LVS+keepalived实现负载均衡&高可用

本文我们主要讲解的是LVS通过keepalived来实现负载均衡和高可用，而不是我们第三篇文章介绍的通过手动的方式来进行配置。通过脚本的方式来显示RS节点的健康检查和LVS的故障切换。此文会通过一个实例来讲解，本文目录如下：

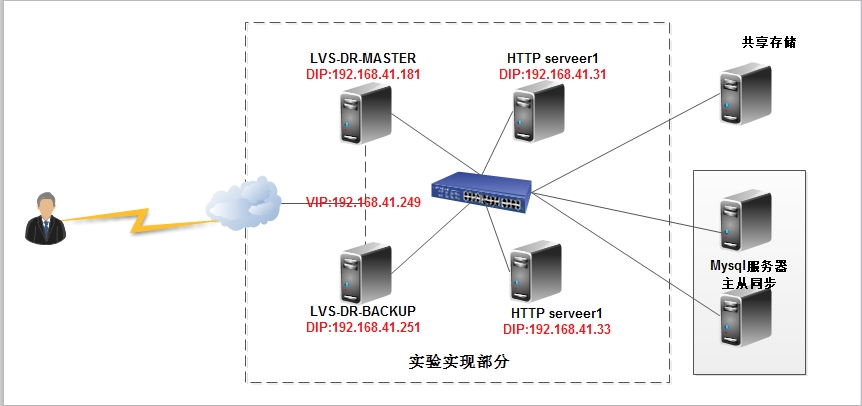
一、实验环境需求&准备

二、配置keepalived实现负载均衡&高可用

**一、实验环境需求&准备**

我们这次实验要完成的一个架构如下图所示，我们通过LVS-DR-MASTER,LVS-DR-BACKUP作为LVS负载均衡调度器，并且两者之间通过keepalived来两者之间的HA。keepalived本身就是为了LVS为开发的，所以说我们通过keepalived来进行LVS的配置就显得十分的方便。而且keepalived是直接操作ip\_vs不用通过ipvsadm，所以更加方便。

1）实验架构图&需求表：

[](http://s3.51cto.com/wyfs02/M02/11/90/wKiom1LVC6LwkWadAAGyY4ob04w803.jpg)

|  |  |  |
| --- | --- | --- |
| **角色** | **IP地址** | **备注** |
| 主LVS调度器（MASTER) | 192.168.41.181 | 使用keepalived配置 |
| 备LVS调度器（BACKUP) | 192.168.41.251 |  |
| HTTP服务器（RS1) | 192.168.41.31 | apache服务器（一般生产环境需要外网IP地址，这里用内网IP地址替代） |
| HTTP服务器（RS2) | 192.168.41.33 |  |
| 虚拟IP地址（VIP) | 192.168.41.249 | 虚拟IP地址 |

2）部署http服务器，验证能正常访问

这里就不多费篇幅介绍了，就是要保证http能正常访问。

**二、配置keepalived实现负载均衡&高可用**

**1）安装keepalived软件**

wget <http://www.keepalived.org/software/keepalived-1.2.8.tar.gz>

tar -zxf keepalived-1.2.8.tar.gz

cd keepalived-1.2.8

./configure --prefix=/usr/local/keepalived

make

make install

#配置keepalived的自启动&拷贝keepalived的执行程序

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

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/

#开启内核的转发功能

vi /etc/sysctl  net.ipv4.ip\_forword = 1

**2）配置LVS-DR-MASK的keepalived.conf配置文件**

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {    notification\_email {      752119102@qq.com            #设置报警邮箱，一般不再这做，而是用其他方式报警。    }    notification\_email\_from keepalived@localhost   #设定发送邮件地址    smtp\_server 127.0.0.1                          #设定发送邮件服务器    smtp\_connect\_timeout 30                        #设定SMTP连接超时时间    router\_id LVS\_DEVEL          #查阅说明文档得知route\_id配置是为了标识当前节点，我将其设置为NodeA。当然两个节点的此项设置可相同，也可不相同。  }  vrrp\_instance VI\_1 {                        #定义虚拟路由实例，不同实例ID不同。     state MASTER                           #定义服务器在keepalived中的角色主服务器     interface eth0                          #定义进行检测的端口eth0     virtual\_router\_id 51               #定义虚拟路由ID，同一个实例的主从一样。     priority 100                      #定义在虚拟路由器组的权限，越大越高     advert\_int 1                       #定义检测时间间隔     authentication {                   #定义认证方式密码，主从必须一样         auth\_type PASS         auth\_pass 1111     }     virtual\_ipaddress {                #指定虚拟IP地址         192.168.41.249     }  }  virtual\_server 192.168.41.249 80 {  #定义虚拟服务，需指定IP地址和端口，空格隔开。     delay\_loop 6                      #定义RS运行情况监测时间间隔     lb\_algo rr                        #定义负载调度算法     lb\_kind DR                      #定义LVS的工作模式     nat\_mask 255.255.255.0            #定义虚拟服务的mask     persistence\_timeout 50            #定义会话保持时间，S为单位     protocol TCP                      #指定转发协议     real\_server 192.168.41.31 80 {    #定义真实服务器IP地址和端口         weight 1                      #定义RS的权重         TCP\_CHECK{                    #RS server健康检查部分         connect\_timeout 10            #定义超出10s连接超时         nb\_get\_retry 3                #定义重试次数         delay\_before\_retry 3          #定义重试时间间隔         connect\_port 80               #定义健康检查端口     }     real\_server 192.168.41.33 80 {         weight 1         TCP\_CHECK{         connect\_timeout 10         nb\_get\_retry 3         delay\_before\_retry 3         connect\_port 80     }  } |

