LVS简介及LVS-DR模式配置搭建

## 一、 LVS集群简介

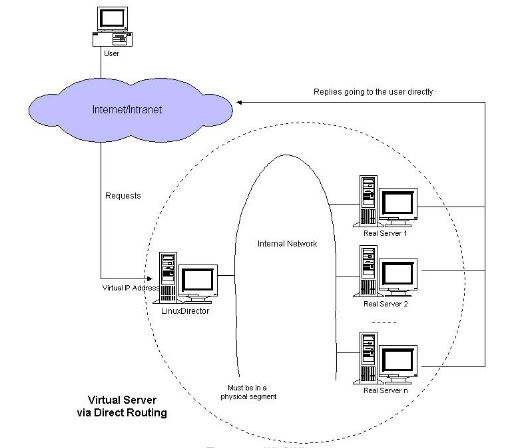
一 LVS集群简介

   LVS:linux虚拟服务器，是一个虚拟服务器集群系统

   LVS开源站点 官网：<http://www.linuxvirtualserver.org/>

中文站点：<http://zh.linuxvirtualserver.org/>

二 LVS-DR模式介绍

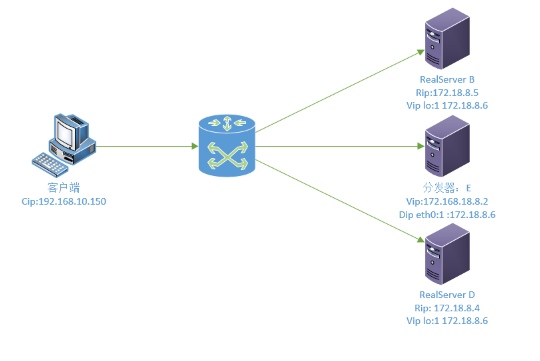


LVS-DR模式架构图

VS/DR模式交互过程:客户机-->linuxDirector ->Real Server-->客户机。

## 二、LVS安装步骤

具体部署图



#### 1****配置分发器****

cd /etc/sysconfig/network-scripts/

cp ifcfg-eth0 ifcfg-eth0:1

vi ifcfg-eth0:1

编辑由拷贝而来的ifcfg-eth0：1，需要注意的是必须把mac地址与eth0的保持一致

TYPE="Ethernet"

BOOTPROTO="static"

DEFROUTE="yes"

PEERDNS="yes"

PEERROUTES="yes"

IPV4\_FAILURE\_FATAL="no"

NAME="eth0:1"

DEVICE="eth0:1"

ONBOOT="yes"

IPADDR="172.18.8.6"

NETMASK="225.225.225.0"

HWADDR="00:0c:29:af:ff:3a"

#### 2 重启网络

service network restart

#### 3安装ipvsadm

##### 3.1yum 安装

yum install ipvsadm -y

##### 3.2 tar包安装

3.2.1下载安装包

wget http://www.linuxvirtualserver.org/software/kernel-2.6/ipvsadm-1.26.tar.gz

3.2.2解压

tar -xvf ipvsadm-1.26.tar.gz

3.2.3建立软连接

 ln -s /usr/src/kernels/2.6.32-696.13.2.el6.centos.plus.lux.1.x86\_64   /usr/src/linux

建立软连接之前先查看3.10.0-862.11.6.el7.x86\_64 此目录是否存在

若不存在，则需要下载安装

3.2.4 编译安装

**make && make ins**tall

#### 4配置路由

ipvsadm -A -t 172.18.8.6:80 -s rr

ipvsadm -a -t 172.18.8.6:80 -r 172.18.8.5 -g

ipvsadm -a -t 172.18.8.6:80 -r 172.18.8.4 -g

经测试可以通过设置端口号使用不同的端口

##### ipvsadm命令方法

ipvsadm组件定义规则的格式：

#virtual-service-address:是指虚拟服务器的ip 地址

#real-service-address:是指真实服务器的ip 地址

#scheduler：调度方法

#ipvsadm 的用法和格式如下：

ipvsadm -A|E -t|u|f virutal-service-address:port [-s scheduler] [-p[timeout]] [-M netmask]

