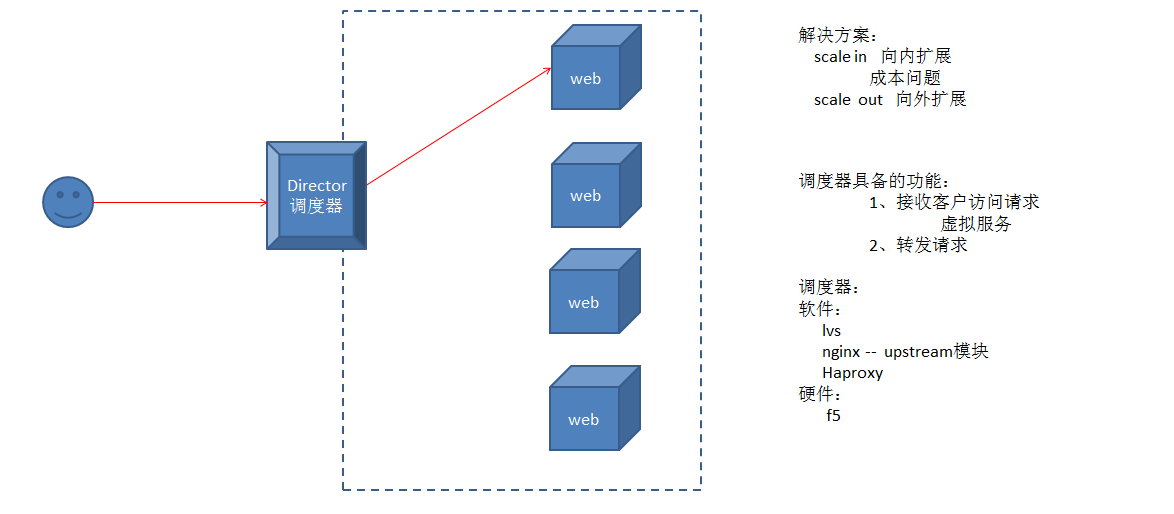
## 集群概述 cluster

类型：

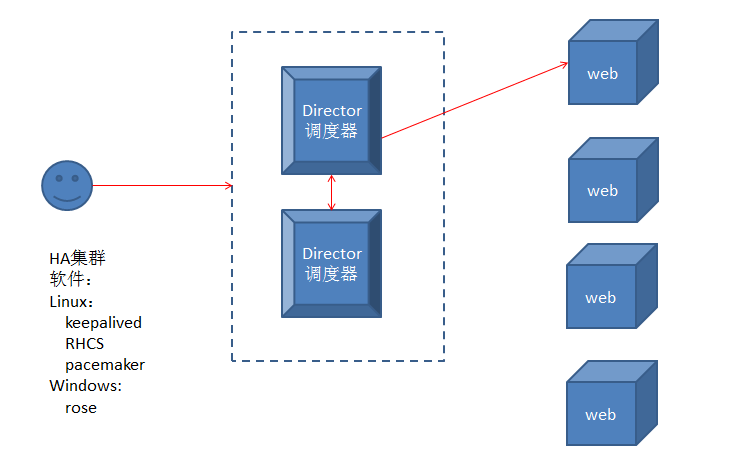
LB Load Balance 负载均衡集群

作用：提升服务器的并发能力



HA High Avalibility 高可用集群

作用：提升服务在线时间/可用性



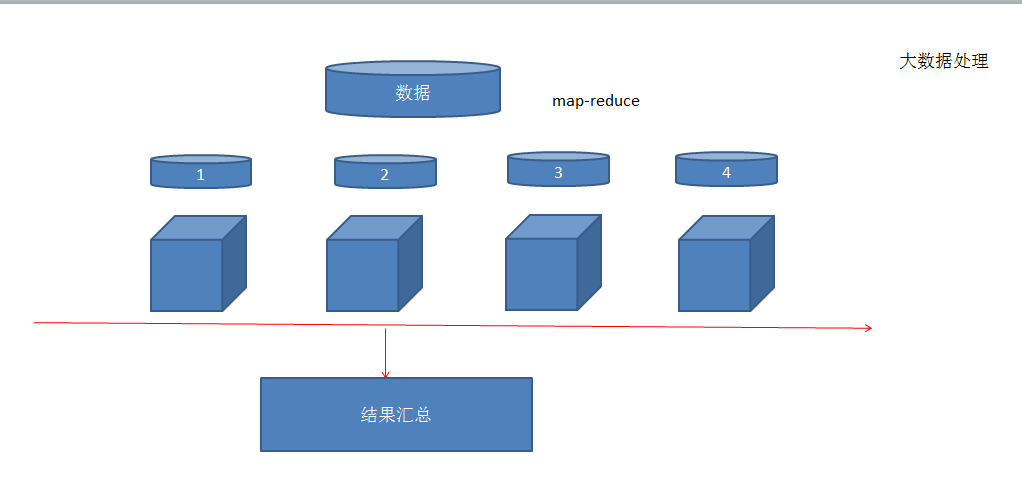
可用性 MTBRF

在线时间/(在线时间＋故障恢复时间)

HPC High Perforce Compute 高性能集群

作用：解决海量数据处理的问题

软件：Hadoop



## LB负载均衡集群

软件：

lvs Linux Virutal Service

直接工作在linux内核；速度快

nginx

haproxy

一、lvs --- Linux Virtual Service

**1、lvs调度算法**

1) rr round robin 轮询

