# Keepalived高可用软件

Keepalived软件起初是专为LVS负载均衡软件设计的，用来管理并监控LVS集群系统中各个服务节点的状态，后来又加入了可以实现高可用的VRRP功能。因此，keepalived除了能够管理LVS软件外，还可以作为其他服务的高可用解决方案软件。

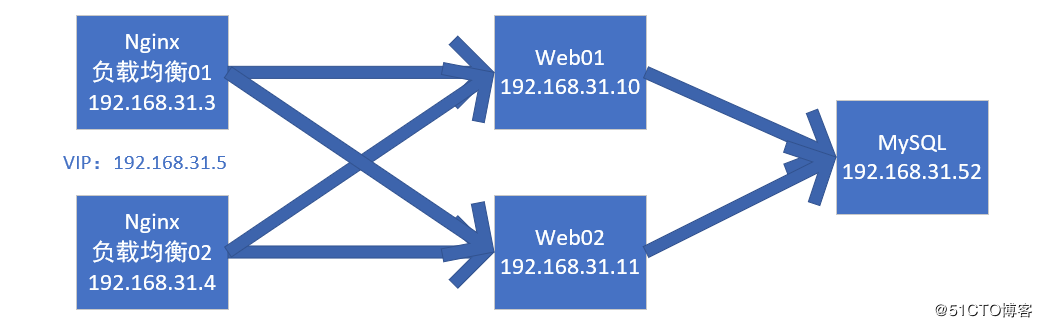
keepalived软件主要是通过VRRP协议实现高可用功能的。VRRP是Virtual Router Redundancy Protocol（虚拟路由冗余协议）的缩写，VRRP出现的目的就是为了解决静态路由的单点故障问题的，它能保证当个别节点宕机时，整个网络可以不间断地运行。所以，keepalived一方面具有配置管理LVS的功能，同时还具有对LVS下面节点进行健康检查的功能，另一方面也可以实现系统网络服务的高可用功能。

# Keepalived高可用故障切换转移原理

Keepalived高可用服务对之间的故障切换转移，是通过VRRP来实现的。在keepalived服务工作时，主Master节点会不断地向备节点发送（多播的方式）心跳消息，用来告诉备Backup节点自己还活着。当主节点发生故障时，就无法发送心跳的消息了，备节点也因此无法继续检测到来自主节点的心跳了。于是就会调用自身的接管程序，接管主节点的IP资源和服务。当主节点恢复时，备节点又会释放主节点故障时自身接管的IP资源和服务，恢复到原来的备用角色。

# Keepalived高可用实验环境说明

如下图所示，前端有两台的Nginx负载均衡器，用来分发接收到客户端的请求。在前文已经配置好了Nginx01，Nginx02也是一样的配置。现在要在两个Nginx负载均衡器上做高可用配置，Nginx01作为主节点，Nginx02作为备节点。



# 安装并启用keepalived

keepalived的安装非常简单，直接使用yum来安装即可。

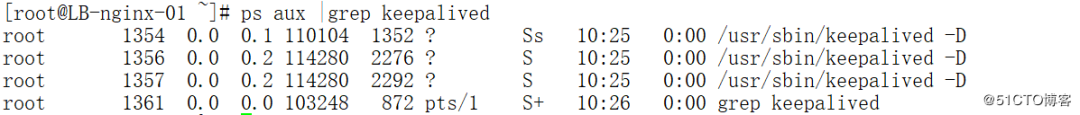
yum install keepalived -y

安装之后，启动keepalived服务，顺便把keepalived写入开机启动的脚本里面去。。

/etc/init.d/keepalived star

echo "/etc/init.d/keepalived start" >>/etc/rc.local

启动之后会有三个进程，没问题之后可以关闭keepalived软件，接下来要修改keepalived的配置文件。



# 修改keepalived配置文件并且重启keepalived服务

/etc/init.d/keepalived stop #关闭keepalived服务

vim /etc/keepalived/keepalived.conf #用vim打开编辑

主节点的配置文件

! Configuration File for keepalived

global\_defs {

notification\_email {

acassen@firewall.loc

failover@firewall.loc

sysadmin@firewall.loc

}

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

smtp\_server 192.168.200.1

smtp\_connect\_timeout 30

router\_id lb01

}

vrrp\_instance VI\_1 {

state MASTER

interface eth1

virtual\_router\_id 55

priority 150

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 123456

}

virtual\_ipaddress {

192.168.31.5/24 dev eth1 label eth1:1

}

}

......

