Virtual Private Cloud
Best Practice

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Best Practice
Migrating from the Classic Network to VPC
Notes for migration

Last updated: 2024-01-12 15:02:02

The classic network is an earlier cloud network provided by Tencent Cloud. As the user scale and services expand, Virtual Private Cloud (VPC) evolves from the classic network to provide independent, controllable, and more secure networks. As a mainstream cloud network, VPC provides better user experience. This document provides answers to questions that you may have when you migrate resources from the classic network to VPC.

Why do I need to migrate to VPC?

A VPC is a logically isolated network space in Tencent Cloud and has the following advantages:

- It allows you to customize IP ranges, IP addresses, and routing policies.
- It supports more complex scenarios such as ENI, network ACL, and cross-region communication.
- It improves the disaster recovery capability and availability greatly.
- It supports multiple CVM models.

VPC is more suitable for use cases that require custom configurations compared with the classic network. To provide you with better services, Tencent Cloud will fully upgrade the classic network. From **March 31, 2022**, resources cannot be created on the classic network. Services on the classic network are officially discontinued on **December 31, 2022**. These services are replaced with the corresponding VPC services.

Is My Business Running on the Classic Network Affected After March 31, 2022?

Your existing resources running on the classic network are still available until **December 31, 2022**. We recommend you migrate your resources to VPC as soon as possible.

What Is the Impact If I Do Not Migrate Resources to VPC?

Before the classic network is discontinued, your existing services will not be affected if you do not migrate. However, after the classic network is discontinued, related services will become unavailable. Please ensure that you complete the migration from the classic network before **December 31, 2022**.

How Do I Determine Whether I Need to Migrate Resources to VPC?

1. Check whether your account is created before **00:00 June 13, 2017**. If your account is created after 00:00 June 13, 2017, the classic network is not supported for your account and the cloud resources that you purchase are already in VPC. In this case, you do not need to migrate resources to VPC. If your account is created before 00:00 June 13, 2017, go to Step 2.
2. Go to the Tencent Cloud Console, select **Billing** > **My Orders**, and check if you have purchased the following resources:
   Classic network resources include:

- Classic network servers: CVM, GPU Cloud Computing, and FPGA Cloud Computing.
- Classic network database: TencentDB for MySQL, Redis, SQL Server, PostgreSQL, and MongoDB, as well

as TDSQL for MySQL.
- Classic network CLB: classic CLB and application CLB
- Others: Classic Network Cloud File Storage (CFS) and Classic Network Cloud Kafka (CKafka).

> **Reminder**
> Lighthouse instances are not part of the classic network resources; they inherently belong to the Virtual Private Cloud (VPC) and do not require migration.

3. If you have purchased resources, please log in to the corresponding resource console to check if the resource is on the classic network. If you haven't purchased any resources, you can ignore the network migration notifications.
For example, in the Cloud Virtual Machine Console, if the network attribute of the cloud resource is **Classic Network**, you need to perform network migration. If it is **Virtual Private Cloud**, you can ignore the network migration notifications.

> **Reminder**
> If you have numerous resources and it's inconvenient to assess them individually, please submit a ticket for further assistance.

Will the Billing Mode of an Instance Change After the Migration from the Classic Network to VPC?

The billing mode is not changed.

Will the Configuration of an Instance Change After the Migration from the Classic Network to VPC?

- The public IP remains unchanged, and the instance can still be accessed by using the original domain name.
- The Media Access Control (MAC) address remains unchanged, but the private IP will be changed.

> **Reminder**
> If the IP of the instance is within the target VPC IP range, you can keep the private IP unchanged by specifying it as the new IP. Otherwise, the private IP will change.

Will the Services Be Interrupted During the Migration from the Classic Network to VPC?

This depends on the specific Tencent Cloud service that you use:
- During the migration of a CVM instance, the instance must be restarted, which will interrupt your service for a short while. We recommend you migrate the instance during off-peak hours.
- For TencentDB services, your services are not affected as dual-IP accessing is supported during migration.
- CLB doesn't support direct migration. You can rebuild instances with the same configuration and gradually migrate the business traffic.

Can I Migrate an Instance Back to the Classic Network?

No, you cannot migrate an instance back to the classic network after it is migrated to VPC.

Can I Migrate a Classic Network Instance in a Region to a VPC Instance in Another Region?

No, instances can be migrated only to the same region and availability zone.

Is Network Migration Implemented by Tencent Cloud?

No, you need to manually perform the migration. If you have any questions, you can submit a ticket for further assistance.

Migration Solutions

Last updated: 2024-01-12 15:02:19

This document describes how to migrate your resources from the classic network to VPC.