会话session保持

1) 方案一：在服务器间同步会话信息

2) 方案二：使用共享存储保存会话 redis数据库

2) wrr 基于权重的轮询 weight

3) lc least connection 最少连接

4) wlc 基于权重的最少连接 默认

5) sh source hash 源hash

根据客户端IP计算hash值，相同hash值的请求转发到同一个后端服务器

一定程度可以解决会话保持问题

6) dh destination hash 目的地址hash

根据目的IP地址计算hash值，后续所有请求定位同一个目的IP上

适用于后端是缓存服务器，提升缓存命中率

**二、lvs工作模式**

1、NAT模式

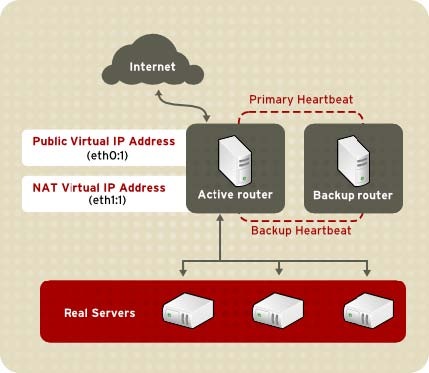
1) 访问请求、响应数据都要经过调度器

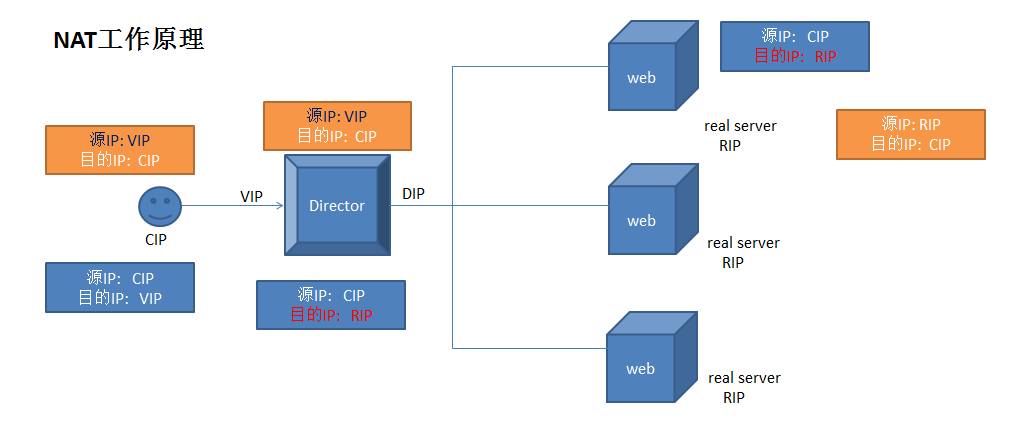
2) 调度器与客户端要在同一个网络中，调度器与real server需要在另一个网络中

