Virtual Private Cloud

Best Practise

Product Documentation
Copyright Notice

© 2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud’s products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.
Contents

Best Practise

Configure public gateway
Building High Availability Master/Slave Cluster in VPC Through Keepalived
Creating a High-availability Database by Using HAVIP + Windows Server Failover Cluster
Use keepalived to Establish Master/Slave Clusters
Scenario

If some of your CVMs in Tencent Cloud VPC do not have common public IP but need to access the Internet, you can use a CVM with a public IP (common or elastic public IP) as the public gateway to enable them to access the Internet. The public gateway CVM translates the source IP of outbound traffic. When any other CVMs access the Internet through the public gateway CVM, the public gateway CVM translates their IPs to the public IP of the public gateway CVM, as shown in the figure below.

Prerequisites

- Log into CVM Console.
- The public gateway CVM and the CVMs that need to access the Internet through the public gateway CVM are located in different subnets because the public gateway CVM can only forward routing requests from different subnets.
- The public gateway CVM must be a Linux CVM. A Windows CVM cannot work as a public gateway.

Directions

(Optional) Step 1: Bind an elastic public IP.
If the CVM that serves as the public gateway already has a public IP address, skip this step.

1. In the left sidebar, click EIP to open the EIP management page.
2. Find the target elastic public IP and choose More > Bind in the Operation column.

3. In the Bind resources pop-up window, select a CVM instance to serve as the public gateway and bind it to the elastic public IP.

Step 2: Configure a route table for the subnet where the gateway is located.

The gateway subnet and a common subnet cannot be associated with the same route table. An independent route table must be created for the gateway and be associated with the gateway subnet.

1. Create a custom route table.
2. Associate the route table with the subnet where the public gateway CVM is located as prompted.

**Step 3: Configure a route table for a common subnet.**

Configure the route table for the common subnet and direct the default route to the public gateway CVM, so that CVMs in the common subnet can access the Internet through the routing capability of the public gateway.

In the route table for the common subnet, add the following routing policy:

- **Destination**: public IP to be accessed.
- **Next-hop type**: CVM.
Next hop: private IP of the CVM instance to which the elastic public IP is bound in step 1.

Step 4: Configure the public gateway.

1. Log in to the public gateway CVM, enable network forwarding and NAT proxy, and optimize related parameters.

   i. Run the following command to create the `vpcGateway.sh` script in the `/usr/local/sbin` directory:

```
vim /usr/local/sbin/vpcGateway.sh
```

   ii. Press `i` to switch to the edit mode and write the following code to the script:

```
#!/bin/bash

echo "----------------------------------------------------"
echo "$(date)"
echo "(1)ip_forward config......"
file=/etc/sysctl.conf
grep -i "^net.ipv4.ip_forward.*" $file &>/dev/null && sed -i '/net.ipv4.ip_forward.*/net.ipv4.ip_forward = 1/' $file || echo "net.ipv4.ip_forward = 1" >> $file
1 >/proc/sys/net/ipv4/ip_forward
[ `cat /proc/sys/net/ipv4/ip_forward` -eq 1 ] && echo "-->ip_forward:Success" || echo "-->ip_forward:Fail"

echo "(2)Iptables set......"
iptables -t nat -A POSTROUTING -j MASQUERADE && echo "-->nat:Success" || echo "-->nat:Fail"
iptables -t mangle -A POSTROUTING -p tcp -j TCPOPTSTRIP --strip-options timestamp &
```

```
### iii. Press Esc, enter :wq, save the file and return. Then, run the following command:

```bash
chmod +x /usr/local/sbin/vpcGateway.sh
```

```bash
echo "/usr/local/sbin/vpcGateway.sh >/tmp/vpcGateway.log 2>&1" >> /etc/rc.local
```

### 2. Set the RPS of the public gateway.

#### i. Run the following command to create the `setrps.sh` script in the `/usr/local/sbin` directory:

```bash
vim /usr/local/sbin/setrps.sh
```

#### ii. Press i to switch to the edit mode and write the following code to the script:

```bash
#!/bin/bash

echo "---------------------------------------------"
date
mask=0
i=0
total_nic_queues=0

get_all_mask() {
    local cpu_nums=$1
    if [ $cpu_nums -gt 32 ]; then
        mask_tail=""
        mask_low32="ffffffff"
        idx=$((cpu_nums / 32))
        cpu_reset=$((cpu_nums - idx * 32))
```
if [ $cpu_reset -eq 0 ]; then
    mask=${mask_low32}
    for ((i = 2; i <= idx; i++)); do
        mask+="$mask ${mask_low32}"
    done
else
    for ((i = 1; i <= idx; i++)); do
        mask+="${mask_tail} ${mask_low32}"
    done
    mask_head_num=$((2 ** cpu_reset - 1))
    mask=$(printf "%x%" $mask_head_num $mask_tail)
fi
else
    mask_num=$((2 ** cpu_nums - 1))
    mask=$(printf "%x" $mask_num)
fi

set_rps() {
    if ! command -v ethtool &>/dev/null; then
        source /etc/profile
    fi

ethtool=$(which ethtool)

cpu_nums=$(cat /proc/cpuinfo | grep processor | wc -l)
if [ $cpu_nums -eq 0 ]; then
    exit 0
fi

mask=$(get_all_mask $cpu_nums)

echo "cpu number:$cpu_nums mask:0x$mask"

ethSet=$(ls -d /sys/class/net/eth*)

for entry in $ethSet; do
    eth=$(basename $entry)
    nic_queues=$(ls -l /sys/class/net/$eth/queues/ | grep rx- | wc -l)
    if ((${nic_queues} == 0)); then
        continue
    fi

cat /proc/interrupts | grep "LiquidIO.*rxtx" &>/dev/null
    if [ $? -ne 0 ]; then
        # not smartnic
        #multi queue don't set rps
        max_combined=$(ethtool -l $eth 2>/dev/null | grep -i "combined" | head -n 1 | awk '{print $2}')
    fi
}
# if ethtool -l $eth goes wrong.
[[ ! "${max_combined}" =~ ^[0-9]+$ ]] && max_combined=1
if [ $(max_combined) -ge $(cpu_nums) ]; then
echo "$eth has equally nic queue as cpu, don't set rps for it..."
continue
fi
else
echo "$eth is smartnic, set rps for it..."
fi

  echo "eth:$eth queues:$nic_queues"
  total_nic_queues=($total_nic_queues + $nic_queues)
  i=0
  while (($i < $nic_queues)); do
    echo "$mask" >/sys/class/net/$eth/queues/rx-$i/rps_cpus
    echo 4096 >/sys/class/net/$eth/queues/rx-$i/rps_flow_cnt
    i=$(($i + 1))
done

  flow_entries=$((total_nic_queues * 4096))
  echo "total_nic_queues:$total_nic_queues flow_entries:$flow_entries"
  echo "$flow_entries" >/proc/sys/net/core/rps_sock_flow_entries
}

set_rps

iii. Press Esc, enter :wq, save the file and return. Then, run the following command:

  chmod +x /usr/local/sbin/set_rps.sh
  echo "/usr/local/sbin/set_rps.sh >/tmp/setRps.log 2>&1" >> /etc/rc.local

3. After completing the preceding configurations, restart the public gateway CVM to apply the configurations. Then, test whether a CVM that has no public IP can access the Internet through the public gateway CVM.
Building High Availability Master/Slave Cluster in VPC Through Keepalived

Last updated: 2020-05-28 10:08:52

This document shows how to build a highly available master/slave cluster in Tencent Cloud VPC with keepalived. The practical part of this document will explain how to comprehensively use the new Tencent Cloud product High Availability Virtual IP (HAVIP) and will provide some suggestions related to the use of group broadcast.

Overview

To provide a clear description of the practical use of keepalived on Tencent Cloud CVMs, this document will:

1. Give a brief introduction to keepalived and explain the different ways it is used on CVMs and physical networks.
2. Explain the steps for establishing the modes of non-standing-master mode and standing master/slave mode.
3. Provide many “keepalived configuration and script files + methods for configuration in different scenarios” to guide the user to put it into practice on CVMs.
4. The practical guide mainly introduces how to configure keepalived VRRP Instances as unicast VRRP messages.

Basic Principle

Typically, the high availability master/slave cluster consists of two servers, with the master server being in active status for a service and the slave server being in standby status for the service. The two servers share the same VIP (virtual IP) which is only valid for the master device if there is no failure. In case of the master server failure, the slave server will take over the VIP to continue providing services. The high availability master/slave mode is widely used in MySQL master/slave
Virtual Private Cloud

switchover and Nginx web access.

![Diagram](image)

Keepalived on CVMs vs. Keepalived on Physical Networks

- In traditional physical networks, the master-slave state can be negotiated using the VRRP protocol of keepalived based on the following principles:
  - The master device periodically sends free-of-charge ARP messages to purge the MAC table or terminal ARP table of the uplink switch, triggering the VIP migration to the master device.
- The keepalived can be deployed in Tencent Cloud VPCs to build a high availability master/slave cluster. The difference between this mode and the deployment in physical networks is:
  - The VIP to be used MUST be an HAVIP that has been applied for from Tencent Cloud.
  - There is a subnet attribute that can only be bound by announcement of a machine under the same subnet.

Procedure Overview

1. Apply for a VIP, which can only be migrated within a subnet (the master and slave servers must be in the same subnet).
2. Install and configure keepalived *(Version 1.2.24 or above)* on master and slave servers, and modify the configuration files.
3. Edit the notify mechanism that uses keepalived and do simple log recording using notify_action.sh.
4. Verify whether or not the VIP is switched normally when the master/slave switchover occurs.

Detailed Steps
Step 1: apply for VIP

For detailed instructions on how to apply for a VIP, see High availability virtual IPs.

Step 2: install keepalived (**Version 1.2.24 or above) on master and slave CVMs.

Take CentOS as an example:

- Using yum installation
  Check whether the “yum list keepalived” version number meets requirements. If so, then go to the next step. If not, then use the method of source package installation with “yum –y install keepalived”.

- Using source package installation