Reminder

- For better understand, please read Notes for Migration before you start.
- Before switching the network, you need to create a VPC in the same region as the classic network instance to be migrated and create a subnet in the same AZ as the instance. For more information, see Creating VPCs.

Tencent Cloud provides the two migration solutions below:

- Single-instance network migration: If you only have one Cloud Virtual Machine or TencentDB instance, or if you can accept migrating instances separately, you can use this solution.
- 1.Multi-instance network migration: This solution is recommended for complex business scenarios, such as those involving CVM, CLB, and cloud databases. It allows for IP migration and requires operation during a maintenance window.

Migrating a single instance

You can migrate different types of instance from the classic network to a VPC. See below for details.

Instance type	Features
Cloud Virtual Machine (CVM)	• Restart instance • The Classic Network IP is immediately changed to the VPC IP without any retention period. • The existing public IP of the CVM remain unchanged after the network switch, and domain access will not be affected.
TencentDB for MySQL TencentDB for SQL Server TencentDB for MariaDB TencentDB for TDSQL TencentDB for Redis TencentDB for MongoDB	The classic network IP and VPC IP will be both available for a period of time as described below: • MySQL: Retained for 24 hours (1 day) by default. Range: 0 to 168 hours (7 days). • SQL Server: Retained for 24 hours (1 day) by default. Range: 0 to 168 hours. • MariaDB: Retained for 24 hours (1 day) • TDSQL: Retained for 24 hours (1 day) • Redis: Options include immediate expiration or release after 1 day/2days/3 days7 days. • MongoDB: For versions below 4.0, you can choose to release the IP immediately, or release it after 1 day/2 days/3 days/7 days. For versions 4.0 and above, only the API method is supported to retain the classic network VIP. Make sure that the VPC IP retains the classic network IP; otherwise, access to the classic network will be disrupted if the IP changes.
TencentDB for PostgreSQL	You can configure up to two networks for each instance, both of which can be used for business access. The IPs of different networks can be the same.

> **Reminder**
> If you want to keep the resource IP addresses unchanged after the network switch, try to create a VPC that covers the classic network IP.
> - Set up a private DNS service and perform domain-based modifications. After migrating to Virtual Private Cloud, you can use Tencent Cloud Private DNS.
> - Access via Public IP.

Migrate multiple instances (retain the IP, service downtime required)

Directions

1. Create a VPC as instructed in Creating VPCs.

> **Reminder**
> The subnet must cover the classic network IP. You can configure the secondary CIDR block to expand the VPC IP range. See Editing IPv4 CIDR.

2. Create a new private network CLB within the VPC (**Note: The newly created CLB is application-based and differs from the classic type. For more details, please refer to Classic Cloud Load Balance Upgrade Announcement**). You can specify the VIP of the private network CLB through the API (see Purchasing Cloud Load Balance Instances) and configure the corresponding listeners and rules.
3. Shut down the related server.
4. Migrate a specific Database IP to VPC network. Refer to the corresponding Database migration method in **Single Instance Network Migration**.
5. Unbind the backend CVM from the CLB.
   - For console operations, see Managing Backend Servers.
   - For API operations, see BatchDeregisterTargets.
6. For public network CLB, you can contact us to retain the VIP and switch the configuration to VPC.
7. Switch the specified IP of the CVM instance to the VPC. See Switching to VPC.
8. Bind private and public CLBs to backend CVMs.
   - For console operations, see Managing Backend Servers.
   - For API operations, see BatchRegisterTargets.
9. Verify the service.

> **Reminder**
> If you have any questions about the solutions or have other specific requirements, please contact us.

Example: Migrating a Public Network CLB

Last updated: 2024-01-12 15:02:28

This document describes how to smoothly migrate your public network CLB service from the classic network to a VPC.

> **Reminder**
> This example is only for reference. In actual migration, please carefully assess the impact and develop the migration plan in advance.

Scenario

Resource configuration of the customer service deployed in the classic network:

- The DNS domain name is resolved to the public CLB's VIP in the classic network.
- The public CLB is bound with two CVMs (CVM 1 and CVM 2) as the backend servers.
- The application services deployed on CVM1 and CVM2 need to access TencentDB for Redis and TencentDB for MySQL instances at the backend.

Requirement: Migrate to VPC without interrupting the service

Migration process

1. Create a VPC
2. Migrate TencentDB services
3. Create CVM instances and deploy applications
4. Create a public CLB and associate it with the CVMs
5. Change the IP of the DNS domain name
6. Release the classic network resources

Migration Directions

1. Refer to Creating a Virtual Private Cloud to set up a VPC network environment.
2. Migrate TencentDB for MySQL and TencentDB for Redis instances to the VPC.