ipvsadm -D -t|u|f virtual-service-address

ipvsadm -C

ipvsadm -R

ipvsadm -S [-n]

ipvsadm -a|e -t|u|f service-address:port -r real-server-address:port [-g|i|m] [-w weight]

ipvsadm -d -t|u|f service-address -r server-address

ipvsadm -L|l [options]

ipvsadm -Z [-t|u|f service-address]

ipvsadm --set tcp tcpfin udp

ipvsadm --start-daemon state [--mcast-interface interface]

ipvsadm --stop-daemon

ipvsadm -h

#命令选项解释：有两种命令选项格式，长的和短的，具有相同的意思。在实际使用时，两种都可以。

-A --add-service #在内核的虚拟服务器表中添加一条新的虚拟服务器记录。也就是增加一台新的虚拟服务器。

-E --edit-service #编辑内核虚拟服务器表中的一条虚拟服务器记录。

-D --delete-service #删除内核虚拟服务器表中的一条虚拟服务器记录。

-C --clear #清除内核虚拟服务器表中的所有记录。

-R --restore #恢复虚拟服务器规则

-S --save #保存虚拟服务器规则，输出为-R 选项可读的格式

-a --add-server #在内核虚拟服务器表的一条记录里添加一条新的真实服务器记录。也就是在一个虚拟服务器中增加一台新的真实服务器

-e --edit-server #编辑一条虚拟服务器记录中的某条真实服务器记录

-d --delete-server #删除一条虚拟服务器记录中的某条真实服务器记录

-L|-l --list #显示内核虚拟服务器表

-Z --zero #虚拟服务表计数器清零（清空当前的连接数量等）

--set tcp tcpfin udp #设置连接超时值

--start-daemon #启动同步守护进程。他后面可以是master 或backup，用来说明LVS Router 是master 或是backup。在这个功能上也可以采用keepalived 的VRRP 功能。

--stop-daemon #停止同步守护进程

-h --help #显示帮助信息

#其他的选项:

-t --tcp-service service-address #说明虚拟服务器提供的是tcp 的服务[vip:port] or [real-server-ip:port]

-u --udp-service service-address #说明虚拟服务器提供的是udp 的服务[vip:port] or [real-server-ip:port]

-f --fwmark-service fwmark #说明是经过iptables 标记过的服务类型。

-s --scheduler scheduler #使用的调度算法，有这样几个选项rr|wrr|lc|wlc|lblc|lblcr|dh|sh|sed|nq,默认的调度算法是： wlc.

-p --persistent [timeout] #持久稳固的服务。这个选项的意思是来自同一个客户的多次请求，将被同一台真实的服务器处理。timeout 的默认值为300 秒。

-M --netmask #子网掩码

-r --real-server server-address #真实的服务器[Real-Server:port]

-g --gatewaying 指定LVS 的工作模式为直接路由模式（也是LVS 默认的模式）

-i --ipip #指定LVS 的工作模式为隧道模式

-m --masquerading #指定LVS 的工作模式为NAT 模式

-w --weight weight #真实服务器的权值

--mcast-interface interface #指定组播的同步接口

-c --connection #显示LVS 目前的连接 如：ipvsadm -L -c

--timeout #显示tcp tcpfin udp 的timeout 值 如：ipvsadm -L --timeout

--daemon #显示同步守护进程状态

--stats #显示统计信息

--rate #显示速率信息

--sort #对虚拟服务器和真实服务器排序输出

--numeric -n #输出IP 地址和端口的数字形式

##### LVS的10种调度算法

1.Fixed Scheduling Method 静态调服方法

RR #轮询

#调度器通过"轮叫"调度算法将外部请求按顺序轮流分配到集群中的真实服务器上，它均等地对待每一台服务器，而不管服务器上实际的连接数和系统负载。

WRR #加权轮询

#调度器通过"加权轮叫"调度算法根据真实服务器的不同处理能力来调度访问请求。 这样可以保证处理能力强的服务器处理更多的访问流量。调度器 可以自动问询真实服务器的负载情况，并动态地调整其权值。