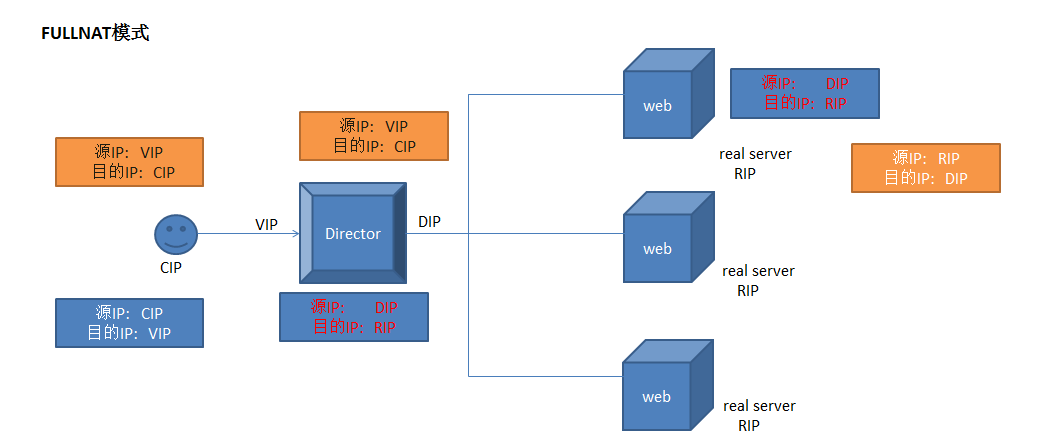
3) 所有real server的网关要指向DIP

4) 后端real server操作系统可以是任意类型

5) 调度器需要启用路由转发功能







2、DR直接路由模式

1) 调度器、real server要在同一个网络中

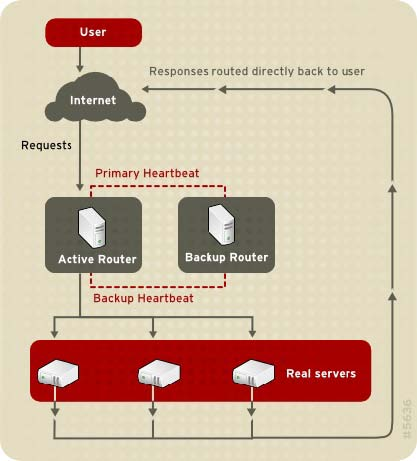
2) real server的网关要指向网络中真实的网关

3) 所有real server要存在VIP

arp\_ignore: 1 只响应针对物理网卡的ARP请求

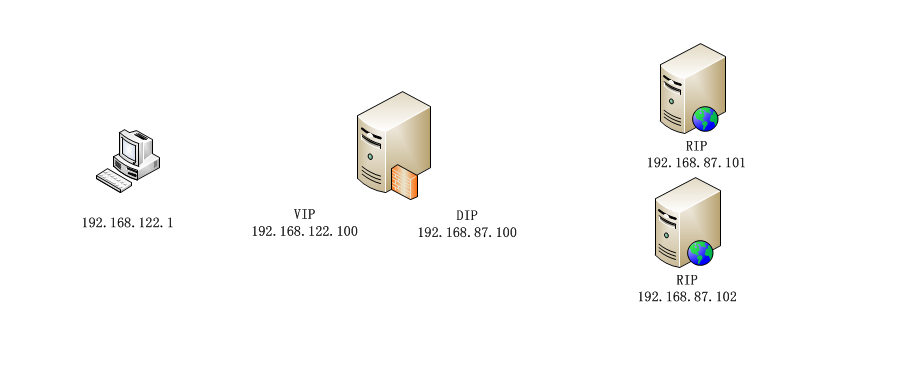
arp\_announce：2 以适当的IP地址的响应数据

4) 后端real server只能是类Linux系统



3、TUN隧道模式

### 示例01： 部署lvs nat模式



1、配置时间同步、按图配置IP地址

[root@web\_server\_01 ~]# ip addr show eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 52:54:29:9f:bd:b2 brd ff:ff:ff:ff:ff:ff

inet 192.168.87.101/24 brd 192.168.87.255 scope global eth0

valid\_lft forever preferred\_lft forever

inet6 fe80::5054:29ff:fe9f:bdb2/64 scope link

valid\_lft forever preferred\_lft forever

[root@web\_server\_01 ~]#

[root@web\_server\_01 ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 192.168.87.100 0.0.0.0 UG 100 0 0 eth0