> **Reminder**
> During the migration, the connection with TencentDB instances is not interrupted. To maintain the availability of your service, both the original classic network IP and VPC IP will be available for a certain period after the migration. Please complete the migration progress within this period.

3. Create images of CVM1 and CVM2 in the classic network (see Creating a Custom Image). Create two new CVMs with the two images in the VPC (see Creating Instances via Images). Then check whether the CVMs can access TencentDB instances normally.

> **Reminder**
> If restarting CVM instances during the migration is acceptable, you can directly switch to VPC during off-peak hours. For details, see Migrating to VPC.

4. Refer to Cloud Load Balance Quick Start, create a new public CLB within the VPC, and bind the two newly created CVMs. Make sure to check the health status to avoid service disruption due to abnormal conditions.
5. Resolve the DNS domain name to the public CLB's VIP in the VPC.

> **Reminder**
> If you are using Tencent Cloud DNSPod, see Modifying Resolution Records.

6. Check whether the VPC works well. If yes, release the original public CLB and CVM resources in the classic network to finish the migration.

> **Reminder**
> The original classic network IP of a TencentDB instance are automatically released after expiration.

Example: Configuring Hybrid Access for a Private Network CLB

Last updated: 2024-01-12 15:02:37

This document provides a sample configuration for the scenario that both the VPC and classic network are required during the business migration.

Scenario

Resource configuration of the classic network-based business:

- The CVM client accesses a private network CLB.
- The private network CLB is bound with two CVMs (CVM 1 and CVM 2) as the backend servers.
- Applications deployed in CVM 1 and CVM 2 can access the backend TencentDB for MySQL services.

Requests:

- Migrates resources from the classic network to a VPC
- The VPC-based clients has a priority access to the private network CLB service in the classic network.
- The classic network access remains available for one month after the migration.

Workflow

1. Create a VPC
2. Migrate TencentDB services
3. Configure a terminal connection
4. Create a private network CLB and configure its backend service
5. Configure a Classiclink
6. Release the classic network resources

Migration Directions

1. Create a VPC as instructed in Creating VPCs.
2. Migrate the TencentDB for MySQL services to the VPC as instructed in Network Switch.

> **Reminder**
> During the migration, the TencentDB instance still connects. Both the original classic network IP and VPC IP addresses remain valid after the migration, thus maintaining your service availability.

3. Configure a terminal connection service to allow the CVM client in the VPC to access the public network CLB service in the classic network.

> **Reminder**
> Terminal connections do not support cross-region or cross-account configurations. If you need to establish a terminal connection, please contact us.

4. Create a private network CLB instance and its real server in the VPC, and configure the related services.
5. Configure a Classiclink to allow the classic network-based CVM to access the private network CLB

instance in the VPC. Test whether the VPC provides services normally.

6. After the VPC service is normal and VPC-based CVM starts accessing the private network CLB in the VPC, delete the terminal connection, maintain Classiclink, and release the resources in the classic network.

Best Practices of Security Group Change
Security Group Change Process Overview

Last updated：2024-01-12 15:02:47

A security group is a virtual firewall that features stateful data packet filtering. It is used to configure the traffic throttling at the instance level, such as CVM, Cloud Load Balancer and TencentDB. It is an important means of network security isolation.
To meet business requirements, security groups need to be changed during daily OPS, which may affect the associated instances. This document provides you a recommended process to change the security group to minimize the impact on your business.
Watch video

Security Group Change Process

By following the security group change process, you can not only find issues in time, but also minimize the impact during the change.

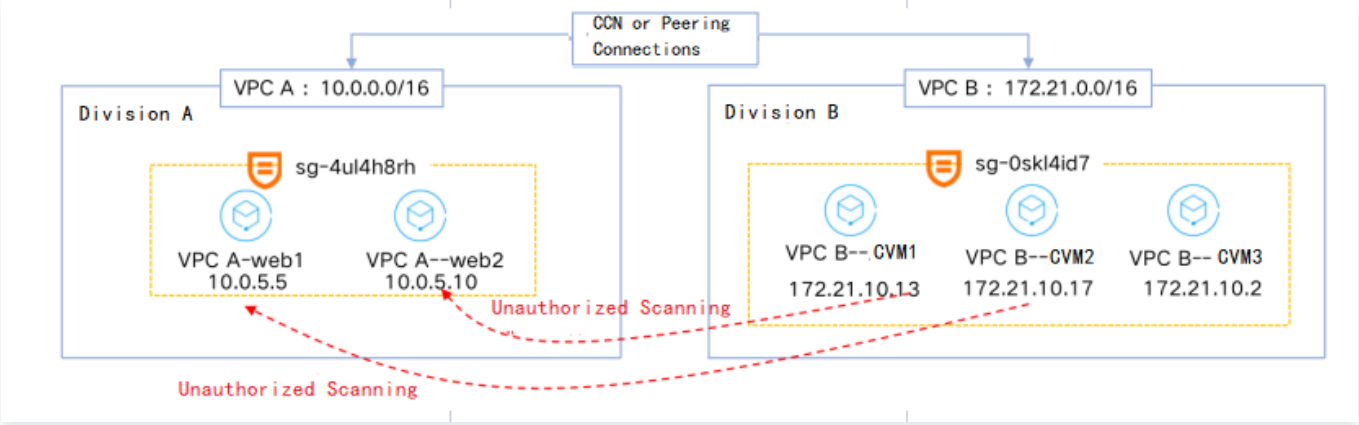
> **Note**
> We recommend that you follow the process. You can also simplify the process if necessary.

