LVS集群TUN模式实例(5) - skyflask

目录

1、 实验拓扑图

2、 实验环境

4台CentOS6.2的服务器。

类型	IP
DR	eth0:10.20.73.20
VIP	eth0:0 10.20.73.30
RS	10.20.73.22（web01） 10.20.73.23（web02） 10.20.110.140（web03）【不同网段】

3、安装和配置

3.1 安装

在DS上安装lvs：yum install ipvsadm

3.2 配置

配置lvs启动脚本：

[root@master]# cat /etc/init.d/ipvsnat

```
#!/bin/bash
#lvs script(tunnel mode)

VIP=10.20.73.30
RIP1=10.20.73.22
RIP2=10.20.73.23
RIP3=10.20.110.140

. /etc/rc.d/init.d/functions

case "$1" in
start)
echo "start LVS TUN"
/sbin/ifconfig eth0:0 $VIP broadcast $VIP netmask 255.255.255.255 up
/sbin/route add -host $VIP dev eth0:0
echo "1" > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/ipvsadm -C
/sbin/ipvsadm -A -t $VIP:80 -s rr
/sbin/ipvsadm -a -t $VIP:80 -r $RIP1:80 -i
/sbin/ipvsadm -a -t $VIP:80 -r $RIP2:80 -i
/sbin/ipvsadm -a -t $VIP:80 -r $RIP3:80 -i
/sbin/ipvsadm
;;
stop)
echo "stop LVS TUN"
echo "0" > /proc/sys/net/ipv4/ip_forward
/sbin/ipvsadm -C
/sbin/ifconfig eth0:0 down
;;
*)
echo :Usage:$0{start|stop}
exit 1
esac
```

3. 3 后端真实机安装应用

后端真实机脚本：

```
#!/bin/bash
#lvs script(dr mode)
```

```
VIP=10.20.73.30

. /etc/rc.d/init.d/functions

case "$1" in
start)
echo "start LVS TUNL"
/sbin/ifconfig tunl0 $VIP broadcast $VIP netmask 255.255.255.255 up
/sbin/route add -host $VIP dev tunl0
echo "1" > /proc/sys/net/ipv4/conf/tunl0/arp_ignore
echo "2" > /proc/sys/net/ipv4/conf/tunl0/arp_announce
echo "1" > /proc/sys/net/ipv4/conf/all/arp_ignore
echo "2" > /proc/sys/net/ipv4/conf/all/arp_announce
echo "0" >/proc/sys/net/ipv4/conf/tunl0/rp_filter
sysctl -p
;;
stop)
echo "stop LVS TUN"
/sbin/ifconfig tunl0 down
echo "0" > /proc/sys/net/ipv4/conf/tunl0/arp_ignore
echo "0" > /proc/sys/net/ipv4/conf/tunl0/arp_announce
echo "0" > /proc/sys/net/ipv4/conf/all/arp_ignore
echo "0" > /proc/sys/net/ipv4/conf/all/arp_announce
echo "1" >/proc/sys/net/ipv4/conf/tunl0/rp_filter
;;
*)
echo :Usage:$0{start|stop}
exit 1
esac
```

注意：

1、另外两台台RS脚本一模一样

2、chmod 755 /etc/init.d/ipvstunl

安装和启动服务：

Web01上安装http服务：yum install httpd && service httpd start

Web02上安装http服务：yum install httpd && service httpd start

Web03上安装http服务：yum install httpd && service httpd start

3.4 DR启动脚本并测试

1、在DR服务器上，查看开启tunnel模式前的网卡情况：

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:d4:d3:aa brd ff:ff:ff:ff:ff:ff
    inet 10.20.73.20/23 brd 10.20.73.255 scope global eth0
    inet6 fe80::20c:29ff:fed4:d3aa/64 scope link
       valid_lft forever preferred_lft forever
```

2、开启tunnel服务，service ipvstunl start

```
[root@master ~]# service ipvstunl start
start LVS TUN
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.20.73.30:http rr
  -> 10.20.73.22:http             Tunnel  1      0          0
  -> 10.20.73.23:http             Tunnel  1      0          0
  -> 10.20.110.140:http           Tunnel  1      0          0
[root@master ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> 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: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:d4:d3:aa brd ff:ff:ff:ff:ff:ff
    inet 10.20.73.20/23 brd 10.20.73.255 scope global eth0
    inet 10.20.73.30/32 brd 10.20.73.30 scope global eth0:0
    inet6 fe80::20c:29ff:fed4:d3aa/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 00:0c:29:d4:d3:b4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eth1
    inet6 fe80::20c:29ff:fed4:d3b4/64 scope link
       valid_lft forever preferred_lft forever
[root@master ~]#
```

3、在3台RS上开启ipvstunl服务

