集群

什么是集群

一组通过高速网络互连的计算组,并以单一系统的模式加以管理

将很多服务器集中起来一起,提供同一种服务,在客户端看起来就像只有一台服务器

可以再付出较低成本的情况下获得在性能,可靠性,灵活性方面相对较高的收益

任务调度是集群系统中的核心技术

集群分类

\*高性能计算集群(HPC)

\*负载均衡集群(LB)

\*高可用集群(HA)

案例:ipvsadm命令使用

1.1问题

准备一台Linux服务器,安装ipvsadm软件包,练习使用ipvssadm命令,实现如下功能:

\*使用命令添加基于TCP一些的集群服务

\*在集群中添加若干后端真实服务器

\*实现同一客户端访问,调度器分配固定服务器

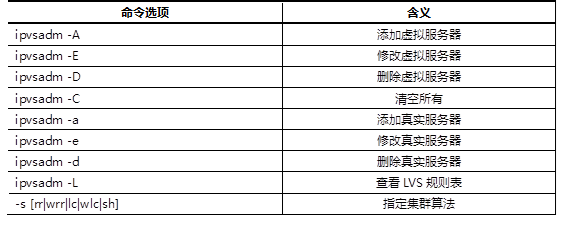
\*会使用ipvaadm实现规则的增删改

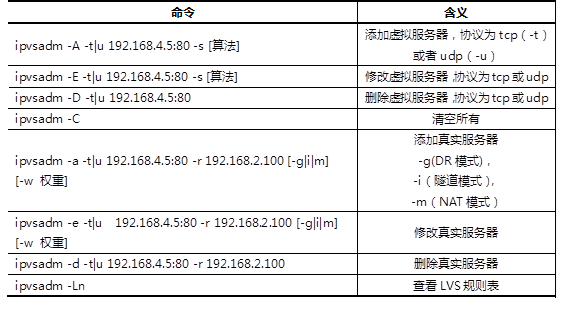
\*保存ipcsadm规则

1.2方案

安装ipvsadm软件包,关于ipvsadm的用法参考man ipvsadm

常用ipvsadm命令语法格式表





1.3步骤

实现此案例需要按照以下步骤进行

步骤一:使用命令增,删,改LVS集群规则

[1]创建LVS虚拟机集群服务器(算法为加权轮询)

[root@proxy ~]# yum -y install ipvsadm

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:80 -s wrr

[root@proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 wrr

[2]为集群添加若干real server

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.100 -w 1

[root@proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 wrr

-> 192.168.2.100:80 router 1 0 0

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -m -w 2 //-m表示NAT模式 i 隧道模式 默认DR模式

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.201 -m -w 3

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.202 -m -w 4

[3]修改集群服务器设置(修改调度算法,将加权轮询修改为轮询)

[root@proxy ~]# ipvsadm -E -t 192.168.4.5:80 -s rr

[root@proxy ~]# ipvsadm -Ln

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.4.5:80 rr

-> 192.168.2.100:80 router 1 0 0

-> 192.168.2.200:80 masq 2 0 0

-> 192.168.2.201:80 masq 2 0 0

-> 192.168.2.202:80 masq 1 0 0

[4]修改read server(使用-g选项,将模式改为DR模式)

[root@proxy ~]# ipvsadm -e -t 192.168.4.5:80 -r 192.168.2.202 -g

[5]查看LVS状态

[root@proxy ~]# ipvsadm -Ln

[6]创建另一个集群（算法为最少连接算法；使用-m选项，设置工作模式为NAT模式）

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:3306 -s lc

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:3306 -r 192.168.2.100 -m

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:3306 -r 192.168.2.200 -m

[7]永久保存所有规则

[root@proxy ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm

[8]清空所有规则

[root@proxy ~]# ipvsadm -C

案例2:部署LVS-NAT集群

2.1问题

使用LVS实现NAT模式的集群调度服务器，为用户提供Web服务：

集群对外公网IP地址为192.168.4.5

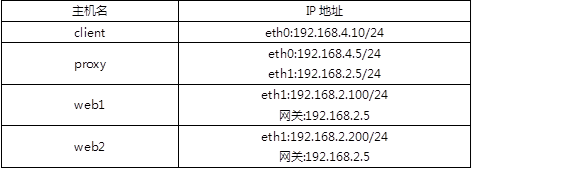
调度器内网IP地址为192.168.2.5

真实Web服务器地址分别为192.168.2.100、192.168.2.200

使用加权轮询调度算法，真实服务器权重分别为1和2

2.2方案

实验拓扑结构主机配置详细表

使用4台虚拟机，1台作为Director调度器、2台作为Real Server、1台客户端，拓扑结构如图-1所示，注意：web1和web2必须配置网关地址。

2.3步骤

实现此案例需要按照如下配置

步骤一:配置基础环境

[1]设置web服务器

[root@web1 ~]# yum -y install httpd

[root@web1 ~]# echo "192.168.2.100" > /var/www/html/index.html

[2]启动web服务器软件

[root@web1 ~]# systemctl restart httpd

[3]关闭防火墙与selinux

[root@web1 ~]# systmctl stop firewalld

[root@web1 ~]# setenforce 0

步骤二:部署LVS-NAT模式调度

[1]确认调度器的路由转发功能(如果已经开启,可以忽略)