Sample of Security Group Change

Last updated: 2024-01-12 15:02:58

Background

- VPC A of Division A and VPC B of Division B were connected through CCN or peering connections. The CVMs in VPC B can access port 80 on the CVMs in VPC A.
- The Security Department of Division A found that the CVM Service1 (`172.21.10.13`) and Service2 (`172.21.10.17`) in VPC B kept scanning closed ports on the CVMs in VPC A.
- The OPS team of Division A decided to modify the security group rule to block access requests from `172.21.10.13` and `172.21.10.17` .

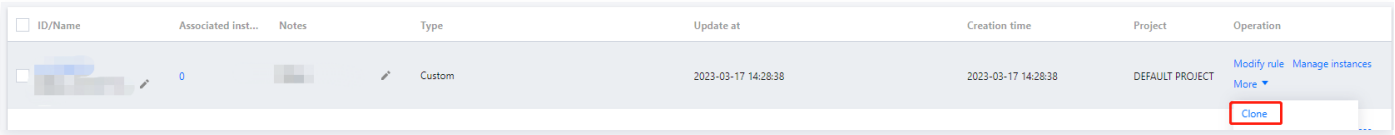


Instructions

> **Note:**
> It is recommended to operate during downtime or off-peak hours of the CVMs.

Step 1. Clone the security group

1. Log in to the VPC console and select **Security** > **Security Groups** in the left sidebar.
2. Locate the security group associated with the attacked instance in Department A, such as sg-4ul4h8rh in this example. Click **More** > **Clone**.



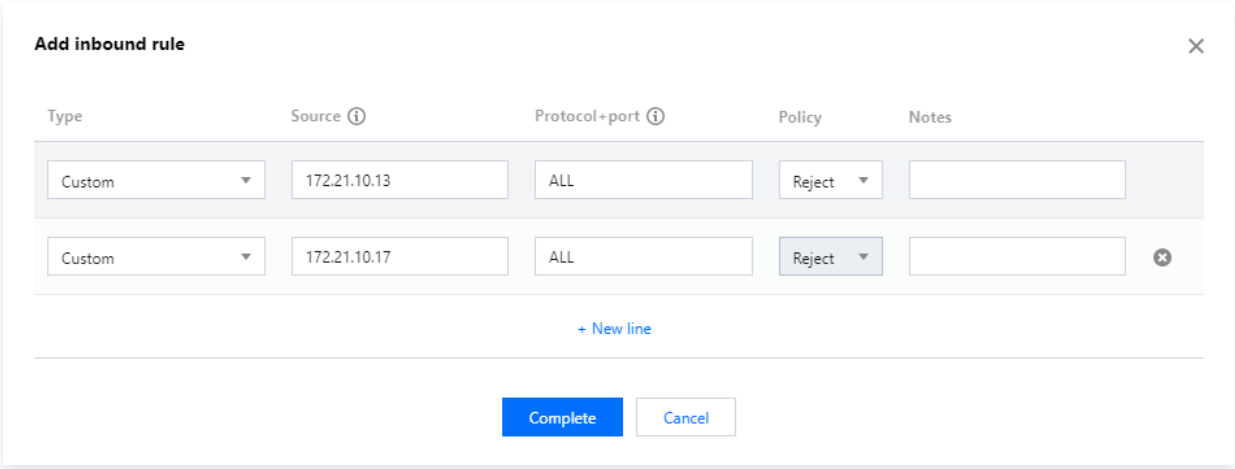
3. Enter a new name for the cloned security group and click **OK**.

Step 2. Modify the cloned security group

> **Note:**
> In this sample, the security group is associated with multiple instances. Upon assessment, web1 in VPC A can be used as a test instance, to which the cloned security group will be bound after the modification.

1. Click **Clone security group ID**.
2. On the Inbound Rules tab, click **Add Rule**.
3. In the pop-up window, select "Reject" for source IP addresses (in this sample, `172.21.10.13` and `172.21.10.17`). Click **OK**.

