

私有网络

最佳实践

产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

最佳实践

VPC内无外网IP主机通过公网网关访问外网的优化配置

VPC 内通过 keepalived 搭建高可用主备集群

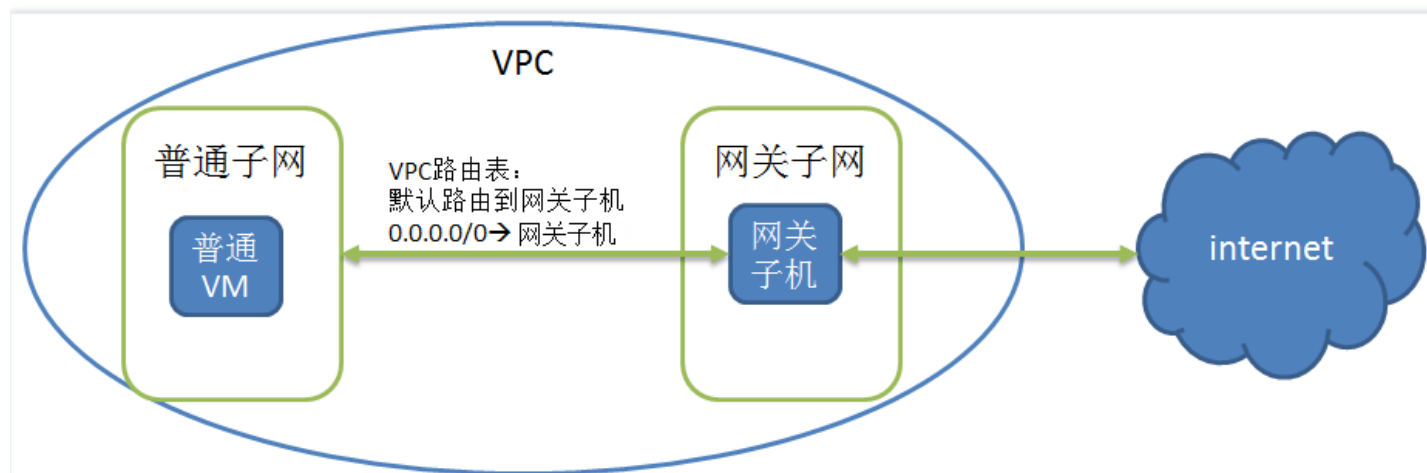
最佳实践

VPC内无外网IP主机通过公网网关访问外网的优化配置

最近更新时间：2017-12-18 12:39:44

1. 环境说明

客户在腾讯云VPC中的部分主机没有外网IP但需要访问外网时，可以通过购买公网网关主机作为其它没有外网IP的主机访问Internet的外网出口。公网网关主机将对出网流量进行源地址转换，所有其他主机访问外网的流量经过公网网关后，源IP都被转换为公网网关主机的IP地址，如下图：



普通CVM没有外网IP，但是可以借助公网网关访问Internet。

要完成上述架构，需要经过后文所述的几个步骤。

2. 架构搭建

2.1. 创建网关子网

由于公网网关只能转发非所在子网的路由转发请求，因此公网网关主机不能和需要借助公网网关访问外网的CVM处于同一个子网下，所以需要先建立一个独立的网关子网。

创建子网

所属网络 hank test 已有2个子网

子网名称	CIDR
网关子网	10.20. 0 .0 / 24

2.2. 购买公网网关

在刚刚创建好的网关子网下购买公网网关，详细步骤请见[购买私有网络的公网网关](#)

2.3. 创建网关子网路由表

网关子网和普通子网不能关联同一张路由表，需要新建一张独立的网关路由表，且网关子网关联该路由表。

新建路由表

名称 gw_route_table

所属网络 hank test(10.20.0.0/24)

路由策略

目的端	下一跳类型	下一跳	操作
Local	Local	Local	- ?