  ```
  tar zxvf keepalived-1.2.24.tar.gz
  cd keepalived-1.2.24
  ./configure --prefix=/
  make; make install
  chmod +x /etc/init.d/keepalived **Prevent occurrence of env: /etc/init.d/keepalived: Permission denied**
  ```

Step 3: confirm master and slave requirements

This document introduces two modes:

- Non-standing master mode: the priorities of the two devices to be chosen as the master device are the same.
- Standing master/slave mode: keep one of the devices as the master as long as no failure occurs.

Compared to the non-standing master mode, this mode increases the number of switchovers between the master and the slave. It is recommended to use the non-standing master mode (non-standing master/slave mode, also called the dual slave mode).

Step 4: modify the configuration  `keepalived.conf`

Modifying the configuration file:

- In the standing master/slave mode, modify `keepalived.conf`, taking the master device as the example:
0) state  The initial role. Enter `MASTER` as the master device, and `BACKUP` as the slave device.
1) interface Change to the ENI name of the local device, such as `eth0`.
2) priority The value of the master is higher than that of the slave, for example, `50` for the master and `30` for the slave.
3) unicast_src_ip Change to the private IP of the local device.
4) unicast_peer Change to the private IP of the peer device.
5) virtual_ipaddress Change to the private VIP.
6) track_interface Change to the ENI name of the local device, such as `eth0`.

- In non-standing master/slave mode, modify keepalived.conf with the same changes made to master and slave devices:

0) state  The initial role. Enter `BACKUP` for both master and slave devices.
1) interface Change to the ENI name of the local device, such as `eth0`.
2) priority Both devices are configured with the same integer, such as `50`.
3) unicast_src_ip Change to the private IP of the local device.
4) unicast_peer Change to the private IP of the peer device.
5) virtual_ipaddress Change to the private VIP.
6) track_interface Change to the ENI name of the local device, such as `eth0`.

The practical guide in this document demonstrates the unicast mode, which requires specifying the IP address of the peer device.