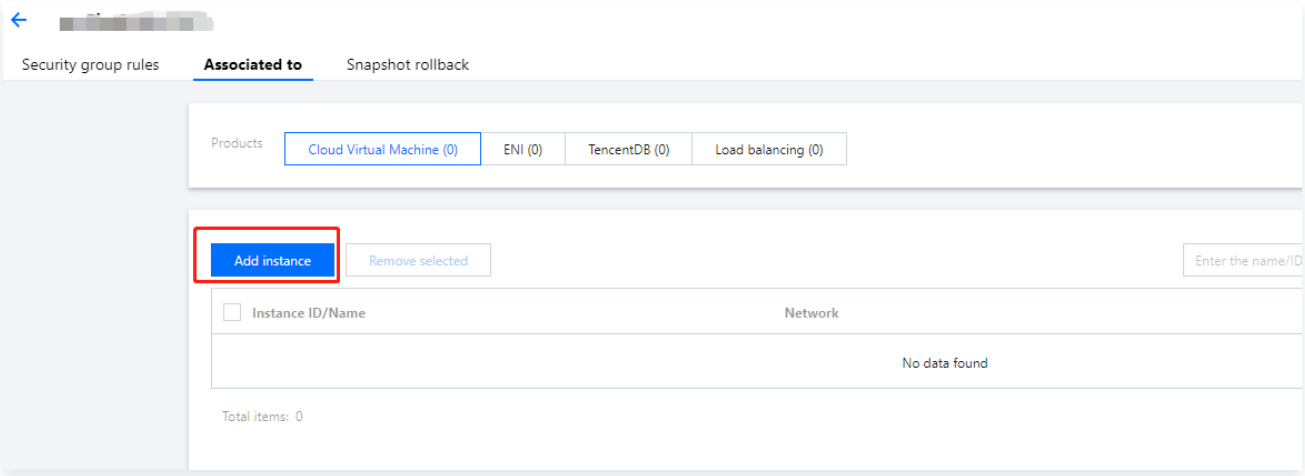
> **Note:**
> The common mistakes here are as follows:
> - **Incorrect policy action**: To block requests from the specific IPs, the action should be **Reject**.
> - **IP range too large**: In this example, we only need to block two CVMs (Service1 and Service2) in the VPC B, while the CVM Service3 3 still needs to communicate with VPC A. If the "Source" in the inbound rule is set to the IP range " `172.21.10.0/24` ", it will also block Service3 in VPC B from accessing CVMs in VPC A, which is not the expected outcome. Therefore, if there are only a few instances involved, try to specify specific IP addresses. If there are many instances, specify the IP range as accurately as possible to minimize unnecessary impact.



Step 3. Bind the modified cloned security group to the test instance

> **Note:**
> When multiple security groups are bound, they will be matched from top to bottom. In this sample, we move the cloned security group to the top to ensure that it is matched first.

1. Click **Manage Instances** in the operation column to enter the **Associated Instances** page.
2. Click **Add instance**, and select the desired instance. In this sample, it is the test instance web1.



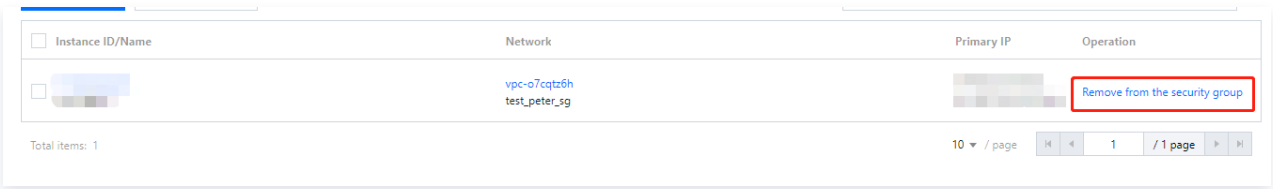
3. Click **OK**.
4. Check whether the test instance continues to run properly.
   - If yes, proceed to the next step.
   - If no, roll back and backtrack the issue. If the issue is identified in the change process, assess whether to continue modifying the cloned security group. If yes, repeat Step 2. If no, end the process.

> **Note:**
> It is observed that the test instance runs properly, which means the modification for the cloned security group meets expectation.

Step 4: Unbind the cloned security group from the test instance

> **Note:**
> - Unbind the cloned security group from the test instance and modify the original security group.
> - Note that if only one security group is associated with an instance, you cannot unbind the security group.

1. Click **Manage Instances** in the operation column, and enter the **Associated Instances** page.
2. On the right side of the instance that requires unbinding the security group (such as the test instance web1 in this case), click **Remove from Security Group** to complete the unbinding of the security group for that instance.