**3）配置LVS-DR-BACKUP的keepalived.conf配置文件**

|  |
| --- |
| ! Configuration File for keepalived  global\_defs {    notification\_email {      752119102@qq.com            #设置报警邮箱，一般不再这做，而是用其他方式报警。    }    notification\_email\_from keepalived@localhost   #设定发送邮件地址    smtp\_server 127.0.0.1                          #设定发送邮件服务器    smtp\_connect\_timeout 30                        #设定SMTP连接超时时间    router\_id LVS\_DEVEL            #负载均衡器标示，在局域网内是唯一的  }  vrrp\_instance VI\_1 {                        #定义虚拟路由实例，不同实例ID不同。     state BACKUP                           #定义服务器在keepalived中的角色     interface eth0                          #定义进行检测的端口eth0     virtual\_router\_id 51               #定义虚拟路由ID，同一个实例的主从一样。     priority 50                       #定义在虚拟路由器组的权限，越大越高     advert\_int 1                       #定义检测时间间隔     authentication {                   #定义认证方式密码，主从必须一样         auth\_type PASS         auth\_pass 1111     }     virtual\_ipaddress {                #指定虚拟IP地址         192.168.41.249     }  }  virtual\_server 192.168.41.249 80 {  #定义虚拟服务，需指定IP地址和端口，空格隔开。     delay\_loop 6                      #定义RS运行情况监测时间间隔     lb\_algo rr                        #定义负载调度算法     lb\_kind DR                      #定义LVS的工作模式     nat\_mask 255.255.255.0            #定义虚拟服务的mask     persistence\_timeout 50            #定义会话保持时间，S为单位     protocol TCP                      #指定转发协议     real\_server 192.168.41.31 80 {    #定义真实服务器IP地址和端口         weight 1                      #定义RS的权重         TCP\_CHECK{                    #RS server健康检查部分         connect\_timeout 10            #定义超出10s连接超时         nb\_get\_retry 3                #定义重试次数         delay\_before\_retry 3          #定义重试时间间隔         connect\_port 80               #定义健康检查端口     }     real\_server 192.168.41.33 80 {         weight 1         TCP\_CHECK{         connect\_timeout 10         nb\_get\_retry 3         delay\_before\_retry 3         connect\_port 80     } |

说明：这里主LVS-DR-MASTER和LVS-DR-BACKUP之间的配置的差别就只有红色部分：HA的角色（MASTER,BACKUP)和优先级不同，还有router\_id。

**4）客户端配置LVS参数**

客户端需要做的工作就是绑定我们的VIP在lo口，并且进行ARP抑制，之前的文章已经提过此方法咯。现在我们就换成将配置写成脚本来执行。

脚本内容：

|  |
| --- |
| [root@RS2 ~]# cat lvs-client.sh  #!/bin/bask  #       752119102@qq.com  #  . /etc/rc.d/init.d/functions  VIP=(  192.168.41.249  )  function start(){    for ((i=0;i<`echo ${#VIP[\*]}`;i++))        do         echo ${i}  ${VIP[$i]}         ifconfig lo:${i} ${VIP[$i]} netmask 255.255.255.255 up         route add -host ${VIP[$i]} dev lo         done  echo "1">/proc/sys/net/ipv4/conf/lo/arp\_ignore  echo "2">/proc/sys/net/ipv4/conf/lo/arp\_announce  echo "1">/proc/sys/net/ipv4/conf/all/arp\_announce  echo "2">/proc/sys/net/ipv4/conf/all/arp\_announce  }  function  stop(){    for ((i=0;i<${#VIP[\*]};i++))        do         echo ${i}  ${VIP[$i]}         ifconfig lo:${i} ${VIP[$i]} netmask 255.255.255.255 up         route del -host ${VIP[$i]} dev lo:${i}         done  }  case "$1" in     start)         start         exit         ;;     stop)         stop         exit         ;;     \*)         echo "You must use $0:stop|start"         ;;  esac |

**5）测试实验结果**

如果测试部成功可以按照三角的排查原理来进行排查，显示client到RS端是否能通讯，LB到RS能否通讯，client到LB是否能通讯，client到VIP是否能够通讯。并且查看LVS的运行状态。一定要确保keepalived.conf这个配置文件是正确的。