第三十三章：keepalived应用部署

**一、keepalived概述；**

**二、设计原理；**

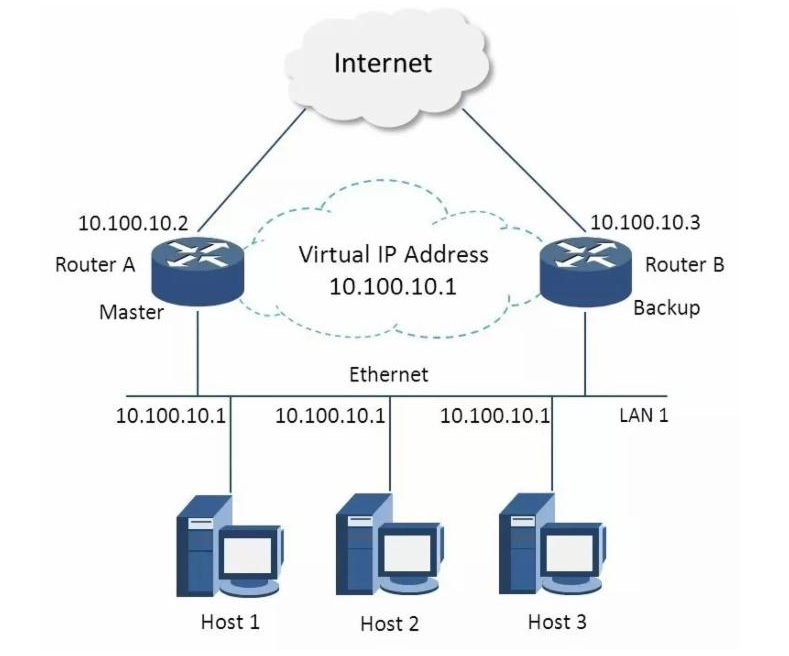
**三、案例：Keepalived实现双机热备；**

**一、keepalived概述：**

**概述：**keepalived起初为Lvs设计的一款强大的辅助工具，保证Lvs负载调度器的故障切换以及web节点的健康状态检查，后续被应用到很多需要容错的场景下，keepalived自身基于VRRP协议--虚拟路由冗余协议，思科公有协议；

**二、设计原理：**

**应用场景：**

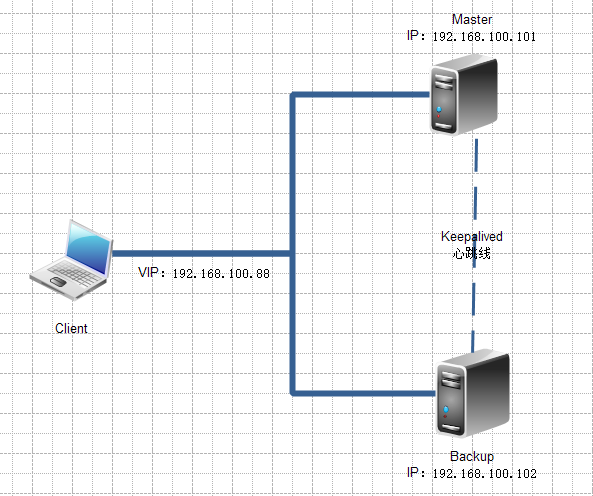
****

**设计模块：**  
1.core模块：为keepalived的核心组件，负责主进程的启动、维护以及全局配置文件的加载和解析；  
2.check模块：负责real server 节点池内的节点的健康检测；  
3.VRRP模块：在master与backup之间执行心跳检测；

**热备实现过程：**将多个主机以软件的方式组成一个热备组，通过共有的虚拟ip（VIP）地址对外提供服务，同一时刻，热备组中只有一台主机在工作，别的主机冗余状态，当当前在线的主机失效时，其他冗余的主机会自动接替虚拟ip地址，继续提供服务，以保证架构的稳定性；

**三、案例：Keepalived实现双机热备；**

**案例拓扑：**



**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 7.4 1708 64bit | 192.168.100.101 | node1.linuxfan.cn | keepalived-1.2.13.tar.gz |
| Centos 7.4 1708 64bit | 192.168.100.102 | node2.linuxfan.cn | keepalived-1.2.13.tar.gz |