可用区	关联路由表
上海一区	vm_subnet_route
上海一区	默认
上海一区	gw_route_table gw_route_table vm_subnet_route 默认

2.4. 配置普通子网路由表

配置普通子网的路由表，配置默认路由走公网网关主机，使得普通子网内主机能通过公网网关的路由转发能力访问外网。

路由策略		
目的端	下一跳类型	下
Local	Local	L
0.0.0.0/0	云主机（公网网关）	

3. 配置优化

公网网关主机默认配置iptables的nat规则，以及打开kernel的ip_forward，基本的公网网关功能已经完全具备。建议经过下述配置，以达到更好的性能。

1) 通过以下命令将net.ipv4.ip_forward配置写到/etc/sysctl.conf文件中

```
echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
```

2) 通过以下命令将nf_conntrack配置参数调大

```
echo "echo 1048576 > /proc/sys/net/netfilter/nf_conntrack_max" >> /etc/rc.local  
echo "echo 262144 > /sys/module/nf_conntrack/parameters/hashsize" >> /etc/rc.local
```

3) 设置转发的nat规则

```
echo "iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE" >> /etc/rc.local
```

4) 关闭时间戳选项

```
echo "iptables -t mangle -A POSTROUTING -p tcp -j TCPOPTSTRIP --strip-options timestamp" >> /etc/rc.local
```

5) 设置公网网关的rps

在/usr/local/sbin/目录下新建脚本set_rps.sh，将以下代码写入脚本中：

```
#!/bin/bash
mask=0
i=0
cpu_nums=`cat /proc/cpuinfo |grep processor |wc -l`
if(($cpu_nums==0));then
exit 0
fi

nic_queues=`cat /proc/interrupts |grep -i virtio0-input |wc -l`
if(($nic_queues==0));then
exit 0
fi

echo "cpu number" $cpu_nums "nic queues" $nic_queues

mask=$(echo "obase=16;2^$cpu_nums - 1" |bc)
flow_entries=$(echo "$nic_queues * 4096" |bc)

echo "mask = "$mask
echo "flow_entries = "$flow_entries

#for i in {0..$nic_queues}
while (($i < $nic_queues))
do
echo $mask > /sys/class/net/eth0/queues/rx-$i/rps_cpus
echo 4096 > /sys/class/net/eth0/queues/rx-$i/rps_flow_cnt
i=$((i+1))
done

echo $flow_entries > /proc/sys/net/core/rps_sock_flow_entries
```

新建完成后执行以下命令：

```
chmod +x /usr/local/sbin/set_rps.sh  
echo "/usr/local/sbin/set_rps.sh" >> /etc/rc.local
```