```
! 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
#
```
Step 5: use notify_action.sh for simple log recording

Modify notify_action.sh for the standing-master/slave mode:

1) N/A

Modify notify_action.sh for the non-standing-master/slave mode:
1) N/A
keepalived main log is still recorded in /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")
            log_write " notify_master"
            echo -n "$1" > /var/keepalived/state
            echo -n "0" > /var/keepalived/vip_check_failed_count
            ;;
        "BACKUP")
            log_write " notify_backup"
            echo -n "$1" > /var/keepalived/state
            ;;
        "FAULT")
            log_write " notify_fault"
            echo -n "$1" > /var/keepalived/state
            ;;
        "STOP")
            log_write " notify_stop"
            echo -n "$1" > /var/keepalived/state
            ;;
        *)
            log_write "notify_action.sh: STATE ERROR!!"
            ;;
    esac

Step 6: scenario in which primary IPs of local master and slave CVMs do not have public IPs
The CVM or its ENI does not require a public IP.

Step 7: verify whether the VIP and public IP are switched normally during master/slave switchover.
1. Enable keepalived: `/etc/init.d/keepalived start` or `systemctl start keepalived` or `service keepalived start`.

2. Verify the disaster recovery capability of master/slave switchover: simulate the CVM failure by restarting the keepalived process or restarting the CVM to check whether the VIP can be migrated. The corresponding logs will be written in `/var/log/keepalived.log`. You can also view the interval from the network suspension to recovery by pinging VIP.

   - For each switchover, the time of ping suspension should be about 4 seconds. It may take 6 seconds if it is in standing master/slave mode. This type of situation usually occurs when the time of master cluster “failure” is **very short**, and the master/slave switchover may occur twice for a short time. Then, the VIP will be re-migrated to the old master server that was just recovered.
   - The script log will be written to “/var/log/keepalived.log” and takes up your disk space. You can clear the accumulated logs with logrotate and other tools. The log related to the keepalived process will be written to “/var/log/message”.

**Tips**

**Each cluster uses a unique VRRP router ID in the VPC**

MAC address conflict will be caused by different clusters using the **same** router ID when in VMAC mode, and this could lead to a communications exception. The reasons for this are:

- Tencent Cloud VPC provides **multicast ability**.
- Tencent Cloud’s multicast domain is the entire VPC, and if the ENIs of different subnets are added into the same multicast group, messages within the group can be received.
- The default multicast mode of keepalived is to use a fixed multicast address and identify different clusters using the IDs in the VRRP messages.
  - If different clusters use the same router ID, then the messages of different masters will interfere with each other and lead to some clusters having no master.
- The router ID will be used to generate the MAC addresses of VMAC devices, and Tencent Cloud requires that the MAC addresses of VPC private network card devices be unique.
- Therefore, when using VMAC mode, having different clusters with the same router ID will cause MAC address conflict, leading to communications exceptions.

**How to use multicast for VRRP communications**

- Submit a ticket to apply for the multicast feature, which will automatically turn on the multicast switch of the VPC for which multicast functionality is desired.
• **Do not configure** `unicast_peer` in the keepalived configuration file.

**Recommendation to use VMAC device**

It is recommended that VMAC mode be used for the following reason:

When keepalived is running, if starting and stopping of network subsystems occurs in a CVM, keepalived may configure the HAVIP earlier than the network subsystem to the ENI so that the virtual IP becomes the primary IP of the ENI. Whether unicast or multicast mode is being used, the subsequent VRRP messages will use the VIP as the source IP for delivery. The keepalived processes in other CVMs will ignore this announced message, causing split brain.

**Controlling the number of VIPs configured in an individual ENI**

- To allow virtual IPs to switch more smoothly, the Tencent Cloud platform sends a free-of-charge ARP announcement to each ENI, and the frequency of VIPs will be limited to a certain degree.
- We recommend that the number of HAVIPs bound to a single EIP not exceed 5, or else the switching latency of some IPs could increase.
- To use multiple VIPs, we recommend you add or modify the configuration `vrrp_garp_master_repeat 1` in the keepalived configuration file's `global_defs` section.

**How to use it in case of multiple ENIs**

- For CVMs with multiple ENIs, it is recommended that the unicast mode be used to configure keepalived.
- If multicast mode is used, it is recommended that the ENI on the default route be used, or else the VRRP multicast messages may encounter exceptions when sent and received by other ENIs.
Creating a High-availability Database by Using HAVIP + Windows Server Failover Cluster

Last updated: 2020-02-19 16:36:27

1. Creating HAVIPs
   Log in to the VPC Console, and create an HAVIP.

2. Binding and configuration
   The configuration is the same as that in the traditional mode. The backend server declares and negotiates on the device that will be bound with the created HAVIP. You simply need to specify the virtual IP address in the configuration file as HAVIP.
   In the cluster manager, add the HAVIP that was just created.

3. Verification
   After the configuration is completed, directly switch nodes for testing.
   In normal situations, you will see that the network recovers after a short interruption (no interruption will be noticed at all if the switching is fast enough), and online services will not be affected.
Use keepalived to Establish Master/Slave Clusters

Last updated: 2020-02-26 16:38:39

This document shows how to build a highly available master/slave cluster in Tencent Cloud VPC with keepalived.

Preface

To provide a clear description of the practical use of keepalived on Tencent Cloud CVMs, this document will:

1. Give a brief introduction to keepalived and explain the different ways it is used on CVMs and physical networks.
2. Explain the steps for establishing the modes of non-standing-master mode and standing master/slave mode.
3. Provide many "keepalived configuration and script files + methods for configuration in different scenarios" to guide the user to put it into practice on CVMs.
4. The practical guide mainly introduces how to configure keepalived VRRP Instances as unicast VRRP messages.

Basic principle

Typically, the high availability master/slave cluster consists of two servers, with the master server being in active status for a service and the slave server being in standby status for the service. The two servers share the same VIP (virtual IP) which is only valid for the master device if there is no failure. In case of the master server failure, the slave server will take over the VIP to continue providing services. The high availability master/slave mode is widely used in MySQL master/slave switchover and Nginx web access.

Keepalived on CVMs vs. Keepalived on Physical Networks

- In traditional physical networks, the master-slave state can be negotiated using the VRRP protocol of keepalived based on the following principles:
  The master device periodically sends free-of-charge ARP messages to purge the MAC table or terminal ARP table of the uplink switch, triggering the VIP migration to the master device.
The keepalived can be deployed in Tencent Cloud VPCs to build a high availability master/slave cluster. The difference between this mode and the deployment in physical networks is:

- The VIP used must be a HAVIP applied for by Tencent Cloud.
- There is a subnet attribute that can only be bound by announcement of a machine under the same subnet.

**Procedure Overview**

1. Apply for a VIP, which can only be migrated within a subnet (the master and slave servers must be in the same subnet).
2. Install and configure keepalived (**Version 1.2.24 or above**) on master and slave servers, and modify the configuration files.
3. Edit the notify mechanism that uses keepalived and do simple log recording using notify_action.sh.
4. Verify whether or not the VIP is switched normally when the master/slave switchover occurs.

**Detailed Steps**

**Step 1: Apply for VIP**

Apply for VIP.

**Step 2: Install keepalived (**Version 1.2.24 or above**) on master and slave CVMs.**

Take CentOS as an example:

- **Using yum installation**
  
  Check whether the "yum list keepalived" version number meets requirements. If so, then go to the next step. If not, then use the method of source package installation with "yum-y install keepalived".

- **Using source package installation**