**案例步骤：**

* 安装node1节点上的httpd的服务；
* 安装node2节点上的httpd的服务；
* 在两台node节点上安装keepalived软件程序（两台安装步骤一致，在此只列出一台）；
* 配置node1上master主节点；
* 配置node2上backup从节点；
* 客户端访问测试双机热备的效果；
* 扩展内容：上述步骤实现了keepalived在real server节点池中进行了四层健康检测（传输层，基于协议或者端口），下边展示keepalived如何对节点池中进行七层健康检测（应用层）；
* **安装node1节点上的httpd的服务；**

[root@node1 ~]# yum -y install httpd

[root@node1 ~]#cat <<END >>/var/www/html/index.html

192.168.100.101

END

[root@node1 ~]# systemctl start httpd

[root@node1 ~]# systemctl enable httpd

[root@node1 ~]# netstat -utpln |grep 80

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 13891/httpd

* **安装node2节点上的httpd的服务；**

[root@node2 ~]# yum -y install httpd

[root@node2 ~]#cat <<END >>/var/www/html/index.html

192.168.100.102

END

[root@node2 ~]# systemctl start httpd

[root@node2 ~]# systemctl enable httpd

[root@node2 ~]# netstat -utpln |grep 80

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 13891/httpd

* **在两台node节点上安装keepalived软件程序（两台安装步骤一致，在此只列出一台）；**

[root@node1 ~]# yum -y install kernel-devel openssl-devel popt-devel ##安装内核开发包，popt支持库等工具；

[root@node1 ~]# tar -zxvf keepalived-1.2.13.tar.gz -C /usr/src/

[root@node1 ~]# cd /usr/src/keepalived-1.2.13/

[root@node1 keepalived-1.2.13]# ./configure --prefix=/usr/local/keepalived

[root@node1 keepalived-1.2.13]# make &&make install

[root@node1 keepalived-1.2.13]# cd

[root@node1 ~]# mkdir -p /etc/keepalived ##程序的主配置目录

[root@node1 ~]# cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/ ##复制主配置文件

[root@node1 ~]# cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/ ##复制启动时需要加载的配置文件

[root@node1 ~]# cp /usr/local/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/ ##复制服务的控制脚本

[root@node1 ~]# cp /usr/local/keepalived/sbin/keepalived /usr/sbin/ ##复制keepalived的命令

[root@node1 ~]# chmod 755 /etc/init.d/keepalived ##为控制脚本指定权限

* **配置node1上master主节点；**

[root@node1 ~]# vi /etc/keepalived/keepalived.conf

global\_defs {

router\_id HA\_TEST\_R1 ##本服务器的名称，若环境中有多个keepalived时，此名称不能一致

}

vrrp\_instance VI\_1 { ##定义VRRP热备实例，每一个keep组都不同

state MASTER ##MASTER表示主服务器

interface eth0 ##承载VIP地址的物理接口

virtual\_router\_id 1 ##虚拟路由器的ID号，每一个keep组都不同

priority 100 ##优先级，数值越大优先级越高

advert\_int 1 ##通告检查间隔秒数（心跳频率）

authentication { ##认证信息

auth\_type PASS ##认证类型

auth\_pass 123456 ##密码字串

}

virtual\_ipaddress {

192.168.100.95 ##指定漂移地址（VIP）

}

}

virtual\_server 192.168.100.95 80 { #vip配置

delay\_loop 2 #每隔2秒检查一次real\_server状态

lb\_algo wrr ##指定lvs的调度算法

lb\_kind DR ##lvs集群模式（如若不结合LVS请删掉此三个配置）

persistence\_timeout 60 ##会话保持时间

protocol TCP ##选择协议

real\_server 192.168.100.101 80 { ##本机地址

weight :3 ##服务器的权重

notify\_down /etc/keepalived/check.sh ##指定节点失效后，采用的脚本，notify\_up表示节点正常后，采用的脚本

##健康检查方式一共有HTTP\_GET|SSL\_GET|TCP\_CHECK|SMTP\_CHECK|MISC\_CHECK这些

TCP\_CHECK {

connect\_timeout 10 ##连接超时时间

nb\_get\_retry 3 ##重连次数

delay\_before\_retry 3 ##重连间隔时间

connect\_port 80 ##健康检查端口

}

}

}

:wq

[root@node1 ~]# vi /etc/keepalived/check.sh

#!/bin/bash

/etc/init.d/keepalived stop

echo -e "$(ip a |grep eth0 |grep inet |awk '{print $2}'|awk -F'/' '{print $1}') (httpd) is down on $(date +%F-%T)" >>/root/check\_httpd.log