Step 5. Modify the original security group

1. Click **Original security group ID**. In this sample, it is the ID of the security group bound with web1 and web2 in VPC A.
2. On the Inbound Rules tab, click **Add Rule**.
3. In the **Add Inbound Rule** dialog box, add a **Deny** policy for the attacker's source IPs (e.g., `172.21.10.13` and

`172.21.10.17`). Ensure that the source, policy, and other information in the rule are correct, then click **Finish**.

Step 6. Roll back the change unconditionally after two minutes

> **Note:**
> Unconditional rollback aims to promptly detect any temporary impact on related services during security group changes, allowing for timely response decisions and reducing the impact.

1. Wait about two minutes after the original security group rule is modified, then delete the modification made to the security group in Step 5.
2. Check whether the associated CVMs run properly for at least 30 minutes.

- If yes, proceed to the next step.
- If no, end the change immediately. Check and solve the issues, and initiate the change again.

> **Note:**
> In this sample, all CVMs run properly during the 30 minutes.

Step 7. Modify the original security group again

1. Repeat Step 5: Click the **original security group ID**, in this case, the security group ID bound to web1 and web2 in VPC A, to access the security group rules details page.
2. On the Inbound Rules tab, click **Add Rule**.
3. In the **Add inbound rule** page, add a "Reject" policy for the attacker's source IPs (e.g., `172.21.10.13` and `172.21.10.17`). Ensure that the source, policy, and other information in the rule are correct, then click **Complete**.
4. Check whether the associated CVMs run properly for at least 30 minutes.
   - If all CVMs run properly, it indicates the change is successful.
   - If no, end the change immediately. Check and solve the issues, and initiate the change again.

> **Note:**
> In this sample, all CVMs run properly. The change process is completed.

Configuring CVM Instance as Public Gateway

Last updated: 2024-01-12 15:03:09

Reminder

Using a single CVMCloud Virtual Machine (CVM) as a public gateway poses a single point of failure risk. For production environments, it is recommended to use NAT Gateway.

Tencent Cloud has stopped providing the option for setting a CVM as the public gateway when you purchase the CVM since December 6, 2019. To use a CVM as the public gateway, please follow the instructions in this document.

Scenario

When some of your Cloud Virtual Machines (CVMs) in Tencent Cloud VPC do not have a public IP but need to access the public network, you can use a CVM with a public IP (either a common public IP or an Elastic IP) to access the public network. The public gateway CVM will perform source address translation for outbound traffic. The source IP of all other CVMs accessing the public network through the public gateway CVM will be converted to the public IP address of the public gateway CVM, as shown in the following diagram:

Preparations

- Log in to the CVM console.
- A public gateway CVM instance cannot forward route forwarding requests from its own subnet. It must be in different subnets from the CVM instances that need to access the public network through it.
- Only Linux CVMs can work as public gateways.

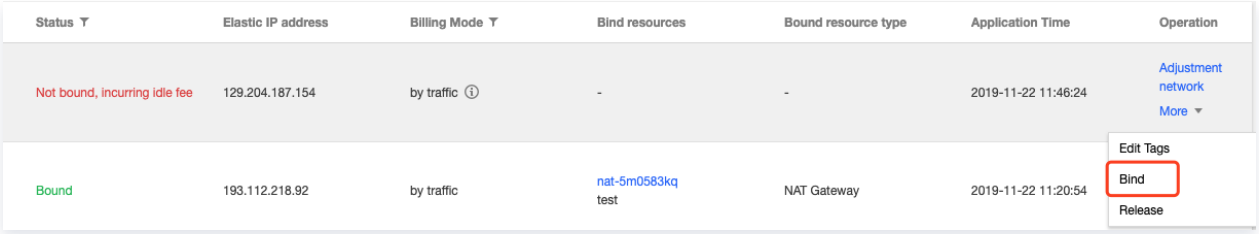
Instructions

(Optional) Step 1. Bind an EIP

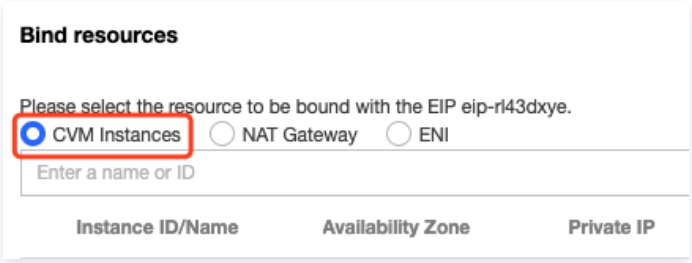
Note

Skip this step if the public gateway CVM already has a public IP address.

1. Log in to the CVM console and select EIP on the left sidebar.
2. Locate the EIP to bind. In the Operation column, select **More** > **Bind**.