```
tar zxvf keepalived-1.2.24.tar.gz
cd keepalived-1.2.24
./configure --prefix=/
make; make install
chmod +x /etc/init.d/keepalived  **Prevent occurrence of env: /etc/init.d/keepalived: Permission denied**
```
Step 3: Confirm master and slave requirements

This document introduces two modes:

- Non-standing master mode: The priorities of the two devices to be chosen as the master device are the same.
- Standing master/slave mode: Keep one of the devices as the master as long as no failure occurs.

Compared to the non-standing master mode, this mode increases the number of switchovers between the master and the slave. It is recommended to use the non-standing master mode (non-standing master/slave mode, also called the dual slave mode).

Step 4: Modify the configuration keepalived.conf

Modifying the configuration file:

- In the standing master/slave mode, modify keepalived.conf, taking the master device as the example:

  0) state The initial role. Enter MASTER as the master device, and BACKUP as the slave device.
  1) interface Change to the ENI name of the local device, such as eth0.
  2) priority The value of the master is higher than that of the slave, for example, 50 for the master and 30 for the slave.
  3) unicast_src_ip Change to the private IP of the local device.
  4) unicast_peer Change to the private IP of the peer device.
  5) virtual_ipaddress Change to the private VIP.
  6) track_interface Change to the ENI name of the local device, such as eth0.

- In non-standing master/slave mode, modify keepalived.conf with the same changes made to master and slave devices:

  0) state The initial role. Enter BACKUP for both master and slave devices.
  1) interface Change to the ENI name of the local device, such as eth0.
  2) priority Both devices are configured with the same integer, such as 50.
  3) unicast_src_ip Change to the private IP of the local device.
  4) unicast_peer Change to the private IP of the peer device.
  5) virtual_ipaddress Change to the private VIP.
  6) track_interface Change to the ENI name of the local device, such as eth0.
The practical guide in this document demonstrates the unicast mode, which requires specifying the IP address of the peer device.