:wq

[root@node1 ~]# chmod 777 /etc/keepalived/check.sh

[root@node1 ~]# /etc/init.d/keepalived start

Starting keepalived (via systemctl): [ 确定 ]

[root@node1 ~]# ip a |grep 192.168.100.95

inet 192.168.100.95/32 scope global eth0

* **配置node2上backup从节点；**

[root@node2 ~]# vi /etc/keepalived/keepalived.conf

global\_defs {

router\_id HA\_TEST\_R2 ##本服务器的名称

}

vrrp\_instance VI\_1 {

state BACKUP ##BACKUP表示从服务器

interface eth0

virtual\_router\_id 1

priority 99 ##优先级，低于主服务器

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 123456

}

virtual\_ipaddress {

192.168.100.95

}

}

virtual\_server 192.168.100.95 80 { ##vip配置

delay\_loop 2 ##每隔2秒检查一次real\_server状态

lb\_algo wrr

lb\_kind DR ##如若不结合LVS，需要去掉此三个配置项

persistence\_timeout 60 ##会话保持时间

protocol TCP

real\_server 192.168.100.102 80 { ##本机地址

weight :3

notify\_down /etc/keepalived/check.sh

TCP\_CHECK {

connect\_timeout 10 ##连接超时时间

nb\_get\_retry 3 ##重连次数

delay\_before\_retry 3 ##重连间隔时间

connect\_port 80 ##健康检查端口

}

}

}

[root@node2 ~]# vi /etc/keepalived/check.sh

#!/bin/bash

service keepalived stop

echo -e "$(ip a |grep eth0 |grep inet |awk '{print $2}'|awk -F'/' '{print $1}') (httpd) is down on $(date +%F-%T)" >>/root/check\_httpd.log

:wq

[root@node2 ~]# chmod 777 /etc/keepalived/check.sh

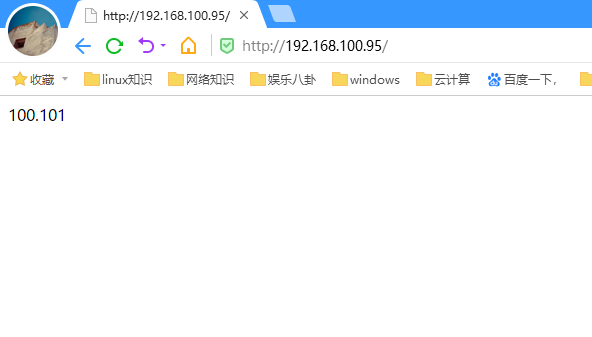
[root@node2 ~]# /etc/init.d/keepalived start

Starting keepalived (via systemctl): [ 确定 ]

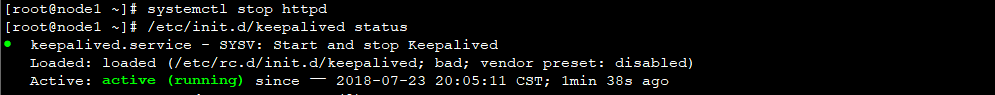
[root@node2~]# ip a |grep 192.168.100.95