```
[root@agent1 ~]# service ipvstunl start
start LVS TUNL
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 4294967295
kernel.shmall = 268435456
[root@agent1 ~]#
```

4、在client上进行测试，client的地址为10.20.122.116（**跨网段**）

测试前，调度器上没有任何连接：

```
[root@master ~]# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.20.73.30:http rr
  -> 10.20.73.22:http             Tunnel  1      0          0
  -> 10.20.73.23:http             Tunnel  1      0          0
  -> 10.20.110.140:http           Tunnel  1      0          0
[root@master ~]# ipvsadm -lnc
IPVS connection entries
pro expire state       source             virtual            destination
[root@master ~]#
```

测试：

for i in `seq 30`;do curl http://10.20.73.30;done

```
[root@master ~]# ipvsadm
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  10.20.73.30:http rr
  -> 10.20.73.22:http             Tunnel  1      0          10
  -> 10.20.73.23:http             Tunnel  1      0          10
  -> 10.20.110.140:http           Tunnel  1      0          10
[root@master ~]# ipvsadm -lnc
IPVS connection entries
pro expire state       source             virtual            destination
TCP 01:54  FIN_WAIT    10.20.122.16:37354 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37377 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37358 10.20.73.30:80     10.20.110.140:80
TCP 01:54  FIN_WAIT    10.20.122.16:37353 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37391 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37371 10.20.73.30:80     10.20.73.23:80
TCP 01:54  FIN_WAIT    10.20.122.16:37352 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37372 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37472 10.20.73.30:80     10.20.73.23:80