[root@proxy ~]# echo 1 > /proc/sys/net/ipv4/ip\_forward

[root@proxy ~]# cat /proc/sys/net/ipv4/ip\_forward

1

[root@proxy ~]# echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

#修改配置文件，设置永久规则

[2]创建集群服务

[root@proxy ~]# yum -y install ipvsadm

[root@proxy ~]# ipvsadm -A -t 192.168.4.5:80 -s wrr

[3]添加规则列表,并保存规则

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.100 -w 1 -m

[root@proxy ~]# ipvsadm -a -t 192.168.4.5:80 -r 192.168.2.200 -w 1 -m

[4]查看规则列表,并保存规则

[root@proxy ~]# ipvsadm -Ln

[root@proxy ~]# ipvsadm-save -n > /etc/sysconfig/ipvsadm

步骤三:客户端测试

客户端使用curl命令反复连接http://192.168.4.5，查看访问的页面是否会轮询到不同的后端真实服务器。

案例3:部署LVS-DR集群

3.1问题

使用LVS实现DR模式的集群调度服务器，为用户提供Web服务：

客户端IP地址为192.168.4.10

LVS调度器VIP地址为192.168.4.15

LVS调度器DIP地址设置为192.168.4.5

真实Web服务器地址分别为192.168.4.100、192.168.4.200

使用加权轮询调度算法，web1的权重为1，web2的权重为2

说明：

CIP是客户端的IP地址；

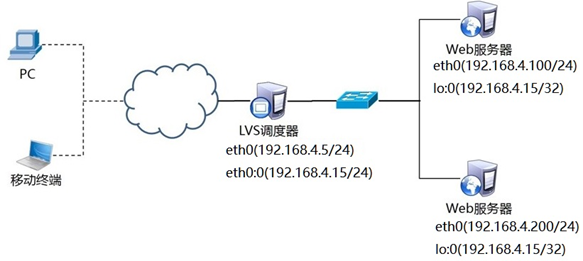
VIP是对客户端提供服务的IP地址；

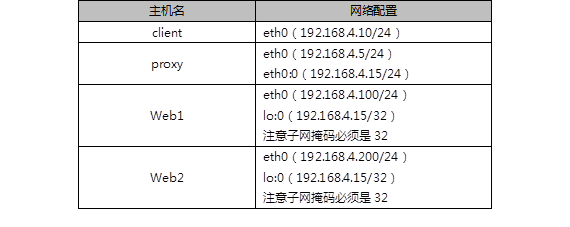
RIP是后端服务器的真实IP地址；

DIP是调度器与后端服务器通信的IP地址（VIP必须配置在虚拟接口）

3.2方案

使用4台虚拟机，1台作为客户端、1台作为Director调度器、2台作为Real Server，拓扑结构如图-2所示。实验拓扑结构主机配置细节如表-4所示。





3.3步骤

实现此案例需要按照如下步骤进行。

说明：

CIP是客户端的IP地址；

VIP是对客户端提供服务的IP地址；

RIP是后端服务器的真实IP地址；

DIP是调度器与后端服务器通信的IP地址（VIP必须配置在虚拟接口）。

步骤一：配置实验网络环境

[1]设置Proxy代理服务器的VIP和DIP

注意：为了防止冲突，VIP必须要配置在网卡的虚拟接口！！！

[root@proxy ~]# cd /etc/sysconfig/network-scripts/

[root@proxy ~]# cp ifcfg-eth0{,:0}

[root@proxy ~]# vim ifcfg-eth0

TYPE=Ethernet

BOOTPROTO=none

NAME=eth0

DEVICE=eth0

ONBOOT=yes

IPADDR=192.168.4.5

PREFIX=24

[root@proxy ~]# vim ifcfg-eth0:0

TYPE=Ethernet

BOOTPROTO=none

DEFROUTE=yes

NAME=eth0:0

DEVICE=eth0:0

ONBOOT=yes

IPADDR=192.168.4.15

PREFIX=24

[root@proxy ~]# systemctl restart network

[2]设置web1服务器网络参数

[root@web1 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.100/24 connection.autoconnect yes

