

包括阿里云在内的很多云环境，因为不支持浮动 IP 广受诟病。目前阿里云在 VPC 网络下发布了 HAVIP，能够实现 arp 宣告 IP。这样也就让自己搭建 HA 成为了可能，有幸拿到了内测权限体验了一下。(classical 网络依然不支持)

测试环境：

1 VPC: 192.168.1.0/24
2 ECS:
3 nginx1: 192.168.1.1
4 nginx2:192.168.1.2
5 HAVIP:192.168.1.3
6 绑定到 havip 的公网 EIP: 121.43.187.37

配置完毕后的拓扑如下：

环境搭建完毕后，登陆主备 ECS 服务器，分别配置 nginx+keepalived

```
[root@Nginx1 ~]# yum install nginx keepalived -y
```

MASTER 服务器 (nginx1) 配置文件/etc/keepalived/keepalived.conf 内容以及解释如下：

! Configuration File for keepalived

#配置 global_defs，主要用于标示机器，以及故障时通知

```
global_defs {  
    router_id Nginx1
```

```
}
```

#配置 vrrp_script，主要用于健康检查，以及检查失败后执行的动作。

```
vrrp_script chk_nginx {
```

#健康检查脚本，当脚本返回值不为 0 时认为失败

```
    script "/etc/keepalived/ck_ng.sh"
```

#检查频率，以下配置每 2 秒检查 1 次

```
    interval 2
```

#当检查失败后，将 vrrp_instance 的 priority 减小 5

```
    weight -5
```

#连续监测失败 3 次，才认为真的健康检查失败。并调整优先级

```
    fall 3
```

#连续监测 2 次成功，就认为成功。但不调整优先级

```
    rise 2
```

```
}
```

#定义对外提供服务的 VIP vrrp_instance 配置

```
vrrp_instance VI_1 {
```

#指定 vrrp_instance 的初始状态，是 MASTER 还是 BackUP 主要还是看优先级。

```
    state MASTER
```

```

#指定 vrrp_instance 绑定的网卡，最终会通过指定的网卡宣告 VIP
interface eth0
#发送心跳包的源 IP，可使用绑定的网卡 IP，也可以使用本服务器上的其他 IP
unicast_src_ip 192.168.1.1
#相当于 VRID，用于在一个网内区分组播，需要组播域内唯一。
virtual_router_id 55
#本机的优先级，VRID 相同的机器中，优先级最高的会被选举为 MASTER
priority 101
#心跳间隔，下面配置，MASTER 会每隔 1 秒发送一个报文高职组内其他机器，自己还活着。
advert_int 1
#定义主从的验证方式以及密码，一般使用 PASS(最长 8 位，超过了只会识别前 8 位作为密码)
authentication {
    auth_type PASS
    auth_pass aliyun
}
#VIP,在阿里云下就是刚才创建的 HAVIP
virtual_ipaddress {
    192.168.1.3
}
#本 vrrp_instance 所引用的脚本配置，名称就是 vrrp_script 定义的容器名
track_script {
    chk_nginx
}
}
BACKUP 服务器（nginx2）的配置需要修改：
state MASTER 改为 state BACKUP
unicast_src_ip 192.168.1.1 改为 backup 服务器实际的 IP unicast_src_ip 192.168.1.2
priority 101 改小一些，比如 priority 100

```

其它保持一致即可

为了实现 nginx 服务异常的时候能够自动切换,需要自己写一个脚本,脚本没有硬性的要求,能够实现目标即可,这里 监控 nginx 进程数为例:

```

vim /etc/keepalived/ck_ng.sh
#!/bin/bash
#检查 nginx 进程是否存在
count=$(ps -C nginx --no-heading|wc -l)
#进程数等于 0 的时候
if [ "${count}" = "0" ]; then
#尝试启动一次 nginx，停止 2 秒后再次检测

```

```
service nginx start
sleep 2
count=$(ps -C nginx --no-heading|wc -l)
if [ "${count}" = "0" ]; then
#如果启动没成功，就杀掉 keepalive 触发主备切换
/etc/init.d/keepalived stop
fi
fi
#增加脚本的执行权限
chmod +x vim /etc/keepalived/ck_ng.sh

#分别启动 nginx1 和 nginx2 的 nginx&keepalived 服务：
[root@Nginx1 ~]# /etc/init.d/nginx start;/etc/init.d/keepalived start
Starting nginx: [ OK ]
Starting keepalived: [ OK ]
```

测试：

NGINX1/192.168.1.1 的 priority 为 101，NGINX2/192.168.1.2 的 priority 为 100，这时候访问 HAVIP 绑定的 EIP：http://121.43.187.37/可以看到访问到了服务器 NGINX1，

到 HAVIP 控制台查看，192.168.1.1 的服务器为主服务器

这时候我们 KILL 掉 nginx1 服务器的 nginx 服务

查看日志，发送了移除 VIP 的报文：

```
Oct 23 17:20:01 iZ239aqzdi7Z Keepalived_vrrp[25019]: VRRP_Instance(VI_1) sending 0 priority
Oct 23 17:20:01 iZ239aqzdi7Z Keepalived_vrrp[25019]: VRRP_Instance(VI_1) removing protocol VIPs.
Oct 23 17:20:01 iZ239aqzdi7Z Keepalived_healthcheckers[25018]: Netlink reflector reports IP 192.168.1.3 removed
```

这时候访问 <http://121.43.187.37/>，可以看到访问自动切换到了 NGINX2

到 HAVIP 控制台查看，192.168.1.2 的服务器为主服务器

重新启动 nginx1 的 nginx 和 keepalive 服务

查看日志可以看到 keepalive 重新发送了 IP 宣告的报文

```
Oct 23 17:22:14 iZ239aqzdi7Z Keepalived_vrrp[25610]: VRRP_Instance(VI_1) Entering MASTER STATE
Oct 23 17:22:14 iZ239aqzdi7Z Keepalived_vrrp[25610]: VRRP_Instance(VI_1) setting protocol VIPs.
Oct 23 17:22:14 iZ239aqzdi7Z Keepalived_vrrp[25610]: VRRP_Instance(VI_1) Sending gratuitous ARPs on eth0 for 192.168.1.3
Oct 23 17:22:14 iZ239aqzdi7Z Keepalived_healthcheckers[25609]: Netlink reflector reports IP 192.168.1.3 added
Oct 23 17:22:19 iZ239aqzdi7Z Keepalived_vrrp[25610]: VRRP_Instance(VI_1) Sending gratuitous ARPs on eth0 for 192.168.1.3
```

重新访问 http://121.43.187.37/测试，重新访问到了服务器 NGINX1，到 HAVIP 控制台查看，192.168.1.1 的服务器重新夺回了控制权，成为了为主服务器

这样就实现了阿里云环境下的 HA 切换。

实际测试，阿里云的多个 HAVIP 可以绑定到同样的两台机器，可以配置多组 vrrp_instance 来实现双主。或者两台服务器同时为两个业务服务，避免资源浪费。同时能够做到主备