备节点的配置文件

! Configuration File for keepalived

global\_defs {

notification\_email {

acassen@firewall.loc

failover@firewall.loc

sysadmin@firewall.loc

}

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

smtp\_server 192.168.200.1

smtp\_connect\_timeout 30

router\_id lb02

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth1

virtual\_router\_id 55

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 123456

}

virtual\_ipaddress {

192.168.31.5 dev eth1 label eth1:1

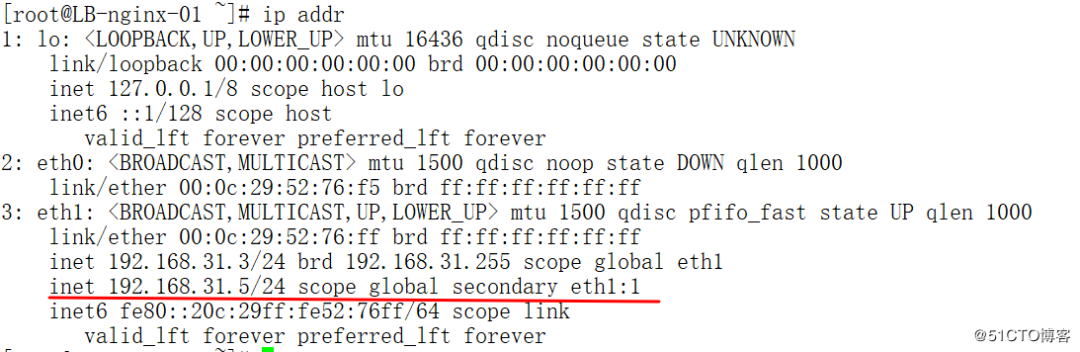
}

}

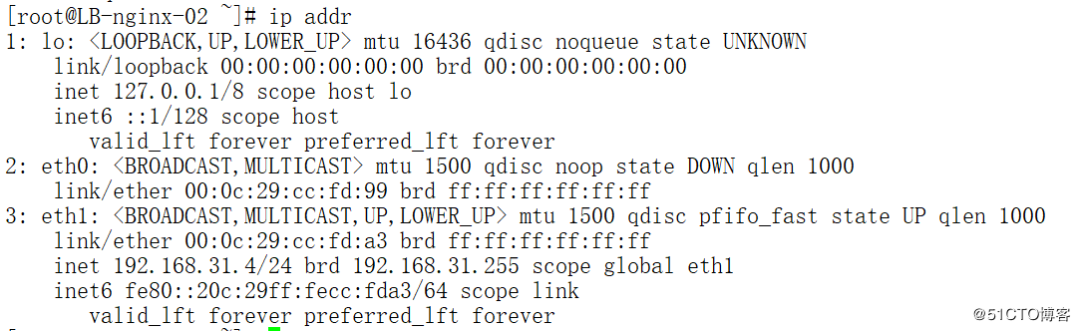
注解：修改配置文件主要就是上面加粗的几个地方，下面说明一下那几个参数的意思：

* router\_id 是路由标识，在一个局域网里面应该是唯一的；
* vrrp\_instance VI\_1{…}这是一个VRRP实例，里面定义了keepalived的主备状态、接口、优先级、认证和IP信息；
* state 定义了VRRP的角色，interface定义使用的接口，这里我的服务器用的网卡都是eth1,根据实际来填写，virtual\_router\_id是虚拟路由ID标识，一组的keepalived配置中主备都是设置一致，priority是优先级，数字越大，优先级越大，auth\_type是认证方式，auth\_pass是认证的密码
* virtual\_ipaddress ｛…｝定义虚拟IP地址，可以配置多个IP地址，这里我定义为192.168.31.5，绑定了eth1的网络接口，虚拟接口eth1:1

修改好主节点之后，保存退出，然后启动keepalived，几分钟内会生成一个虚拟IP：192.168.31.5

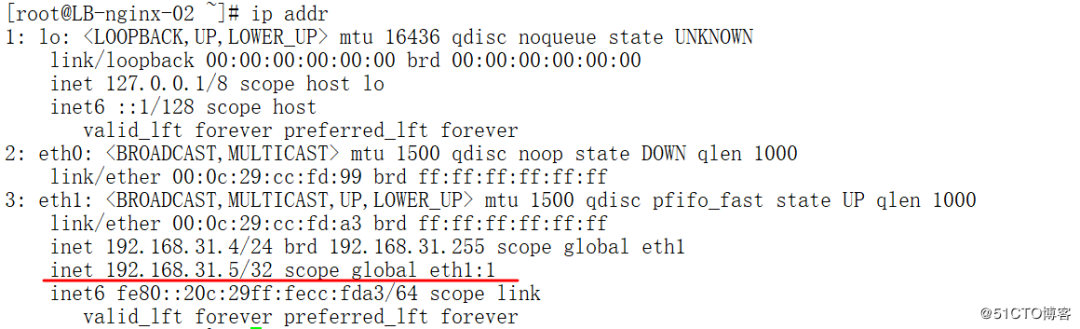


然后修改备节点的配置文件，保存退出后启动keepalived，不会生成虚拟IP，如果生成那就是配置文件出现了错误。备节点和主节点争用IP资源，这个现象叫做“裂脑”。