DH #目标地址hash

#算法也是针对目标IP地址的负载均衡，但它是一种静态映射算法，通过一个散列（Hash）函数将一个目标IP地址映射到一台服务器。

#目标地址散列调度算法先根据请求的目标IP地址，作为散列键（Hash Key）从静态分配的散列表找出对应的服务器，若该服务器是可用的且未超载，将请求发送到该服务器，否则返回空。

SH #源地址hash

#算法正好与目标地址散列调度算法相反，它根据请求的源IP地址，作为散列键（Hash Key）从静态分配的散列表找出对应的服务器，若该服务器是 可用的且未超载，将请求发送到该服务器，否则返回空。

#它采用的散列函数与目标地址散列调度算法的相同。除了将请求的目标IP地址换成请求的源IP地址外，它的算法流程与目标地址散列调度算法的基本相似。在实际应用中，源地址散列调度和目标地址散列调度可以结合使用在防火墙集群中，它们可以保证整个系统的唯一出入口。

2.Dynamic Scheduling Method 动态调服方法

LC #最少连接

#调度器通过"最少连接"调度算法动态地将网络请求调度到已建立的链接数最少的服务器上。 如果集群系统的真实服务器具有相近的系统性能，采用"最小连接"调度算法可以较好地均衡负载。

WLC #加权最少连接

#在集群系统中的服务器性能差异较大的情况下，调度器采用"加权最少链接"调度算法优化负载均衡性能，具有较高权值的服务器将承受较大比例的活动连接负载。调度器可以自动问询真实服务器的负载情况，并动态地调整其权值。

SED #最少期望延迟

#基于wlc算法，举例说明：ABC三台机器分别权重123，连接数也分别是123，name如果使用WLC算法的话一个新请求 进入时他可能会分给ABC中任意一个，使用SED算法后会进行这样一个运算

#A:(1+1)/2

#B:(1+2)/2

#C:(1+3)/3

#根据运算结果，把连接交给C

NQ #从不排队调度方法

#无需列队，如果有台realserver的连接数=0 就直接分配过去，不需要进行sed运算.

LBLC #基于本地的最少连接

# "基于局部性的最少链接" 调度算法是针对目标IP地址的负载均衡，目前主要用于Cache集群系统。

#该算法根据请求的目标IP地址找出该 目标IP地址最近使用的服务器，若该服务器 是可用的且没有超载，将请求发送到该服务器；

#若服务器不存在，或者该服务器超载且有服务器处于一半的工作负载，则用"最少链接"的原则选出一个可用的服务器，将请求发送到该服务器。

LBLCR #带复制的基于本地的最少连接

#"带复制的基于局部性最少链接"调度算法也是针对目标IP地址的负载均衡，目前主要用于Cache集群系统。

#它与LBLC算法的不同 之处是它要维护从一个 目标IP地址到一组服务器的映射，而LBLC算法维护从一个目标IP地址到一台服务器的映射。

#该算法根据请求的目标IP地址找出该目标IP地址对应的服务器组，按"最小连接"原则从服务器组中选出一台服务器，

#若服务器没有超载，将请求发送到该服务器；若服务器超载，则按"最小连接"原则从这个集群中选出一 台服务器 ，将该服务器加入到服务器组中，将请求发送到该服务器。同时，当该服务器组有一段时间没有被修改， 将最忙的服务器从服务器组中删除，以降低复制的程度。

#### 5查看配置

ipvsadm -L -n

#### 6开启防火墙

开启防火墙80端口

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --reload

#### 7.realserver配置

#### 8配置广播网络

cd /etc/sysconfig/network-scripts/

cp /etc/sysconfig/network-scripts/ifcfg-lo /etc/sysconfig/network-scripts/ifcfg-lo:1

vi /etc/sysconfig/network-scripts/ifcfg-lo:1

DEVICE =lo:1

IPADDR=272.18.8.6

NETMASK=255.255.255.255

ONBOOT=yes

NAME=loopback9

重启网络

service network restart

#### 10关闭VIP ARP响应

##### 有两种配置方式

###### 一、修改配置文件

vi /etc/sysctl.conf

添加

net.ipv4.conf.eno16777736.arp\_ignore=1

net.ipv4.conf.eno16777736.arp\_announce=2

重启网卡

###### 二、脚本配置

脚本

#!/bin/bash

#description : start realserver

VIP=125.38.38.64

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

case "$1" in

start)

echo " start LVS of REALServer"