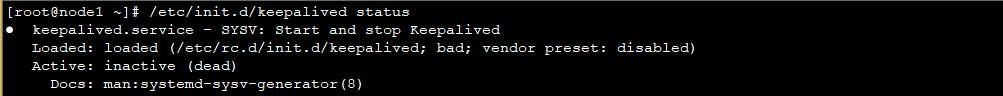
* **客户端访问测试双机热备的效果；**

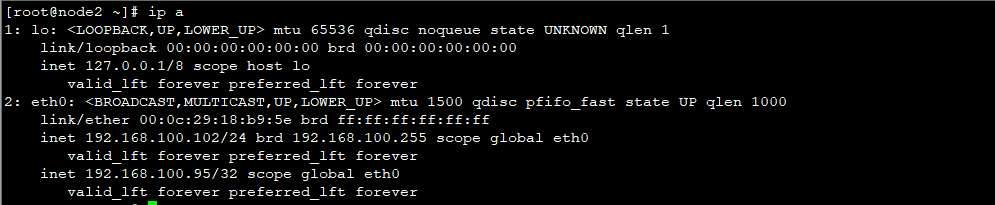
当node1与node2均正常状态：

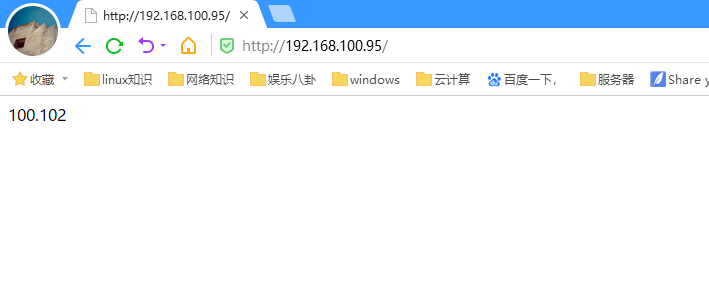


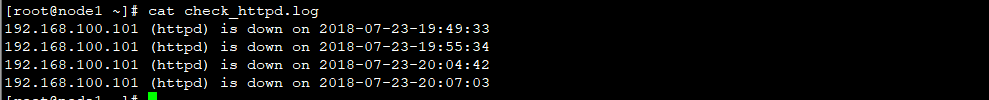
当node1的httpd服务关闭：



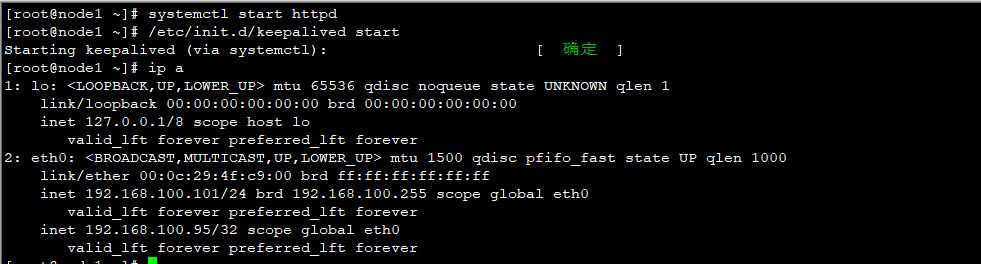








将node1节点的http和keepalived服务重新启动：





* **扩展内容：上述步骤实现了keepalived在real server节点池中进行了四层健康检测（传输层，基于协议或者端口），下边展示keepalived如何对节点池中进行七层健康检测（应用层）；**

[root@node1 ~]# vi /etc/keepalived/keepalived.conf

global\_defs {

router\_id HA\_TEST\_R1 ##本服务器的名称，若环境中有多个keepalived时，此名称不能一致

}

vrrp\_script check\_nginx {

script /etc/check\_nginx.sh ##检测脚本

interval 2 ##执行间隔时间

}

vrrp\_instance VI\_1 { ##定义VRRP热备实例，每一个keep组都不同

state MASTER ##MASTER表示主服务器

interface eth0 ##承载VIP地址的物理接口

virtual\_router\_id 1 ##虚拟路由器的ID号，每一个keep组都不同

priority 100 ##优先级，数值越大优先级越高

advert\_int 1 ##通告检查间隔秒数（心跳频率）

authentication { ##认证信息

auth\_type PASS ##认证类型

auth\_pass 123456 ##密码字串

}

virtual\_ipaddress {

192.168.100.95 ##指定漂移地址（VIP）

}

track\_script { #在实例中引用脚本

check\_nginx

}

}

virtual\_server 192.168.100.95 80 { #vip配置

protocol TCP ##选择协议

real\_server 192.168.100.101 80 { ##本机地址

weight :3 ##服务器的权重

}

}

[root@node1 ~]# /etc/keepalived/check\_nginx.sh

#!/bin/bash

Count1=`netstat -antp |grep -v grep |grep nginx |wc -l`

if [ $Count1 -eq 0 ]; then

/usr/local/nginx/sbin/nginx

sleep 2

Count2=`netstat -antp |grep -v grep |grep nginx |wc -l`

if [ $Count2 -eq 0 ]; then

/etc//init.d/keepalived start

else

exit 0

fi