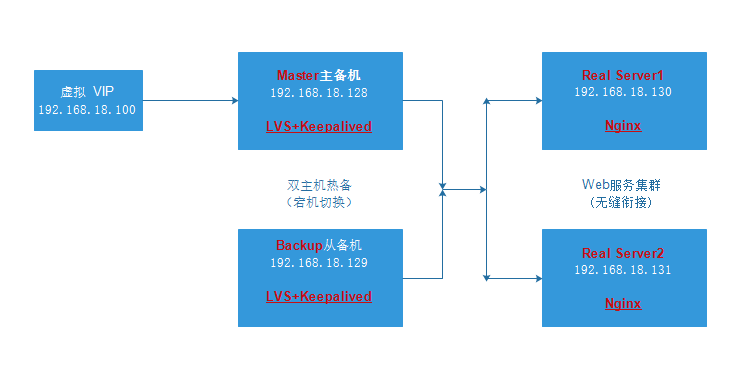
LVS+Keepalived多机热备高可用负载均衡

董仕应

## 一、系统架构



## 二、为什么要做负载均衡

1、系统高可用性

2、系统可扩展性

3、负载均衡能力

LVS+keepalived能够很好的实现以上的要求，其中LVS提供负载均衡，keepalived提供健康检查，故障转移，提高系统的可用性！采用这样的架构可以很容易对现有系统进行扩展，只需在后端添加或者减少realserver节点，接着更新LVS的配置文件，就能实现无缝拓展集群节点！

## 三、LVS+Keepalived 概述

### 3.1、LVS

LVS是Linux Virtual Server的缩写，即Linux虚拟服务器。

LVS是一个开源的软件，可以实现Linux系统下的负载均衡。

LVS有三种IP负载均衡技术（VS/NAT、VS/TUN和VS/DR）；

VS/NAT: 通过网络地址转换将内部地址转化为网络上的可用的外部地址。（最简单）

VS/TUN: 通过IP隧道实现虚拟服务器，将请求报文封装动态地转发给后端服务器，响应报文从后端服务器直接返回给客户。因请求报文较短而响应报文包含大量的数据，可以大大降低负载均衡器的压力。

VS/DR: 利用大多数网络服务的非对称特点，负载均衡器只负责将请求调度到不同的服务器，服务器将应答直接返回给客户端，而不需要经过负载均衡器，其极大的提高整个集群系统的吞吐量。（大型网站应用最广泛）

LVS有八种调度算法（rr,wrr,lc,wlc,lblc,lblcr,dh,sh）。

rr: 轮叫调度 wrr: 加权轮叫调度

lc: 最小连接调度 wlc: 加权最小连接调度

lblc: 基于局部性的最少链接 lblcr: 带复制的基于局部性最少链接

dh: 目标地址散列调度 sh: 源地址散列调度

### 3.2、keepalived

KeepAlived是一个路由软件，通过简单的配置，实现高可用负载均衡，其中负载均衡依赖于Linux虚拟服务器（IPVS）的内核模块，而其高可用性依赖于VRRP协议来实现。KeepAlived运行在lvs之上，不仅会检测负载均衡服务器池中每台机器的健康状况并通知IPVS将非健康的机器从池中移除掉；同时它还能对负载均衡调度器**本身**实现健康状态检查，当主负载均衡调度器出现问题时，备用负载均衡调度器顶替进行工作，实现真实机的故障隔离及负载均衡器间的失败切换，提高系统的可用性，也即常说的多机热备高可用。

## 四、安装与配置

**注意主备负载机器都需要安装配置 LVS + keepalived !**

### 4.1、安装依赖库环境

yum install kernel-devel -y  
yum -y install popt popt-devel libnl libnl-devel popt-static

yum install -y zlib zlib-devel gcc gcc-c++ openssl openssl-devel openssh

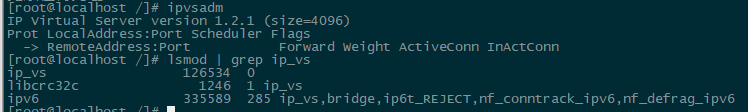
### 4.2、安装LVS

**yum -y install ipvsadm**

### 4.3、把ipvsadm模块加载进系统

**ipvsadm**

**lsmod | grep ip\_vs**



### 4.4、安装keepalived

**yum install keepalived**

**#建立符号链接**

**ln -s /usr/local/sbin/keepalived /sbin/**

**#增加为系统服务**

**chkconfig --add keepalived**

**#设定keepalived在等级3和5为开机运行服务**

**chkconfig --level 35 keepalived on**

#### 打开IP4转发功能:

（LVS现有三种负载均衡规则都需要打开此功能！！！）

vim /etc/sysctl.conf

修改下面的值为1:

net.ipv4.ip\_forward = **1**

让修改生效：

sysctl -p

#### 配置keepalived.conf

! Configuration File for keepalived

# 全局配置

global\_defs {

#不发报警邮件

#notification\_email {

# 设置报警邮件地址，可以设置多个，每行一个。

# 需开启本机的sendmail服务

#acassen@firewall.loc

#failover@firewall.loc

#sysadmin@firewall.loc

#}

# 设置邮件的发送地址

#notification\_email\_from Alexandre.Cassen@firewall.loc

# 设置smtp server地址

#smtp\_server 192.168.200.1

# 设置连接smtp server的超时时间

#smtp\_connect\_timeout 30

**# 表示运行keepalived服务器的一个标识。**发邮件时显示在邮件主题的信息

**router\_id LVS\_MASTER**

vrrp\_skip\_check\_adv\_addr

vrrp\_strict

vrrp\_garp\_interval 0

vrrp\_gna\_interval 0

}

# vrrp实例配置

vrrp\_instance VI\_1 {

# 指定keepalived的角色，MASTER表示此主机是主服务器，BACKUP表示此主机是备用服务器

**# 指定HA监测网络的接口**

**state MASTER**

# 指定HA监测网络的接口,eth0为网卡名称，在**/etc/sysconfig/network-scripts/ifcfg-e（敲下TAB）**

**interface eth0**

# **虚拟路由标识**，这个标识是一个数字，同一个vrrp实例使用唯一的标识。即同一vrrp\_instance下，**MASTER和BACKUP必须是一致的**

**virtual\_router\_id 88**

# 定义优先级，数字越大，优先级越高，在同一个vrrp\_instance下，**MASTER的优先级必须大于BACKUP的优先级**

**priority 100**

# 设定MASTER与BACKUP负载均衡器之间同步检查的时间间隔，单位是秒

advert\_int 1

# 设置验证类型和密码

authentication {

# 设置验证类型，主要有PASS和AH两种

auth\_type PASS

# 设置验证密码，在同一个vrrp\_instance下，**MASTER与BACKUP必须使用相同的密码才能正常通信**

**auth\_pass 1234**

}

**# 设置虚拟服务器IP地址**，可以设置多个虚拟IP地址，每行一个

virtual\_ipaddress {

**192.168.18.100**

}

}

**# 配置虚拟服务器**，需要指定虚拟IP地址和服务端口，IP与端口之间用空格隔开

virtual\_server **192.168.18.100 80** {

# 设置运行情况检查时间，单位是秒

delay\_loop 6

# 设置负载调度算法，这里设置为rr，即轮询算法

**lb\_algo rr**

# 设置LVS实现负载均衡的机制，有NAT、TUN、DR三个模式可选

**lb\_kind DR**

# 定义虚拟服务的mask

nat\_mask 255.255.255.0

#会话保持时间，单位是秒。这个选项对动态网页是非常有用的，为集群系统中的session共享提供了一个很好的解决方案。

#有了这个会话保持功能，用户的请求会被一直分发到某个服务节点，直到超过这个会话的保持时间。

#需要注意的是，这个会话保持时间是最大无响应超时时间，也就是说，用户在操作动态页面时，如果50秒内没有执行任何操作

#那么接下来的操作会被分发到另外的节点，但是如果用户一直在操作动态页面，则不受50秒的时间限制

persistence\_timeout 50

#指定转发协议类型，有TCP和UDP两种

protocol TCP

**# 配置Web服务节点1**，需要指定real server的真实IP地址和端口，IP与端口之间用空格隔开

**real\_server 192.168.18.130 80** {

# 配置服务节点的权值，权值大小用数字表示，数字越大，权值越高，设置权值大小可以为不同性能的服务器

# 分配不同的负载，可以为性能高的服务器设置较高的权值，而为性能较低的服务器设置相对较低的权值，这样才能合理地利用和分配系统资源

weight 1

**TCP\_CHECK** {

# realserver的状态检测设置部分，单位是秒

# 表示3秒无响应超时

connect\_timeout 3

# 表示重试次数

nb\_get\_retry 3

# 表示重试间隔

delay\_before\_retry 3

# 连接端口

connect\_port 80

}

}

**# 配置Web服务节点2**，需要指定real server的真实IP地址和端口，IP与端口之间用空格隔开

**real\_server 192.168.18.131 80** {

# 配置服务节点的权值，权值大小用数字表示，数字越大，权值越高，设置权值大小可以为不同性能的服务器

# 分配不同的负载，可以为性能高的服务器设置较高的权值，而为性能较低的服务器设置相对较低的权值，这样才能合理地利用和分配系统资源

weight 1

