

# Nginx + Keepalived一高可用

## 1、环境简介：

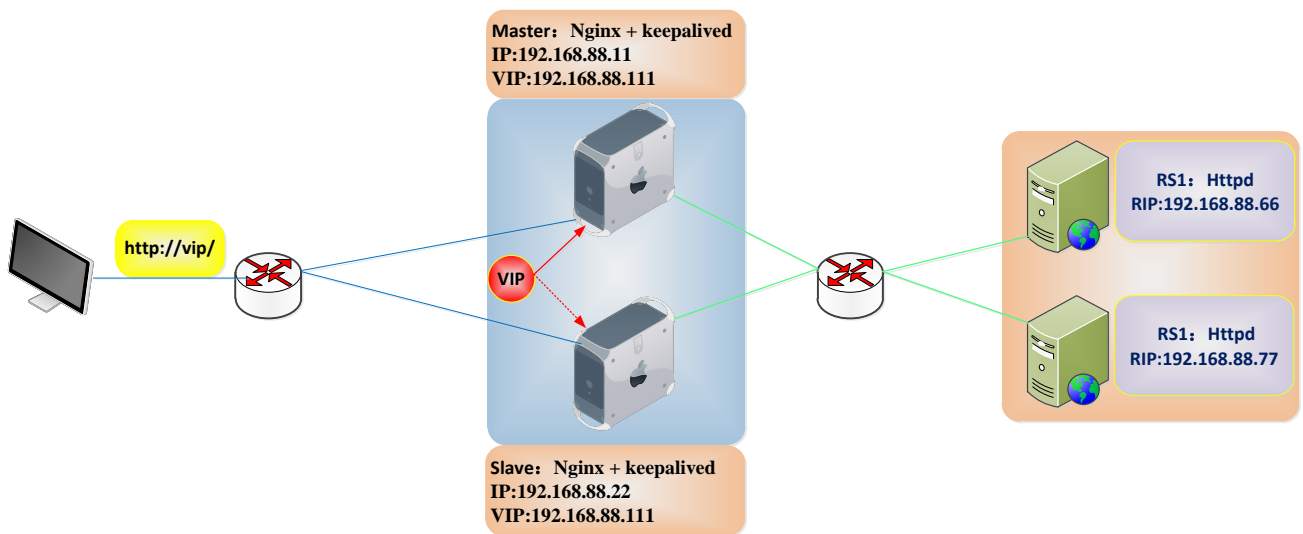
VS1: CentOS 6.8 + Keepalived v1.2.13 + nginx/1.10.2 + eth0:192.168.88.11

VS2: CentOS 6.8 + Keepalived v1.2.13 + nginx/1.10.2 + eth0:192.168.88.22

RS1: CentOS 6.8 + Apache/2.2.15 + eth0:192.168.88.66

RS2: CentOS 6.8 + Apache/2.2.15 + eth0:192.168.88.77

## 2、网络拓扑：



## 3、实验前准备：

注：需配置好本地 yum 源，以下操作需在各主机上进行

(1)修改各主机的/etc/hosts 文件，实现主机名解析：

```
[root@vs1 ~]#cat /etc/hosts
192.168.88.11    vs1.maochen.com  vs1
192.168.88.22    vs2.maochen.com  vs2
192.168.88.66    rs1.maochen.com  rs1
192.168.88.77    rs2.maochen.com  rs2
```

(2)添加主机密钥，实现各主机无密钥登录：

```
[root@vs1 ~]#cat key.sh
#!/bin/bash
HOST=('vs1' 'vs2' 'rs1' 'rs2')
for i in {0..3}
do
```

```

NAME=${HOST[$i]}

[ -e /root/.ssh/id_rsa.pub ] || ssh-keygen -f /root/.ssh/id_rsa -P ""

ssh-copy-id -i /root/.ssh/id_rsa.pub root@${NAME}

scp /etc/hosts ${NAME}:/etc/

done

[root@vs1 ~]#sh key.sh

```

(3)确保各个主机的时间同步(不一定准时，但必须相同):

```

[root@vs1 ~]#service ntpd start
[root@vs2 ~]#ntpdate vs2
[root@vs1 ~]#date;ssh vs2 "date"          #检测时间是否同步

```

(4)确保 iptables 和 selinux 关闭:

```

[root@vs1 ~]#service iptables start
[root@vs1 ~]#sed -i 's/SELINUX=.* /SELINUX=disabled/g' /etc/sysconfig/selinux
[root@vs1 ~]#setenforce 0 ##临时关闭 selinux, 上一条为永久关闭

```

(5)安装 keepalived 软件(本实验选择 yum 安装)

```

[root@vs1 ~]#yum install keepalived nginx -y
[root@rs1 ~]#yum install httpd -y

```

4、修改配置文件:

(1)修改/etc/keepalived/keepalived.conf 配置文件，在 master 上的设置如下，在 slave 上的设置仅修改红线部分的

```

! Configuration File for keepalived
global_defs {
notification_email {
    root@localhost          ###定义节点状态发生变化时发送通知的目标邮箱地址
}
notification_email_from keepalived_test@nwc.com  ###定义发件人的信息
smtp_server 127.0.0.1      ###定义邮箱服务器地址
smtp_connect_timeout 30   ###定义邮箱服务器的链接超时时间
router_id node72          ###定义本机的路由器 ID, 自定义即可
vrrp_mcast_group4 224.0.32.18 ###定义心跳信息通告的组播地址，同一个集群的组播地址一样
}
vrrp_script check {
    script "killall -0 nginx"  ###定义判断 nginx 服务是否正常的脚本，script 后可加脚本路径
    interval 1                ###检测时间间隔
    weight -10                 ###脚本执行失败则节点优先级减 10
}

```

```

fail 2          ###检测两次才算失败
rise 1          ###检测一次成功就算成功
}

vrrp_instance VI_1 {    ###定义一个虚拟路由器的实例，实例名称为 VI_1
state MASTER          ###定义本节点在该虚拟路由器实例中的初始角色，MASTER 还是 BACKUP
interface eth0         ###虚拟路由器的工作的接口
virtual_router_id 32    ###虚拟路由器 ID(也就是 VRID)，用以区别不同的虚拟路由器实例
priority 100           ###定义当前节点在该虚拟路由器实例中优先级，MASTER 要比 BACKUP 节点的优先级高
advert_int 1           ###定义通告信息发送的时间间隔
authentication {       ###定义认证机制和认证的密钥
    auth_type PASS      ###PASS 为简单字符认证，建议使用 PASS 即可
    auth_pass 12345678  ###认证的密钥，最大有效值为 8 位长度
}
track_script {
    check               ###在 vrrp 实例内部调用 track_script 追踪脚本的执行
}

virtual_ipaddress {    ###定义该虚拟路由器的 VIP
    192.168.88.111/24 dev eth0 label eth0:0
    ### VIP 为 192.168.88.111，定义在 eth0 接口的 eth0:0 别名上
}

track_interface {
###定义要监控的接口，接口出现故障，则节点会转为 FAULT 状态，触发重新选举，实现资源转移
    eth0
}

notify_master "/etc/keepalived/script/notify.sh master"    #状态转化为 master 触发脚本
notify_backup "/etc/keepalived/script/notify.sh backup"    #状态转化为 slave 触发脚本
notify_fault "/etc/keepalived/script/notify.sh fault"      #状态转化 fault 触发脚本
}

```

(2)修改/etc/nginx/conf.d/default.conf 配置文件，在 master 和 slave 上的修改相同。主要是利用 nginx 的代理功能(upstream 模块)

```

upstream web {
    server 192.168.88.66;
    server 192.168.88.77;
}

server {
    listen      80 default_server;
    listen      [::]:80 default_server;

```

```

location / {
    proxy_pass http://web;
    root      /usr/share/nginx/html;
    index     index.html;
}
}

```

## 5、在 keepalived 中调用自定义的脚本实现节点状态转移时的通知机制

```

[root@vs1 ~]mkdir /etc/keepalived/script
[root@vs1 ~]vim /etc/keepalived/script/notify.sh
#!/bin/bash
#
receiver='root@localhost'
notify() {
    mailsubject="$(hostname) to $1,vip floating."
    content="$(date + '%F %T') vrrp state transion, $(hostname) changed to be $1"
    echo "$content" | mail -s "$mailsubject" $receiver
}
case $1 in
master)
    notify master;;
backup)
    notify backup;;
fault)
    notify fault;;
*)
    echo "Usage $(basename $0) {master|backup|fault}"
    exit 1;;
esac
[root@vs1 ~]chmod +x /etc/keepalived/script/notify.sh

```

## 7、在 Real Server 上的配置：

```

[root@rs1 ~]#echo "<h1>RS1</h1>" > /var/www/html/index.html
[root@rs1 ~]#service httpd restart
[root@rs2 ~]#echo "<h1>RS2</h1>" > /var/www/html/index.html
[root@rs2 ~]#service httpd restart

```

## 8、启动服务，模拟故障，检测 IP 可用

```
##开启 keepalived 服务:
```

```
[root@vs1 ~]#service keepalived start
[root@vs1 ~]#ifconfig
[root@vs2 ~]#service keepalived start
[root@test ~]#curl http://192.168.88.111
[root@test ~]#curl http://192.168.88.111
##模拟故障，关闭 vs1 的 keepalived:
[root@vs1 ~]#service keepalived stop (或者 service nginx stop)
[root@vs1 ~]#ip a
##检测 web 服务是否可用
[root@test ~]# curl http://192.168.88.111
```

**参考链接:** <http://www.178linux.com/56546>