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

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

;;

stop)

/sbin/ifconfig lo:0 down

echo "close LVS Directorserver"

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 "Usage: $0 {start|stop}"

exit 1

esac

##### 理解arp\_ignore与arp\_announce参数详细

arp响应限制arp\_ignore:

0 - (默认值): 回应任何网络接口上对任何本地IP地址的arp查询请求

1 - 只回答目标IP地址是来访网络接口本地地址的ARP查询请求

2 -只回答目标IP地址是来访网络接口本地地址的ARP查询请求,且来访IP必须在该网络接口的子网段内

3 - 不回应该网络界面的arp请求，而只对设置的唯一和连接地址做出回应

4-7 - 保留未使用

8 -不回应所有（本地地址）的arp查询

arp响应限制arp\_announce:对网络接口上，本地IP地址的发出的，ARP回应，作出相应级别的限制: 确定不同程度的限制,宣布对来自本地源IP地址发出Arp请求的接口

0 - (默认) 在任意网络接口（eth0,eth1，lo）上的任何本地地址

1 -尽量避免不在该网络接口子网段的本地地址做出arp回应. 当发起ARP请求的源IP地址是被设置应该经由路由达到此网络接口的时候很有用.此时会检查来访IP是否为所有接口上的子网段内ip之一.如果改来访IP不属于各个网络接口上的子网段内,那么将采用级别2的方式来进行处理.

2 - 对查询目标使用最适当的本地地址.在此模式下将忽略这个IP数据包的源地址并尝试选择与能与该地址通信的本地地址.首要是选择所有的网络接口的子网中外出访问子网中包含该目标IP地址的本地地址. 如果没有合适的地址被发现,将选择当前的发送网络接口或其他的有可能接受到该ARP回应的网络接口来进行发送.

在dr模式中我们的realserver配置arp\_ignore为1：意思是不是eno16777736所在ip的请求，我不与应答。arp\_announce为2：意思是我不对外公布我有出来eno16777736所在ip之外的任何ip，避免主动宣告ip使arp广播包发送过来，做出应答

## 三、LVS 测试

在 两台 realserver 添加socket 服务 通过客户端访问可以将请求分发服务器，可以访问到realserver部署的socket服务。

## 四、Keepalived安装步骤

### 服务器四台

192.168.148.128 Keepalived+LVS master

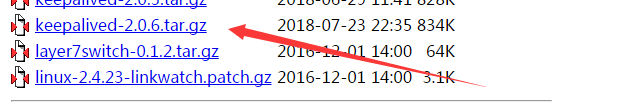
192.168.148.131 Keepalived+LVS slave

192.168.148.129 socket 服务

192.168.148.130 socket 服务

### 一、去官网下载最新的安装包

http://www.keepalived.org/software/



### 二、解压

tar -xvf keepalived-2.0.6.tar.gz

### 三、添加依赖包

yum install -y openssl-devel

yum install -y openssl openssl-devel

yum install zlib

yum install zlib-devel

yum -y install libnl libnl-devel

yum install -y libnfnetlink-devel

或

linux下Zlib的安装与使用

https://blog.csdn.net/qq\_21383435/article/details/79539034

安装openssl和openssl-devel

https://blog.csdn.net/thanklife/article/details/55097429

### 四、编译安装

cd keepalived-2.0.6

./configure --prefix=/usr/local/keepalived&&make&&make install

五、配置

1添加为系统服务

# keepalived启动脚本变量引用文件，默认文件路径是/etc/sysconfig/，也可以不做软链接，直接修改启动脚本中文件路径即可（安装目录下）

cp /usr/local/keepalived/etc/sysconfig/keepalived /etc/sysconfig/keepalived

# 将keepalived主程序加入到环境变量（安装目录下）

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

# keepalived启动脚本（源码目录下），放到/etc/init.d/目录下就可以使用service命令便捷调用

cp /home/app/keepalived-2.0.6/keepalived/etc/init.d/keepalived /etc/init.d/keepalived