完成上述配置后，重启公网网关主机以使配置生效，并在无外网IP的子机上测试是否能够成功访问外网。

VPC 内通过 keepalived 搭建高可用主备集群

最近更新时间：2018-08-24 15:49:07

本文将介绍如何在腾讯云 VPC 内通过 keepalived 搭建高可用主备集群。

实践内容将结合使用腾讯云的新产品“高可用虚拟 IP (HAVIP)”，并作出与组播相关使用建议。

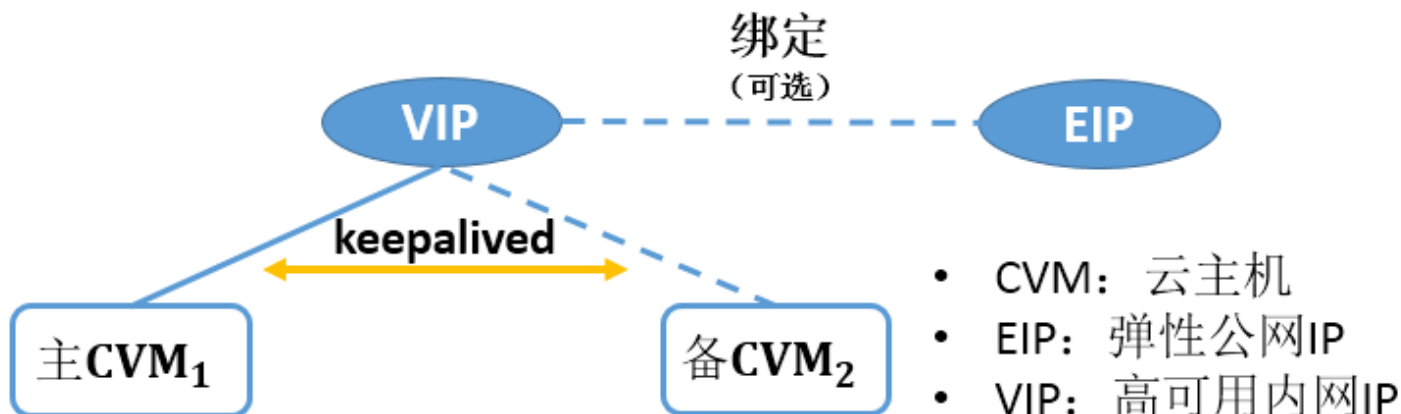
本文小引

为了更清晰地阐述 keepalived 如何在腾讯云服务器上实践，本文：

- 首先对 keepalived 简述，并说明其在云服务器的应用与物理网络的区别。
- 然后开始阐述如何通过何种步骤，达到两种使用模式：
 - 无常主模式，即双机选举主设备的优先级相同；
 - 常主常备模式，即需要让其中一台设备在无故障时尽量当主的场景。常主常备模式较无常主模式增加了主备倒换次数，推荐使用无常主模式（非常主常备模式）
- 本文通过给出若干 keepalived 配置和脚本文件 + 不同场景配置方法 的形式，帮助用户在云服务器上作本次实践。
- 本文实践部分主要介绍 keepalived 的 VRRP Instance 配置为单播 VRRP 报文的用法。

基本原理

通常高可用主备集群包含 2 台服务器，一台主服务器处于某种业务的激活状态（即 Active 状态），另一台备服务器处于该业务的备用状态（即 Standby 状态），它们共享同一个 VIP（Virtual IP），同一时刻 VIP 只在一台主设备上生效，当主服务器出现问题，备用服务器接管 VIP 继续提供服务。高可用主备模式有着广泛的应用，例如：MySQL 主备切换、Nginx Web 接入。（下图所示的EIP绑定功能还未上线，请留意）



高可用主备集群示意图

与物理网络的区别

在传统的物理网络中可以通过 keepalived 的 VRRP 协议协商主备状态，其原理是：主设备周期性发送免费 ARP 报文刷新上联交换机的 MAC 表或终端 ARP 表，触发 VIP 的迁移到主设备上。腾讯云 VPC 内支持部署 keepalived 来搭建主备高可用集群，与物理网络相比，主要区别是：

- 使用的 VIP **必须** 是从腾讯云申请的 [HAVIP](#)。
- 有子网属性，只能被同一个子网下的机器宣告绑定。

本文步骤预览

- 申请 VIP，该 VIP 仅支持在子网内迁移（因此需要保证主备服务器位于同一个子网）。
- 主备服务器安装及配置 keepalived (**1.2.24版本及以上**)，并修改配置文件。**暂不支持VIP对应外网IP。**
- 编辑使用 keepalived 的 notify 机制，借助 notify_action.sh 进行简单的日志记录。
- 验证主备切换时 VIP 是否正常切换。
- TIPS

说明：由于本文给出了数个配置和脚本文件，为了更清晰地说明，**本节先给出各脚本的详细修改步骤**。然后您可以根据后文解决各步骤可能遇到的困难，如VIP的申请等。**修改步骤预览如下：**

```
/etc/keepalived/  
|-- keepalived.conf
```

```
|-- notify_action.sh
```

常主常备用法使用步骤：

主机操作：(常主)

1. 安装 keepalived
2. 在 keepalived 使用的配置目录/etc/keepalived/中，将本目录文件移入，并添加可执行权限 **chmod +x** /etc/keepalived/*.sh; **chmod -x** /etc/keepalived/keepalived.conf
3. 修改 keepalived.conf:
- 0) **state** 初始角色，主机填 MASTER, 备机填 BACKUP
- 1) interface 改成本机网卡名 例如 eth0
- 2) priority 主机值高于备，如：主 50 备 30
- 3) unicast_src_ip 改成本机内网 IP
- 4) unicast_peer 改成对端机器内网 IP
- 5) virtual_ipaddress 改成内网 vip
- 6) track_interface 改成本机网卡名 例如 eth0

备机操作：(常备)

与主机操作类似

```
=====
=====
```

stable 用法使用步骤：(两台设备选举主机优先权相同, 非常主常备) (推荐！)

双机操作相同：

1. 安装 keepalived
2. 在 keepalived 使用的配置目录 /etc/keepalived/ 中，将本目录文件移入，并修改权限 **chmod 744** /etc/keepalived/*.sh; **chmod 644** /etc/keepalived/keepalived.conf
3. 修改 keepalived.conf:
- 0) **state** 初始角色，均填写 BACKUP
- 1) interface 改成本机网卡名 例如 eth0
- 2) priority 两台设备配置大小相同的整数，如 50
- 3) unicast_src_ip 改成本机内网 IP
- 4) unicast_peer 改成对端机器内网 IP
- 5) virtual_ipaddress 改成内网 VIP
- 6) track_interface 改成本机网卡名 例如 eth0