192.168.87.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0

[root@web\_server\_01 ~]#

2、安装lvs软件

[root@lvs\_director ~]# yum install -y ipvsadm.x86\_64

3、启用路由转发功能

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

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

1

4、建立集群/虚拟服务

[root@lvs\_director ~]# ipvsadm -A -t 192.168.122.100:80 -s rr

[root@lvs\_director ~]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.122.100:80 rr

5、添加后端real server

[root@lvs\_director ~]# ipvsadm -a -t 192.168.122.100:80 -r 192.168.87.101 -m

[root@lvs\_director ~]# ipvsadm -a -t 192.168.122.100:80 -r 192.168.87.102 -m

[root@lvs\_director ~]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.122.100:80 rr

-> 192.168.87.101:80 Masq 1 0 0

-> 192.168.87.102:80 Masq 1 0 0

6、测试访问

[root@lvs\_director ~]# ipvsadm -L -n -c

IPVS connection entries

pro expire state source virtual destination

TCP 01:25 TIME\_WAIT 192.168.122.1:39715 192.168.122.100:80 192.168.87.101:80

TCP 01:24 TIME\_WAIT 192.168.122.1:39713 192.168.122.100:80 192.168.87.101:80

TCP 01:23 TIME\_WAIT 192.168.122.1:39712 192.168.122.100:80 192.168.87.102:80

TCP 01:37 TIME\_WAIT 192.168.122.1:39717 192.168.122.100:80 192.168.87.101:80

TCP 01:25 TIME\_WAIT 192.168.122.1:39716 192.168.122.100:80 192.168.87.102:80

TCP 01:24 TIME\_WAIT 192.168.122.1:39714 192.168.122.100:80 192.168.87.102:80

[root@lvs\_director ~]#

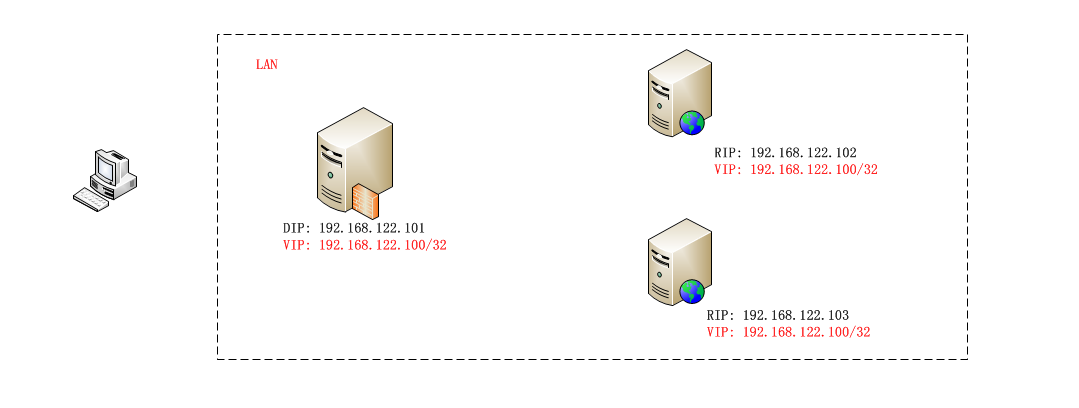
[root@lvs\_director ~]# ipvsadm -S > /tmp/ipvs.rule //保存规则