TCP 01:56  FIN_WAIT    10.20.122.16:37474 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37378 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37373 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37360 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37363 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37364 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37473 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37426 10.20.73.30:80     10.20.73.23:80
TCP 01:56  FIN_WAIT    10.20.122.16:37476 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37357 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37355 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37359 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37361 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37379 10.20.73.30:80     10.20.110.140:80
TCP 01:55  FIN_WAIT    10.20.122.16:37362 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37433 10.20.73.30:80     10.20.73.22:80
TCP 01:55  FIN_WAIT    10.20.122.16:37356 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37381 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37396 10.20.73.30:80     10.20.110.140:80
TCP 01:56  FIN_WAIT    10.20.122.16:37475 10.20.73.30:80     10.20.73.23:80
TCP 01:55  FIN_WAIT    10.20.122.16:37457 10.20.73.30:80     10.20.110.140:80
[root@master ~]#
```

注意事项：1、rp_filter设置为0，忽略模式，因为这个问题，导致我刚开始没有测试成功。

2、防火墙、selinux关闭；

3、网关不能和nat一样指向内网网关；

作者：@skyflask
转载本文请注明出处：https://www.cnblogs.com/skyflask/p/6759733.html

目录

一、Keepalived概述

本文主要了解开源高可用负载均衡集群利器keepalived，掌握keepalived的安装，运用keepalived配置高可用集群，并能够实现keepalived与负载均衡集群LVS的完美组合。

1、什么是keepalived？

keepalived是一个类似于三、四、五层交换机的软件，也是我们平时说的第三层、第四层、第五层交换。Keepalived的作用是检测web服务器的状态，如果有一台web服务器死机，或工作出现故障，keepalived将检测到，并将有故障的web服务器从系统中剔除，当web服务器工作正常后keepalived自动将web服务器加入到服务器集群中，这些工作全部自动完成，不需要人工干涉，需要人工做的知识修复故障的web服务器。

2、Keepalived的工作原理

三层、四层、五层工作在TCP/IP协议栈的IP层、TCP层、应用层。原理如下：

三层：keepalived使用三层方式工作是，keepalived会定期向服务器集群中的服务器发送一个IMCP的数据包，也就是ping程序，如果发现某台服务器的IP地址没有激活，keepalived便报告这台服务器失效，并将它从集群中删除，这种情况的典型例子是某台服务器被非法关机。三层的方式是以服务器的IP地址是否有效作为服务器工作正常与否的标准。

四层：主要是以TCP端口的状态来决定服务器工作正常与否。如web服务器的端口一般是80，如果keepalived检测到80端口没有启动，则keepalived将这台服务器从集群中剔除。