[root@web1 ~]# nmcli connection up eth0

接下来给web1配置VIP地址。

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

[root@web1 ~]# cd /etc/sysconfig/network-scripts/

[root@web1 ~]# cp ifcfg-lo{,:0}

[root@web1 ~]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

防止地址冲突的问题：

这里因为web1也配置与代理一样的VIP地址，默认肯定会出现地址冲突；

sysctl.conf文件写入这下面四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应，这样防止地址冲突的问题。

[root@web1 ~]# vim /etc/sysctl.conf

#手动写入如下4行内容

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

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

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

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

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

[root@web1 ~]# sysctl -p

重启网络服务,设置防火墙SeLinux

[root@web1 ~]# systemctl stop NetworkManager

[root@web1 ~]# systemctl restart network

常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。（非必须的操作）

[root@web1 ~]# systemctl stop NetworkManager

[root@web1 ~]# systemctl restart network

[3]设置web2网络服务参数

[root@web2 ~]# nmcli connection modify eth0 ipv4.method manual \

ipv4.addresses 192.168.4.200/24 connection.autoconnect yes

[root@web2 ~]# nmcli connection up eth0

接下来给web2配置VIP地址

注意：这里的子网掩码必须是32（也就是全255），网络地址与IP地址一样，广播地址与IP地址也一样。

[root@web2 ~]# cd /etc/sysconfig/network-scripts/

[root@web2 ~]# cp ifcfg-lo{,:0}

[root@web2 ~]# vim ifcfg-lo:0

DEVICE=lo:0

IPADDR=192.168.4.15

NETMASK=255.255.255.255

NETWORK=192.168.4.15

BROADCAST=192.168.4.15

ONBOOT=yes

NAME=lo:0

防止地址冲突的问题：

这里因为web1也配置与代理一样的VIP地址，默认肯定会出现地址冲突；

sysctl.conf文件写入这下面四行的主要目的就是访问192.168.4.15的数据包，只有调度器会响应，其他主机都不做任何响应，这样防止地址冲突的问题。

[root@web2 ~]# vim /etc/sysctl.conf

#手动写入如下4行内容

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

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

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

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

#当有arp广播问谁是192.168.4.15时，本机忽略该ARP广播，不做任何回应

#本机不要向外宣告自己的lo回环地址是192.168.4.15

[root@web2 ~]# sysctl -p

重启网络服务,设置防火墙与selinux

[root@web1 ~]# systemctl restart network

[root@web1 ~]# ifconfig

常见错误：如果重启网络后未正确配置lo:0，有可能是NetworkManager和network服务有冲突，关闭NetworkManager后重启network即可。（非必须的操作）

[root@web1 ~]# systemctl stop NetworkManager

[root@web1 ~]# systemctl restart network

步骤二:proxy调度器安装软件并部署LVS-DR模式调度器

[1]安装软件

[root@proxy ~]# yum -y install ipvsadm

[2]清理之前实验规则,创建新的集群服务器规则

[root@proxy ~]# ipvsadm -C #清空所有规则

[root@proxy ~]# ipvsadm -A -t 192.168.4.15:80 -s wrr

[3]添加真实服务器,(-g参数设置LVS工作模式为DR模式,-W设置权重)

[root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.100 -g -w 1

[root@proxy ~]# ipvsadm -a -t 192.168.4.15:80 -r 192.168.4.200 -g -w 1

[4]查看规则列表,保存规则

[root@proxy ~]# ipvsadm -Ln

TCP 192.168.4.15:80 wrr

-> 192.168.4.100:80 Route 1 0 0

-> 192.168.4.200:80 Route 2 0 0

步骤三:客户测试

客户端使用curl命令反复连接http://192.168.4.15，查看访问的页面是否会轮询到不同的后端真实服务器。

扩展知识：默认LVS不带健康检查功能，需要自己手动编写动态检测脚本，实现该功能：(参考脚本如下，仅供参考)

[root@proxy ~]# vim check.sh

#!/bin/bash

VIP=192.168.4.15:80

RIP1=192.168.4.100

RIP2=192.168.4.200

while :

do

for IP in $RIP1 $RIP2

do

curl -s http://$IP &>/dev/vnull

if [ $? -eq 0 ];then

ipvsadm -Ln |grep -q $IP || ipvsadm -a -t $VIP -r $IP

else

ipvsadm -Ln |grep -q $IP && ipvsadm -d -t $VIP -r $IP

fi

done

sleep 1

done