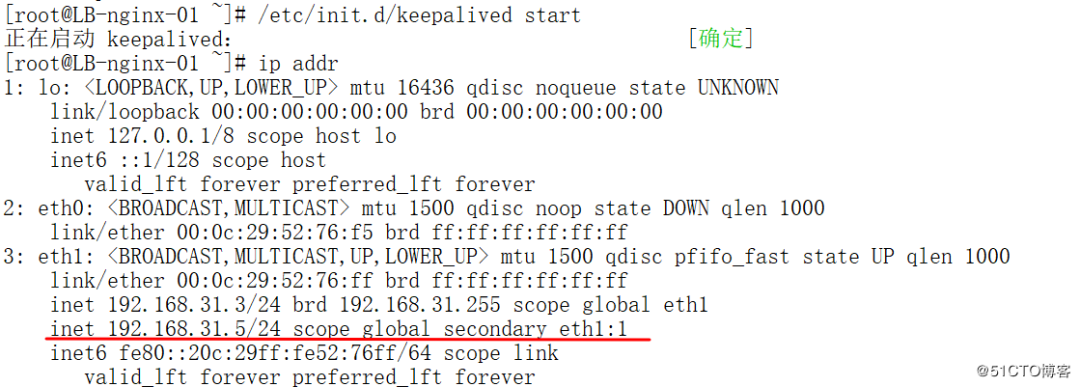


# 6.进行高可用的主备服务器切换实验

停掉主节点的keepalived服务，查看备节点会不会生成VIP：192.168.31.5

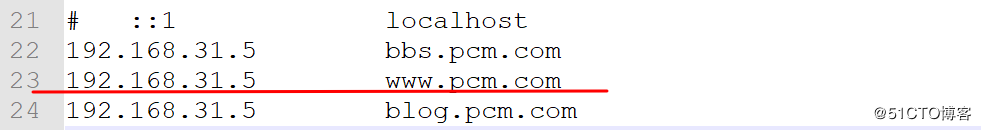


启动主节点的keepalived服务，然后查看主节点和备节点的VIP，主节点应该会抢夺回来VIP：



# 7.搭配Nginx负载均衡来测试

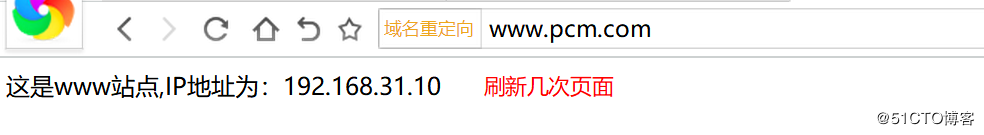
修改windows的hosts文件，把域名指向到VIP上

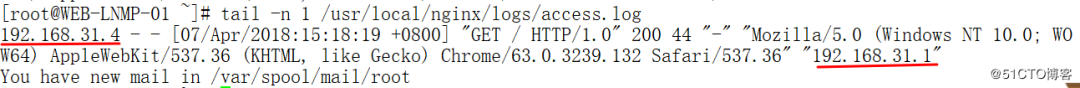


然后用浏览器打开www.pcm.com的页面，在web01上查看access.log日志记录到的客户端IP地址



可以看到日志记录到的客户端的IP地址是192.168.31.1，反向代理服务器是主服务器192.168.31.3.下面我们停止keepalived服务，看备节点会不会接替主节点的VIP和服务。





可以看到，备节点确实接替了主节点的工作。重新启用主节点，实验的结果就不验证了。

# 8.编写Nginx Web服务的守护脚本

上面的实验测试有一个问题就是，我们是用Nginx做负载均衡分发请求的数据包的。如果主节点的Keepalived服务正常运行，而Nginx运行异常，那么将会出现Nginx负载均衡服务失灵，无法切换到Nginx负载均衡器02上，后端的Web服务器无法收到请求。所以，我们应该要检测Nginx的服务是否正常运行，如果不是正常运行，应该停掉Keepalived的服务，这样才能自动切换到备节点上。