五层：应用层，比三层和四层要复杂一点，在网上占用的带宽也大一些，keepalived将根据用户的设定检查服务器程序运行是否正常，如果与用户设定的不相符，则keepalived将把服务器从服务器集群中剔除。

3、**keepalived的作用**

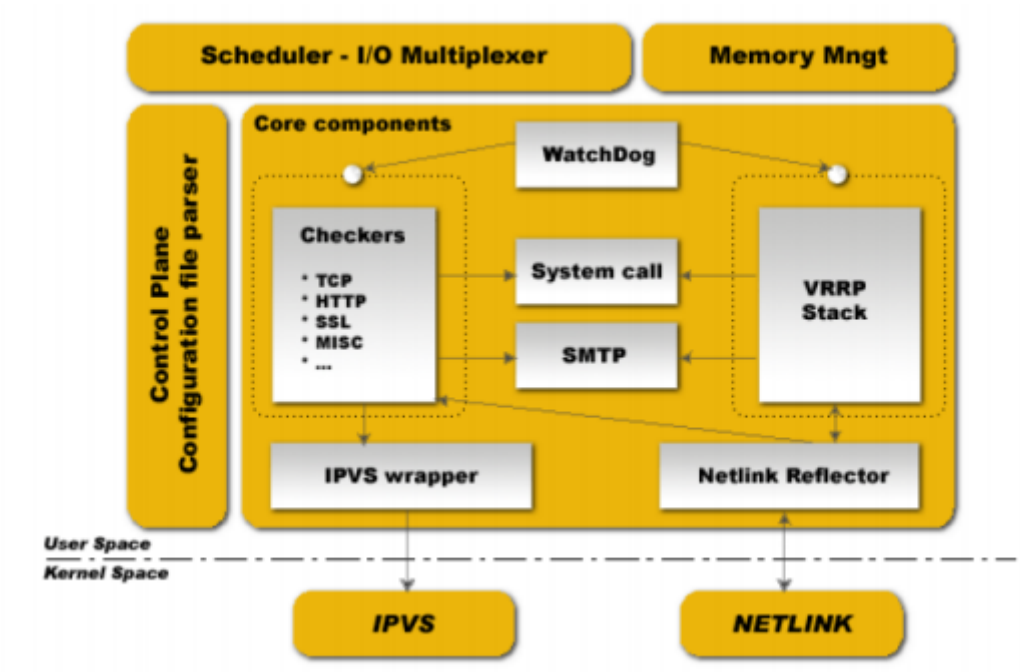
负载均衡-横向扩展

高可用-可持续的服务器质量

实现对失效服务器的隔离-通过健康监测，保证服务的可用性

实现：vrrp协议实现。（冗余网关路由协议）

4、**keepalived体系结构**



1、watchdog 负责监控checkers和vrrp进程的状况。

2、Checkers 负责真实服务器的健康监测，是keepalived最主要的功能，换一句话说，可以没有vrrp statck，但是健康检查healthcheckping一定要有。

3、Vrrp statck 负责负载均衡器之间失败切换failover。如果只用一个负载均衡器，则vrrp不是必须的。

4、Ipvs warpper是用来发送设定的规则封装到内核ipvs代码。

5、Netlink reflector 用来设定vrrp的vip地址等。

Keepalived功能十分强大，但是配置工作十分简单，keepalived各种功能的实现是通过设定配置文件keepalived.conf来完成的。

二、Keepalived的安装

1、**安装keepalived**

二进制编译安装三步骤：./confgiure && make && make install

问题汇总：

1.无gcc等编译工具

Yum install gcc gcc-c++

2.无openssl-devel支持

Yum install openssl-devel

3.无ipvs framework、ipvs syncdeamon support

查看kernels文件：

ls /usr/src/kernels

安装ipvsadm

Yum install kernel-devel ipvsadm

做软连接：

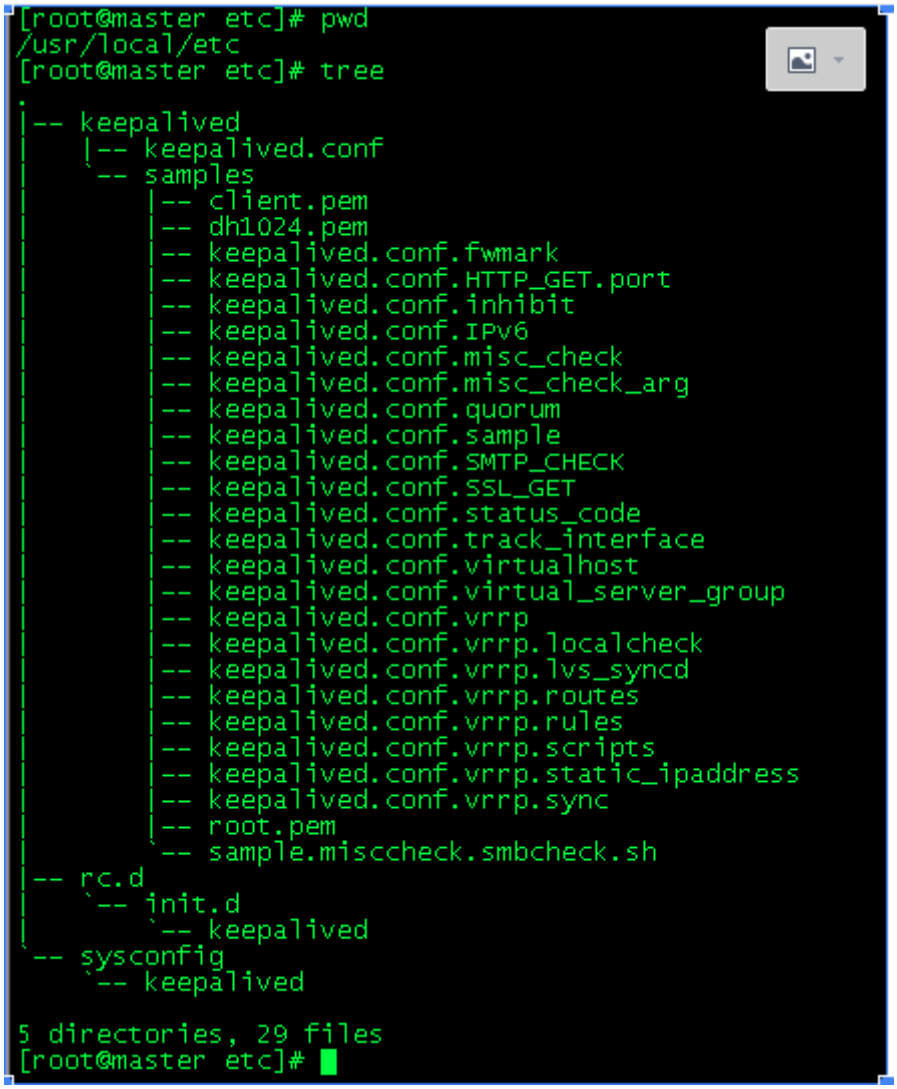
ln -s /usr/src/kernels/2.6.**** /usr/src/linux

4.**无make工具**

Yum install make

编译安装后，默认安装目录为/usr/local/etc/，下面三个目录

Keepalived、rc.d、sysconfig



3.**启动设置**

[root@master etc]# cp /usr/local/etc/rc.d/init.d/keepalived /etc/rc.d/init.d/

[root@master etc]# cp /usr/local/etc/sysconfig/keepalived /etc/sysconfig/

[root@master etc]# mkdir /etc/keepalived

[root@master etc]# cp /usr/local/etc/keepalived/keepalived.conf /etc/keepalived/

[root@master etc]# cp /usr/local/sbin/keepalived /usr/sbin/

[root@master etc]# service keepalived start

env: /etc/init.d/keepalived: 权限不够

[root@master etc]# chmod 755 /etc/init.d/keepalived

[root@master etc]# service keepalived start

正在启动 keepalived：[确定]

[root@master etc]# ps xua|grep keep

root   29312 0.0 0.0   6852   748 ?   Ss   03:57   0:00 keepalived -D

root 29314 0.0 0.2 6908 2024 ? S 03:57 0:00 keepalived -D

root 29315 0.7 0.1 6908 1244 ? S 03:57 0:00 keepalived -D

至此，keepalived安装完成。下一节开始实战。