# 将配置文件放到默认路径下

mkdir /etc/keepalived

cp /usr/local/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/keepalived.conf

2 设置开机启动

chkconfig keepalived on

service keepalived start|stop|restart

3 配置配置文件

cd /etc/keepalived/

master 服务器

vi keepalived.conf

global\_defs {

notification\_email {

root@localhost

}

notification\_email\_from lvs@localhost

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

state MASTER

interface ens33 #网卡名根据实际情况填写

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.148.188/24 dev ens33 label ens33:1 #虚拟ip地址

}

}

virtual\_server 192.168.148.188 90 {

delay\_loop 6

lb\_algo rr # 负载均衡算法

nat\_mask 255.255.255.0 # 网关

lb\_kind DR #lvs 模式

persistence\_timeout 50

protocol TCP

real\_server 192.168.148.129 90 {

weight 1

TCP\_CHECK { #realserver的状态检测设置部分，单位是秒

connect\_timeout 3 #表示3秒无响应超时

nb\_get\_retry 3 #表示重试次数

delay\_before\_retry 3 #表示重试间隔

connect\_port 90

}

}

real\_server 192.168.148.130 90 {

weight 1

TCP\_CHECK { #realserver的状态检测设置部分，单位是秒

connect\_timeout 3 #表示3秒无响应超时

nb\_get\_retry 3 #表示重试次数

delay\_before\_retry 3 #表示重试间隔

connect\_port 90

}

}

slave

global\_defs {

notification\_email {

root@localhost

}

notification\_email\_from lvs@localhost

smtp\_server 127.0.0.1

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

state BACKUP

interface ens33

virtual\_router\_id 51

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.148.188/24 dev ens33 label ens33:1

}

}

virtual\_server 192.168.148.188 90 {

delay\_loop 6

lb\_algo rr

lb\_kind DR

persistence\_timeout 50

protocol TCP

real\_server 192.168.148.129 90 {

weight 1

TCP\_CHECK { #realserver的状态检测设置部分，单位是秒

connect\_timeout 3 #表示3秒无响应超时

nb\_get\_retry 3 #表示重试次数

delay\_before\_retry 3 #表示重试间隔

connect\_port 90

}

}

real\_server 192.168.148.130 90 {

weight 1

TCP\_CHECK { #realserver的状态检测设置部分，单位是秒

connect\_timeout 3 #表示3秒无响应超时

nb\_get\_retry 3 #表示重试次数

delay\_before\_retry 3 #表示重试间隔

connect\_port 90

}

}

}

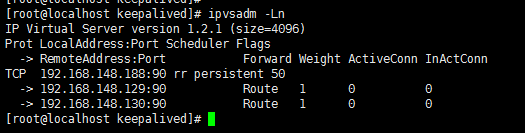
}

### 五、启动服务

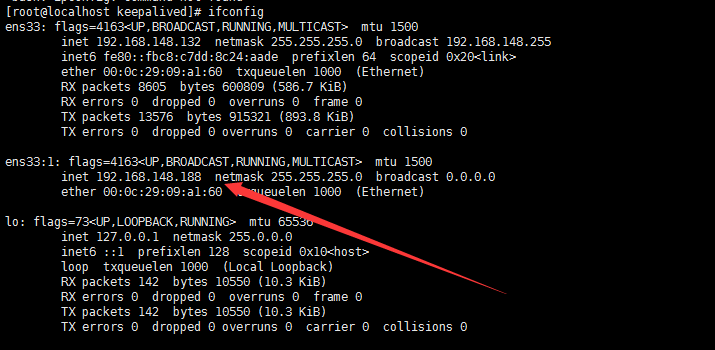
/var/ftp/centos6.5/media/Packages

service keepalived start

### 六、查看状态，自动生成路由信息证明启动成功



master 服务器会自动配置虚拟ip



### 七 日志查看

tail -f /var/log/messages

### 八 两台socket服务器配置

配置广播网络 和 关闭VIP ARP响应 参考lvs 安装步骤

## 五、Keepalived 测试

master Keepalived 服务停掉以后，服务仍可用，关闭其中一台socket 服务，服务仍可用。

iptables -L -n

iptables -L -n

vi /etc/sysconfig/iptables