TCP\_CHECK {

# realserver的状态检测设置部分，单位是秒

# 表示3秒无响应超时

connect\_timeout 3

# 表示重试次数

nb\_get\_retry 3

# 表示重试间隔

delay\_before\_retry 3

# 连接端口

connect\_port 80

}

}

}

好了，LVS+keepalived已经安装配置完成，但在启动之前需要配置好相应的Real Server节点机器， **看步骤五！！！**

此外，如果各机器防火墙端口没有合理开放，可以关闭防火墙。

#### 启动keepalived：

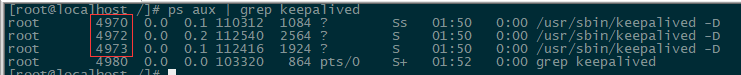
#设定keepalived在各等级为on，“各等级”包括2、3、4、5等级

chkconfig keepalived on

service keepalived start

# 查看进程，必须包含三个进程才启动成功

ps aux | grep keepalived



查看keepalived日志：

**tail -f /var/log/messages**

## 五、Real Server节点配置

### 5.1 安装配置好Nginx（略）

### 5.2 配置虚拟IP启动脚本(realserver.sh):

#!/bin/bash

#description: Config realserver

**VIP=192.168.18.100**

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

case "$1" in

start)

/sbin/ifconfig lo:0 $VIP netmask 255.255.255.255 broadcast $VIP

/sbin/route add -host $VIP dev lo:0

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\_ignore

echo "2" >/proc/sys/net/ipv4/conf/all/arp\_announce

sysctl -p >/dev/null 2>&1

echo "RealServer Start OK"

;;

stop)

/sbin/ifconfig lo:0 down

/sbin/route del $VIP >/dev/null 2>&1

echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_ignore

echo "0" >/proc/sys/net/ipv4/conf/lo/arp\_announce

echo "0" >/proc/sys/net/ipv4/conf/all/arp\_ignore

echo "0" >/proc/sys/net/ipv4/conf/all/arp\_announce

echo "RealServer Stoped"

;;

\*)

echo "Usage: $0 {start|stop}"

exit 1

esac

exit 0

将脚本放于 **/etc/init.d/realserver.sh** 路径下，仅修改VIP即可，**注意如果是在windows系统下编辑可能会因为字符错乱问题导致脚本无法执行！**

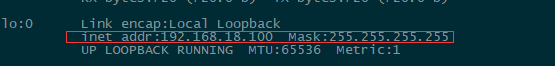
### 5.3 启动realserver.sh脚本

**sh /etc/rc.d/init.d/realserver.sh start**

**此时可能会报没有这个脚本的执行权限，需要加权限:**

**chmod 777 /etc/rc.d/init.d/realserver.sh**

**ifconfig查看脚本是否启动成功：**



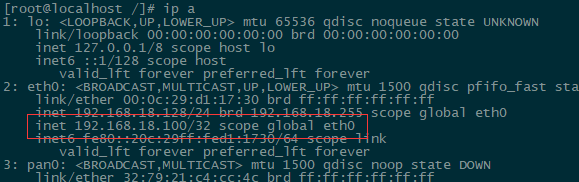
## 六、测试

分别启动主备机的keepalived之后，进行相关测试。

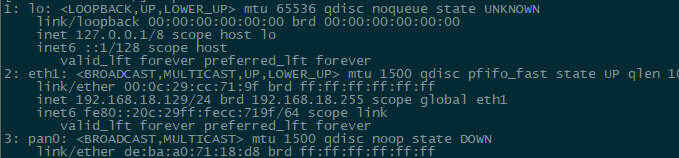
#### 6.1 测试虚拟VIP是否设置成功

**ip a**

**MASTER机上：**



**BACKUP机上：**

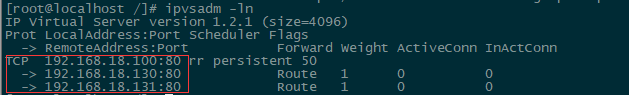


结果显示目前虚拟VIP是由**MASTER**主备机提供，**BACKUP**从备机此时没有提供，此时如果停止**MASTER**主备机的keepalived，那么结果将反过来，表明**BACKUP**从备机接替**MASTER**主备机临时成为主备机提供虚拟VIP。

#### 6.2 分别查询主从备机是否连上Real Server

查询集群中服务器ip信息：

**ipvsadm -ln**



#### 6.3 主从备机分别查看请求转发情况

浏览器访问虚拟VIP：



**ipvsadm -lcn | grep 192.168.18.100**



以上为主备机出现的请求转发信息，而此时从备机是没有转发信息的。

可以停止主备机的keepalived，接着再试一次。



此时将轮到从备机上有转发信息，而主备机上则没有。



参考博客：http://blog.csdn.net/xjtuse2014/article/details/54292600