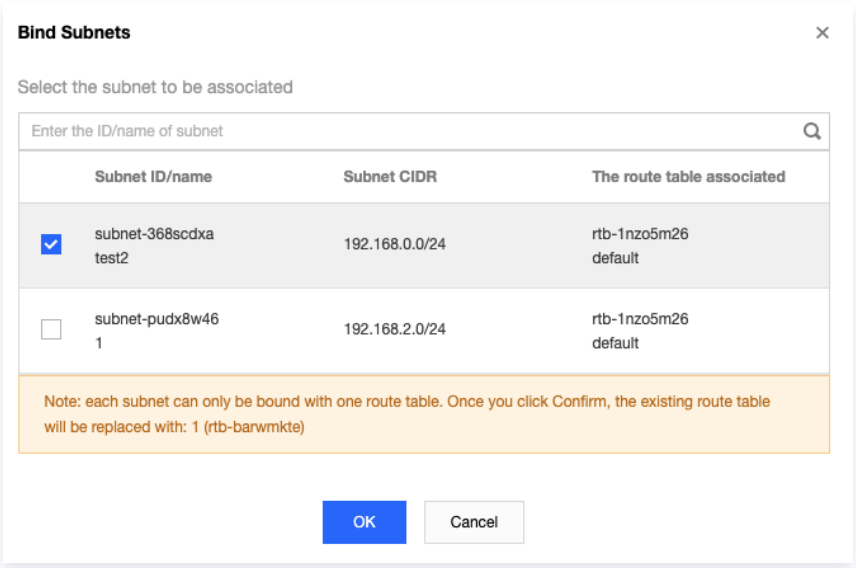
3. In the pop-up window, select a CVM instance to be configured and bind it to the EIP.



Step 2. Configure a route table for the gateway subnet

> **Note**
> The gateway subnet can not share the same route table with other subnets. You need to create a route table dedicated for the gateway subnet and associate them.

1. Create a Custom Route Table.
2. Associate the route table with the subnet where the public gateway CVM resides.



Step 3. Configure a route table for other subnets

Configure a route table for other subnets and a default route through the public gateway CVM instance, so that the CVM instances within these subnets can access the public network through the route forwarding capability of the public gateway.
Add the following routing policies to the route table:

- Destination: The public IP you want to access.
- Next hop type: CVM.
- Next hop: The private IP of the CVM bound with the EIP in Step 1.
  For details, see Configure Route Policy.

Step 4. Configure a public gateway

1. Log in to the public gateway CVM instance and perform the following operations to enable the network forwarding and NAT proxy features:
   1.1 Run the following command to create a new script `vpcGateway.sh` in the `usr/local/sbin` directory.

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

1.2 Press **i** to switch to the edit mode and add 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 \
's/net\.ipv4\.ip_forward.*/net\.ipv4\.ip_forward = 1/' $file || \
echo "net.ipv4.ip_forward = 1" >> $file
echo 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 && \
echo "-->mangle:Success" || echo "-->mangle:Fail"
echo "(3)nf_conntrack config....."
echo 262144 > /sys/module/nf_conntrack/parameters/hashsize
[ cat  /sys/module/nf_conntrack/parameters/hashsize  -eq 262144 ] && \
echo "-->hashsize:Success" || echo "-->hashsize:Fail"
echo 1048576 > /proc/sys/net/netfilter/nf_conntrack_max
[ cat  /proc/sys/net/netfilter/nf_conntrack_max  -eq 1048576 ] && \
echo "-->nf_conntrack_max:Success" || echo "-->nf_conntrack_max:Fail"
echo 10800 >/proc/sys/net/netfilter/nf_conntrack_tcp_timeout_established \
[ cat  /proc/sys/net/netfilter/nf_conntrack_tcp_timeout_established  -eq 10800 ] \
&& echo "-->nf_conntrack_tcp_timeout_established:Success" || \
echo "-->nf_conntrack_tcp_timeout_established:Fail"
```

1.3 Press **Esc** and enter **:wq** to save and close the file.

1.4 Run the following command to set the script permission.

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

2. Set the RPS of the public gateway.

2.1 Run the following command to create a script named `set_rps.sh` in the `usr/local/sbin` directory.

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

2.2 Press **i** to switch to the edit mode and add the following code to the script.

