三安装haproxy 、 keepalived

## 3.1安装

在192.168.203.21和192.168.203.22上安装haproxy和keepalived

1、$sudo apt-get update

2、$sudo apt-get install keepalived haproxy

安装报错

### [E: Sub-process /usr/bin/dpkg returned an error code 解决办法](http://jn615.iteye.com/blog/898176)

E: Sub-process /usr/bin/dpkg returned an error code (1)   
此时可以这样解决：   
**cd /var/lib/dpkg   
sudo mv info info.bak   
sudo mkdir info**  
重新安装

## 3.2修改配置文件

1、修改keepalived配置文件（自个创建）

203.21上为MASTER、203.22为BACKUP，标注红色的为MASTER和BACKUP不同的少许却别

root@haproxy1:~# vi /etc/keepalived/keepalived.conf

global\_defs {

router\_id haproxy1

}

vrrp\_script haproxy {

script "killall -0 haproxy"

interval 2

weight 2

nopreempt

}

vrrp\_instance 50 {

virtual\_router\_id 50

advert\_int 1

priority 101

state MASTER

interface p33p1

virtual\_ipaddress {

192.168.203.20/16 dev p33p1

}

track\_script {

haproxy

}

nopreempt

}

2、修改haproxy配置文件

root@haproxy1:~#vi /etc/haproxy/haproxy.cfg

global

log haproxy1 local0

chroot /var/lib/haproxy

user haproxy

group haproxy

daemon

#stats socket /var/lib/haproxy/stats

maxconn 4000

defaults

log global

#mode http

#option httplog

#option dontlognull

option redispatch

#contimeout 5000

#clitimeout 50000

#srvtimeout 50000

retries 3

#stats enable

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout check 10s

listen admin\_stats

bind 192.168.203.20:8080

mode http

stats uri /stats

stats refresh 30s

stats realm HAProxy\ Statistics

stats auth admin:password

stats hide-version

stats admin if TRUE

listen galera

bind 192.168.203.20:3306

balance source

mode tcp

option tcpka

option mysql-check user haproxy

server mysql1 192.168.203.26:3306 check weight 1

server mysql2 192.168.203.27:3306 check weight 1

listen keystone\_admin

bind 192.168.203.20:35357

mode http

balance roundrobin

server controller1 192.168.203.23:35357 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:35357 check inter 2000 rise 2 fall 3

listen keystone\_api

bind 192.168.203.20:5000

mode http

balance roundrobin

server controller1 192.168.203.23:5000 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:5000 check inter 2000 rise 2 fall 3

listen cinder\_api

bind 192.168.203.20:8776

mode http

balance roundrobin

server controller1 192.168.203.23:8776 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:8776 check inter 2000 rise 2 fall 3

listen glance\_api

bind 192.168.203.20:9292

mode http

balance roundrobin

server controller1 192.168.203.23:9292 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:9292 check inter 2000 rise 2 fall 3

listen nova\_api

bind 192.168.203.20:8774

mode http

balance roundrobin

server controller1 192.168.203.23:8774 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:8774 check inter 2000 rise 2 fall 3

listen neutron\_api

bind 192.168.203.20:9696

mode http

balance roundrobin

server controller1 192.168.203.23:9696 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:9696 check inter 2000 rise 2 fall 3

listen horizon\_api

bind 192.168.203.20:80

balance source

mode http

server controller1 192.168.203.23:80 check inter 2000 rise 2 fall 3

#server controller2 192.168.203.25:80 check inter 2000 rise 2 fall 3

listen metadata\_api

bind 192.168.203.20:8775

mode http

balance roundrobin

server controller1 192.168.203.23:8775 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:8775 check inter 2000 rise 2 fall 3

listen vnc\_proxy

bind 192.168.203.20:6080

mode http

balance source

server controller1 192.168.203.23:6080 check inter 2000 rise 2 fall 3

server controller2 192.168.203.25:6080 check inter 2000 rise 2 fall 3

listen rabbitmq\_mirros\_queue

bind 192.168.203.20:5672

mode tcp

balance roundrobin

server controller1 192.168.203.23:5672 check inter 5s rise 2 fall 3

server controller2 192.168.203.25:5672 check inter 5s rise 2 fall 3

3、编辑haproxy配置，设置为可用状态

vi /etc/default/haproxy

ENABLED=1

# 四、启动服务

1、$sudo service keepalived restart

2、$sudo service haproxy restart

# 五、验证

在两个节点上完成上述工作之后，可以做如下验证：

哪个节点先启动上述服务，则该节点将作为主负载均衡，另一个是备用；一旦该主负载均衡down掉，则keepalive将会自动接管故障，由备负载均衡接管工作。

在主负载均衡上执行如下命令：

#ip a |grep eth1

将会显示本地IP和虚拟IP

如果我们把主负载均衡上的服务停掉，并且立即ping 虚拟IP, 看是否能PING通？ 理论上应可可以Ping通，因为此时备负载均衡已经接管工作。

然后登录到备负载均衡，执行以下命令：

#ip a |grep eth1

将会显示本地IP和虚拟IP

# 六、添加日记

haproxy日志

1.在/etc/rsyslog.conf中：

添加haproxy日志路径:

  local0.\* /var/log/haproxy.log

  local3.\* /var/log/haproxy.log（多个）

取消注释:

  #$ModLoad imudp  ==》$ModLoad imudp

  #$UDPServerRun 514 ==》$UDPServerRun 514

重启rsyslog服务

  service rsyslog restart

2.核实services文件

  grep 514 /etc/services

显示：

  syslog 514/udp    为ok!!!

如不存在，手动添加：

  vi /etc/default/rsyslog

  RSYSLOGD\_OPTIONS="-r -c 5"

重启syslog服务，

  service rsyslog restart