注意：

1. 脚本日志将会写到/var/log/keepalived.log中。日志会占用您的磁盘空间。您可以自行借助 logrotate 等工具处理日志累积的问题
2. keepalived 进程的日志仍会写到 /var/log/message 中。

详细步骤

步骤 1. 申请 VIP

- 申请VIP的操作指南详见文档[高可用虚拟 IP](#)

步骤 2. 主备子机安装 keepalived (1.2.24 版本及以上)

- 安装

以 CentOS 为例：

- yum安装方式

`yum list keepalived` 查看版本号是否符合要求。若是，下一步；若否，用源码包安装方式

`yum -y install keepalived`

- 源码包安装方式

```
tar zxvf keepalived-1.2.24.tar.gz
cd keepalived-1.2.24
./configure --prefix=/
make; make install
chmod +x /etc/init.d/keepalived 【防止出现 env: /etc/init.d/keepalived: Permission denied】
```

步骤 3. 确定主备需求

本文并行介绍两种使用模式：

- 无常主模式，即双机选举主设备的优先级相同；
- 常主常备模式，即需要让其中一台设备在无故障时尽量当主的场景。

常主常备模式较无常主模式增加了主备倒换次数，推荐使用无常主模式（非常主常备模式，又叫双备模式）

步骤 4. 修改配置 keepalived.conf

- 配置文件修改

常主常备模式步骤，以主设备为例，修改 keepalived.conf:

0) state 初始角色，主机填 MASTER, 备机填 **BACKUP**

1) **interface** 改成本机网卡名 例如 eth0

2) **priority** 主机值高于备，如：主 50 备 30

3) **unicast_src_ip** 改成本机内网 IP

4) **unicast_peer** 改成对端机器内网 IP

5) **virtual_ipaddress** 改成内网 VIP

6) **track_interface** 改成本机网卡名 例如 eth0

非常主常备步骤，双机改法相同，修改 keepalived.conf:

- 0) state 初始角色，均填写 **BACKUP**
- 1) **interface** 改成本机网卡名 例如 eth0
- 2) **priority** 两台设备配置大小相同的整数，如 50
- 3) unicast_src_ip 改成本机内网 IP
- 4) unicast_peer 改成对端机器内网 IP
- 5) virtual_ipaddress 改成内网 VIP
- 6) track_interface 改成本机网卡名 例如 eth0

注意：此文档实践部分演示单播模式，需要指定对端设备的 IP 地址。

! 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 LVS_DEVEL
vrrp_skip_check_adv_addr
vrrp_garp_interval 0
vrrp_gna_interval 0
}

#vrrp_script checkhaproxy
#{
# script "/etc/keepalived/do_sth.sh"
# interval 5
#}

vrrp_instance VI_1 {
#注意主备参数选择
#state MASTER #主 #修改点, 主机为 MASTER , 备机为 BACKUP
state BACKUP #备
interface eth0 #改成本机网卡名 例如 eth0
virtual_router_id 51
nopreempt #非抢占模式
# preempt_delay 10
priority 30 #主高于备, 例如 主 50 , 备 30
```

```
advert_int 1
authentication {
  auth_type PASS
  auth_pass 1111
}
unicast_src_ip 172.16.0.16 #本机内网 IP
unicast_peer {
  172.16.0.14 #对端设备的 IP 地址，例如：10.0.0.1
}
virtual_ipaddress {
  172.16.0.135 #内网 VIP
}

notify_master "/etc/keepalived/notify_action.sh MASTER"
notify_backup "/etc/keepalived/notify_action.sh BACKUP"
notify_fault "/etc/keepalived/notify_action.sh FAULT"
notify_stop "/etc/keepalived/notify_action.sh STOP"
garp_master_delay 1
garp_master_refresh 5

track_interface {
  eth0 #改成本机网卡名 例如 eth0
}

# track_script {
#   checkhaproxy
# }
}
```

步骤 5. 使用 notify_action.sh 进行简单的日志记录

常主常备模式步骤. 修改 notify_action.sh:

1) 无

非常主常备模式步骤. 修改 notify_action.sh:

1) 无

keepalived主要日志仍然记录在/var/log/message