[root@lvs\_director ~]# ipvsadm -C //清除所有规则

[root@lvs\_director ~]# ipvsadm -R < /tmp/ipvs.rule //恢复规则

[root@lvs\_director ~]# ipvsadm -L -n

### 示例02： 部署lvs DR模式



1、配置所有real server

1) 配置vip

[root@web\_server\_01 ~]# ip addr add dev lo 192.168.122.100/32

[root@web\_server\_01 ~]# ip a

2) 修改arp\_ignore

[root@web\_server\_01 ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore

3) 修改arp\_announce

[root@web\_server\_01 ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce

2、配置调度器

1) 配置vip

[root@lvs\_director ~]# ip addr add dev lo 192.168.122.100/32

[root@lvs\_director ~]# ip addr show

2) 创建虚拟服务

[root@lvs\_director ~]# yum install -y ipvsadm

[root@lvs\_director ~]# ipvsadm -A -t 192.168.122.100:80 -s rr

3) 添加后端的real server

[root@lvs\_director ~]# ipvsadm -a -t 192.168.122.100:80 -r 192.168.122.102 -g

[root@lvs\_director ~]# ipvsadm -a -t 192.168.122.100:80 -r 192.168.122.103 -g

[root@lvs\_director ~]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

TCP 192.168.122.100:80 rr

-> 192.168.122.102:80 Route 1 0 0

-> 192.168.122.103:80 Route 1 0 0

## 持久性连接

-p <timeout>

将虚拟服务定义为持久性服务

一定程度解决会话保持问题

[root@lvs\_director ~]# ipvsadm -E -t 192.168.122.100:80 -s rr -p 300

[root@lvs\_director ~]# ipvsadm -L -n

通过防火墙标记的建立虚拟服务

针对多端口的服务，实现端口亲缘性

1、vsftpd服务器配置

pasv\_min\_port=20000

pasv\_max\_port=30000

2、调度器

针对ftp的访问数据标记

[root@lvs\_director ~]# iptables -t mangle -A PREROUTING -d 192.168.122.100 -p tcp --dport 20:21 -j MARK --set-mark 77

[root@lvs\_director ~]# iptables -t mangle -A PREROUTING -d 192.168.122.100 -p tcp --dport 20000:30000 -j MARK --set-mark 77

创建虚拟服务

[root@lvs\_director ~]# ipvsadm -A -f 77 -s rr -p 300

[root@lvs\_director ~]# ipvsadm -a -f 77 -r 192.168.122.102 -g

[root@lvs\_director ~]# ipvsadm -a -f 77 -r 192.168.122.103 -g

[root@lvs\_director ~]# ipvsadm -L -n

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port Forward Weight ActiveConn InActConn

FWM 77 rr

-> 192.168.122.102:0 Route 1 0 0

-> 192.168.122.103:0 Route 1 0 0

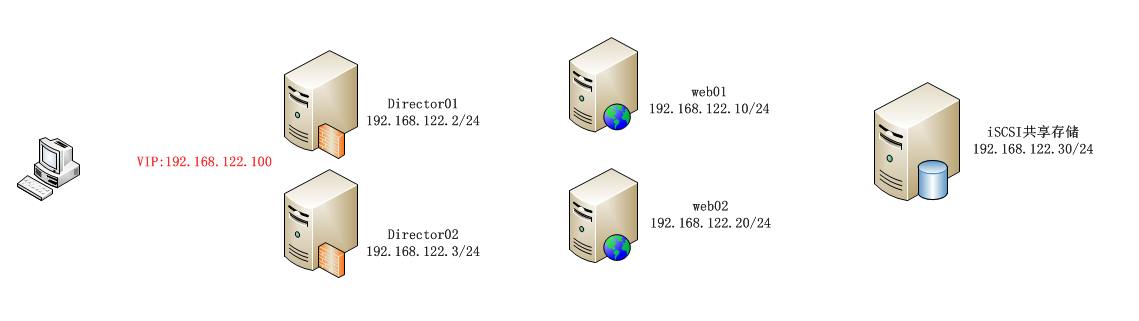
## keepalived + lvs + iSCSI共享存储

keepalived + lvs集群：

1) 实现调度器的高可用

2) 生成lvs负载均衡的规则

3) 针对后端的real server实现健康状态检查



1、主机名、SELinux, 防火墙、时间同步

2、配置iscsi共享存储

1) 准备逻辑卷

[root@shared\_storage ~]# pvcreate /dev/vdb

Physical volume "/dev/vdb" successfully created.

[root@shared\_storage ~]# vgcreate vg01 /dev/vdb

Volume group "vg01" successfully created

[root@shared\_storage ~]# lvcreate -L +6G -n lv01 vg01

Logical volume "lv01" created.