```plaintext
! Configuration File for keepalived
global_defs {
  notification_email {
    acassen@firewall.loc
    failover@firewall.loc
    sysadmin@firewall.loc
  }
noteification_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
#}
vrhap_instance VI_1 {
  # Select proper parameters for the master and slave servers
  # state MASTER #Master device #Modification item. "MASTER" is for the master device and "BACKUP" is for the slave device.
  state BACKUP #Slave device
  interface eth0 #Change it to the ENI name of the local device, such as eth0.
  virtual_router_id 51
  nopreempt #Non-preemptive mode
  # preempt_delay 10
  priority 30 #The priority of the master should be greater than that of the slave, for example, 50 for the master and 30 for the slave.
  advert_int 1
  authentication {
    auth_type PASS
    auth_pass 1111
  }
  unicast_src_ip 172.16.0.16 #Private IP of local device
  unicast_peer {
    172.16.0.14 #IP address of the peer device, for example: 10.0.0.1
  }
  virtual_ipaddress {
```

©2013-2019 Tencent Cloud. All rights reserved.
172.16.0.135 # The private network 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 # Change it to the ENI name of the local device, such as eth0.
}
# track_script {
# checkhaproxy
#
}

Step 5: Use notify_action.sh for simple log recording

Modify notify_action.sh for the standing-master/slave mode:
1) N/A

Modify notify_action.sh for the non-standing-master/slave mode:
1) N/A
keepalived main log is still recorded in /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

Step 6: Scenario in which primary IPs of local master and slave CVMs do not have public IPs

The CVM or its ENI does not require a public IP.

Step 7: Verify whether the VIP and public IP are switched normally during master/slave switchover.

1. Enable keepalived: /etc/init.d/keepalived start Or systemctl start keepalived Or service keepalived start

2. Verify the disaster recovery capability of master/slave switchover: Simulate the CVM failure by restarting the keepalived process or restarting the CVM to check whether the VIP can be migrated. The corresponding logs will be written in /var/log/keepalived.log. You can also view the interval from the network suspension to recovery by pinging VIP.

- For each switchover, the time of ping suspension should be about 4 seconds. It may take 6 seconds if it is in standing master/slave mode. This type of situation usually occurs when the time of master "failure" is Very short, and the master/slave switchover may occur twice for a short time. Then, the VIP will be re-migrated to the old master server that was just recovered.
- The script log will be written to "/var/log/keepalived.log" and takes up your disk space. You can clear the accumulated logs with logrotate and other tools. The log related to the keepalived process will be written to "/var/log/message".

Tips
Each cluster uses a unique VRRP router ID in the VPC

MAC address conflict will be caused by different clusters using the Same Router ID when in VMAC mode, and this could lead to a communications exception. The reasons for this are:

- Tencent Cloud VPC provides Multicast ability.
- Tencent Cloud's multicast domain is the entire VPC, and if the ENIs of different subnets are added into the same multicast group, messages within the group can be received.
- The default multicast mode of keepalived is to use a fixed multicast address and identify different clusters using the IDs in the VRRP messages.
  - If different clusters use the same router ID, then the messages of different masters will interfere with each other and lead to some clusters having no master.
- The router ID will be used to generate the MAC addresses of VMAC devices, and Tencent Cloud requires that the MAC addresses of VPC private network card devices be unique.
- Therefore, when using VMAC mode, having different clusters with the same router ID will cause MAC address conflict, leading to communications exceptions.

How to use multicast for VRRP communications

- Submit a ticket to apply for the multicast feature, which will automatically turn on the multicast switch of the VPC for which multicast functionality is desired.
- Do not configure unicast_peer In the keepalived configuration file.

Recommendation to use VMAC device

It is recommend that VMAC mode be used for the following reason:

When keepalived is running, if starting and stopping of network subsystems occurs in a CVM, keepalived may configure the HAVIP earlier than the network subsystem to the ENI so that the virtual IP becomes the primary IP of the ENI. Whether unicast or multicast mode is being used, the subsequent VRRP messages will use the VIP as the source IP for delivery. The keepalived processes in other CVMs will ignore this announced message, causing split brain.

Controlling the number of VIPs configured in an individual ENI

- To allow virtual IPs to switch more smoothly, the Tencent Cloud platform sends a free-of-charge ARP announcement to each ENI, and the frequency of VIPs will be limited to a certain degree.
- We recommend that the number of HAVIPs bound to a single EIP not exceed 5, or else the switching latency of some IPs could increase.
- If there is a need to use multiple VIPs, we recommend that the configuration
  `vrrp_garp_master_repeat 1` In the keepalived configuration file's global_defs section be added or modified.
How to use it in the situation of having multiple ENIs

- For CVMs with multiple ENIs, it is recommended that the unicast mode be used to configure keepalived.
- If multicast mode is used, it is recommended that the ENI on the default route be used, or else the VRRP multicast messages may encounter exceptions when sent and received by other ENIs.