```
#!/bin/bash
#/etc/keepalived/notify_action.sh
log_file=/var/log/keepalived.log
log_write()
{
  echo "`date '+%Y-%m-%d %T'" $1" >> $log_file
}
```

```
[ ! -d /var/keepalived/ ] && mkdir -p /var/keepalived/
```

```
case "$1" in
"MASTER" )
echo -n "$1" > /var/keepalived/state
log_write " notify_master"
echo -n "0" > /var/keepalived/vip_check_failed_count
;;

"BACKUP" )
echo -n "$1" > /var/keepalived/state
log_write " notify_backup"
;;

"FAULT" )
echo -n "$1" > /var/keepalived/state
log_write " notify_fault"
;;

"STOP" )
echo -n "$1" > /var/keepalived/state
log_write " notify_stop"
;;
*)
log_write "notify_action.sh: STATE ERROR!!!"
;;
esac
```

步骤 6.主备云服务器本机主 IP 没有外网IP的场景

- 云服务器或其网卡不需要外网IP

步骤 7. 验证主备切换时 VIP 及外网 IP 是否正常切换

- 启动 keepalived : `/etc/init.d/keepalived start` 或 `systemctl start keepalived` 或 `service keepalived start`
- 验证主备切换容灾效果：通过重启 keepalived 进程、重启子机等方式模拟主机故障，检测 VIP 是否能迁移。/var/log/keepalived.log中同时会留下相应的日志。通过 ping VIP 的方式，可以查看网络中断到恢复的时间间隔。

说明：

- 1) 每切换一次，ping中断的时间大致为4秒。如果是常主常备模式，有可能达到6秒，这种情况通常发生在主的“故障”时间**极短**时，可能发生两次短时间的主备状态倒换，然后VIP重新落地到刚恢复的旧的主机上。
- 2) 脚本日志将会写到 /var/log/keepalived.log 中。日志会占用您的磁盘空间。您可以自行借助 logrotate 等工具处理日志累积的问题。keepalived 进程的日志仍会写到 /var/log/message 中。

TIPS

1) 每个集群使用VPC内唯一的vrrp router id

在使用vmac模式时，不同集群使用 **相同** 的router id会造成mac地址冲突，可能导致通讯异常。因为：

- 腾讯云私有网络提供了 **组播能力**。
- 腾讯云的组播域是整个VPC，不同子网的网卡如果加入相同的组播组，组内消息都能收到。
- Keepalived的组播模式默认使用固定的组播地址，通过vrrp消息里的id来区分不同的集群。
- 假如不同的集群使用相同的router id，不同MASTER的消息会互相干扰导致某些集群没有MASTER。
- router id还会用来生成vmac设备的mac地址，腾讯云要求VPC内网卡设备的mac地址唯一。
- 所以在使用vmac模式时，不同集群相同的router id也会造成mac地址冲突，可能导致通讯异常。

2) 如何使用组播方式进行vrrp通讯

- 提工单申请组播特性。然后自助打开希望使用组播的VPC的组播开关
- keepalived配置文件中 **不配置** unicast_peer。

3) 推荐使用vmac设备

推荐使用vmac模式，原因是：

- 在keepalived运行中如果CVM内发生网络子系统的停启，keepalived可能先于网络子系统把高可用虚拟IP配置到网卡上，使虚拟IP成为网卡的主IP，这样后续的vrrp报文会使用虚拟IP作为源IP来发送，无论是单播模式或是组播模式。其他CVM内的keepalived进程会忽略这种宣告报文，造成脑裂。

4) 控制单个网卡上配置的VIP数量

- 为了虚拟IP的切换更顺畅，腾讯云平台对单个网卡发送免费ARP宣告虚拟IP的频率会进行一定限制。
- 建议目前在单个网卡绑定的高可用虚拟IP数量不超过5个，否则其中一部分IP的切换延迟可能较大。
- 如果需要使用多个虚拟IP，建议在keepalived配置文件的global_defs段落添加或修改配置

```
vrrp_garp_master_repeat 1
```

5) 多网卡情况下如何使用

- 多网卡的CVM推荐使用单播模式配置keepalived。
- 假如使用组播模式，推荐使用默认路由所在的网卡，否则vrrp组播报文在其他网卡上的收发可能存在异常。