[root@shared\_storage ~]# lvscan

ACTIVE '/dev/vg01/lv01' [6.00 GiB] inherit

2) 安装targetcli软件

[root@shared\_storage ~]# yum install -y targetcli

3) 配置共享存储

[root@shared\_storage ~]# targetcli

/> /backstores/block create name=bk\_01 dev=/dev/vg01/lv01 //创建后端存储

/> /iscsi create iqn.2018-09.com.linux:jifang01-jigui03-IBM-disk //创建一个iscsi的共享名称

/> /iscsi/iqn.2018-09.com.linux:jifang01-jigui03-ibm-disk/tpg1/luns create /backstores/block/bk\_01 //绑定后端存储

/> /iscsi/iqn.2018-09.com.linux:jifang01-jigui03-ibm-disk/tpg1/acls create iqn.2018-09.com.linux:client //设置客户端iscsi认证名称

/> /iscsi/iqn.2018-09.com.linux:jifang01-jigui03-ibm-disk/tpg1/portals create ip\_address=192.168.122.30 //设置监听IP及端口

启动target服务

[root@shared\_storage ~]# systemctl start target.service

[root@shared\_storage ~]# systemctl enable target.service

[root@shared\_storage ~]# netstat -antp | grep :3260

tcp 0 0 192.168.122.30:3260 0.0.0.0:\* LISTEN -

3、配置web\_server连接共享存储

1) 安装客户端软件

[root@web\_server\_01 ~]# yum install -y iscsi-initiator-utils

2) 设置客户端的iscsi名称

[root@web\_server\_01 ~]# cat /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-09.com.linux:client

3) 探索共享存储设备

[root@web\_server\_01 ~]# iscsiadm -m discovery -t st -p 192.168.122.30

192.168.122.30:3260,1 iqn.2018-09.com.linux:jifang01-jigui03-ibm-disk

4) 连接共享存储设备

[root@web\_server\_01 ~]# iscsiadm -m node -T iqn.2018-09.com.linux:jifang01-jigui03-ibm-disk -p 192.168.122.30 -l

[root@web\_server\_01 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 6G 0 disk

5) 使用共享存储设备

[root@web\_server\_01 ~]# yum install -y httpd

[root@web\_server\_01 ~]# vim /etc/fstab

/dev/sda1 /var/www/html ext4 defaults,\_netdev 0 0

[root@web\_server\_01 ~]# mount -a

第二台web\_server同样连接共享存储

4、所有的real server配置VIP, 修改arp的内核参数

[root@web\_server\_01 ~]# ip addr add dev lo 192.168.122.100/32

[root@web\_server\_01 ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore

[root@web\_server\_01 ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce

5、配置调度器

1) 安装软件

[root@director\_01 ~]# yum install -y keepalived ipvsadm.x86\_64

2) 编辑keepalived的配置文件，生成负载均衡规则、实现调度的高可用

[root@director\_01 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id director01

}

//实现调度器的高可用

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 88

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

//生成lvs的负载均衡规则

virtual\_server 192.168.122.100 80 {

delay\_loop 6

lb\_algo rr

lb\_kind DR

persistence\_timeout 300

protocol TCP

real\_server 192.168.122.10 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

}

}

real\_server 192.168.122.20 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

}

}

}

将keepalived的配置文件拷贝一份

[root@director\_01 ~]# rsync -av /etc/keepalived/keepalived.conf 192.168.122.3:/etc/keepalived/keepalived.conf

编辑第二台调度器的配置文件

[root@director\_02 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id director02

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth0

virtual\_router\_id 88

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

virtual\_server 192.168.122.100 80 {

delay\_loop 6

lb\_algo rr

lb\_kind DR

persistence\_timeout 300

protocol TCP

real\_server 192.168.122.10 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

}

}

real\_server 192.168.122.20 80 {

weight 1

TCP\_CHECK {

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

connect\_port 80

}

}

}

5、分别在两台调度器上启动keepalived的服务

[root@director\_01 ~]# systemctl start keepalived.service

[root@director\_01 ~]# systemctl enable keepalived.service

测试：

1) 服务正常访问

2) 停止主调度的keepalived服务，测试VIP的故障转移failover

3) 后端服务器的健康状态检测