我们可以通过检测80端口是否开启来判定Nginx的运行情况，2秒钟检测一次，脚本如下

#!/bin/bash

while true

do

if [ $(netstat -tlnp|grep nginx|wc -l) -ne 1 ]

then

/etc/init.d/keepalived stop

fi

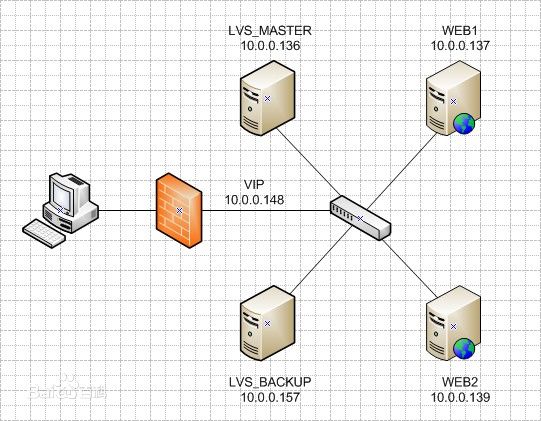
sleep 2

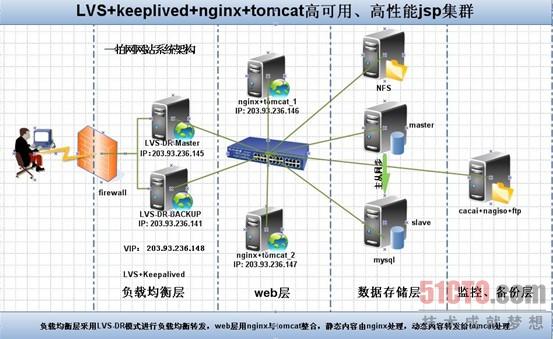
done

实验的结果可以后台执行命令之后然后停止Nginx服务检验

# 高可用HA

1. 需要心跳机制探测后端RS是否提供服务。
   1. 探测down，需要从lvs中删除该RS
   2. 探测发送从down到up，需要从lvs中再次添加RS。
2. Lvs DR，需要主从（HA）





Keepalived : 提供高可用

Vrrp协议，实现ＩＰ的漂移

Zookeeper：

选举、心跳机制提供接管

# Keepalived 配置文件

至少两台服务器

1. 地址：/etc/keepalived.conf
2. 帮助手册：man keepalived.conf
3. 编辑之前，需要备份：cp –a keepalived.conf backup. keepalived.conf
4. 编辑：

! Configuration File for keepalived

global\_defs {

notification\_email {

root@localhost#发送提醒邮件的目标地址可有多个

goldbin@126.com

}

notification\_email\_from test@localhost#发送邮件的from地址，可以随意写，邮件地址不存在都无所谓

smtp\_server 127.0.0.1#邮件服务的地址，一般写本地

smtp\_connect\_timeout 30

router\_id LVS\_DEVEL

}

vrrp\_instance VI\_1 {

state BACKUP# MASTER 主 和 从

interface eth0#VIP需要绑定的网卡名称

virtual\_router\_id 51

priority 101#优先级 主的优先级要高

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 1111

}

virtual\_ipaddress {

192.168.1.200/24 dev eth0 label eth0:0 #设置VIP

}

}

virtual\_server 192.168.1.200 80 {#设置虚拟lvs服务，VIP PORT

delay\_loop 6

lb\_algo rr#调度算法wrr

lb\_kind DR#lvs的模式

nat\_mask 255.255.255.0

persistence\_timeout 50 同一个IP地址在50秒内lvs转发给同一个后端服务器

protocol TCP

real\_server 192.168.1.139 80 {#设置真实服务器的心跳机制 RID PORT

weight 1#权重

HTTP\_GET {#心跳检测的方式

url {

path /#心跳检查的地址

status\_code 200#心跳检查返回的状态

}

connect\_timeout 2 #超时时间

nb\_get\_retry 3#重复检查3次

delay\_before\_retry 1#每隔1秒钟再次检查

}

}

real\_server 192.168.1.134 80 {#第二个真实服务器设置

weight 1

HTTP\_GET {

url {

path /

status\_code 200

}

connect\_timeout 2

nb\_get\_retry 3

delay\_before\_retry 1

}

}

}

# Keepalived的启动和日志

Service keepalived start启动

tail /var/log/message 查看日志。

课后作业：

1. 思考nginx + keepalived 高可用（不需要lvs集群）？
2. Lvs + keepalived +msyql （主主复制）的HA集群。