```
# !/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_tail="$mask_tail,$mask_low32"
done
mask_head_num=$((2 ** cpu_reset - 1))
mask=$(printf "%x%s" $mask_head_num $mask_tail)
fi
else
mask_num=$((2 ** cpu_nums - 1))
mask=$(printf "%x" $mask_num)
fi
echo $mask
}
set_rps() {
if ! command -v ethtool &>/dev/null; then
source /etc/profile
fi
ethtool=$(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}'
)
#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-$i/rps_flow_cnt
i=$(($i + 1))
done
done
flow_entries=$((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
```

2.3 Press **Esc** and enter **:wq** to save and close the file.

2.4 Run the following command to set the script permission.

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

3. Restart the public gateway CVM instance to make the configuration take effect. Then, test whether a CVM instance without a public IP can access the public network.

Building HA Primary/Secondary Cluster with HAVIP + Keepalived

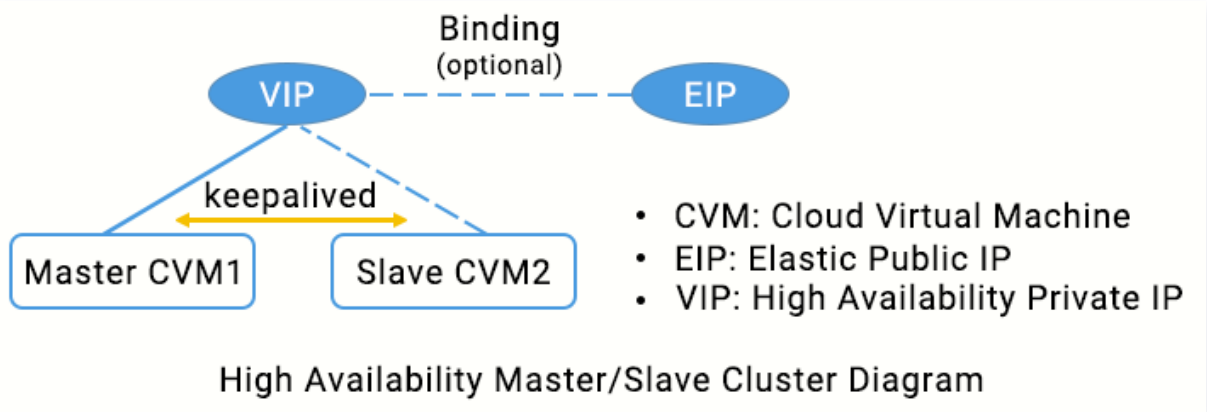
Last updated: 2024-01-12 15:03:19

This document elucidates the process of constructing a high-availability primary and standby cluster within Tencent Cloud VPC, utilizing the keepalived software in conjunction with High Availability Virtual IP (HAVIP).

> **Reminder**
> The HAVIP feature is in beta now, with a switchover latency of around 10 seconds. To try it out, please submit a ticket.

Principles

Typically, a high availability primary/secondary cluster consists of two servers: an active primary server and a standby secondary server. The two servers share the same VIP (virtual IP) which is only valid for the primary server. When the primary server fails, the secondary server will take over the VIP to continue providing services. This mode is widely used in MySQL source/replica switch and Ngnix web access.
Keepalived is a VRRP-based high availability software that can be used to build a high availability primary/secondary cluster among VPC-based CVMs. To use Keepalived, first complete its configuration in the `keepalived.conf` file.



High Availability Master/Slave Cluster Diagram

- In traditional physical networks, the primary/secondary status can be negotiated with Keepalived's VRRP protocol. The primary device periodically sends free-of-charge ARP messages to purge the MAC table or terminal ARP table of the uplink exchange to trigger the VIP migration to the primary device.
- In a Tencent Cloud VPC, a high availability primary/secondary cluster can also be implemented by deploying Keepalived on CVMs, with the following differences:
  - The VIP utilized must be the High Availability Virtual IP (HAVIP) procured from Tencent Cloud.
  - HAVIP is subnet-based and can only be bound to a server under the same subnet.

Supports and Limits

- The Unicast mode is recommended for VRRP communications.

> **Reminder**

> This article demonstrates the Unicast configuration. To use multicast for VRRP communication, you need to join the multicast beta test. Then you can enable VPC multicast as instructed in Enabling and Disabling Multicast. There is no need to configure the IP of the peer device in the keepalived configuration file, which means you **do not need to configure** the "unicast_peer" parameter.

- Keepalived **1.2.24 and later versions** are recommended.
- Ensure that the `garp` parameters have been configured. Because Keepalived relies on ARP messages to update the IP address, these configurations ensure that the primary device always sends ARP messages for the communication.

```
garp_master_delay 1
garp_master_refresh 5
```

- Configure a unique VRRP router ID for each primary/secondary cluster in the VPC.
- Do not use the strict mode. Ensure the "vrrp_strict" configurations have been deleted.
- Control the number of HAVIPs bound to a single ENI to be no more than 5. If you need to use multiple VIPs, add or modify `vrrp_garp_master_repeat 1` in the "global_defs" section of the Keepalived configuration file.
- Adjust the adver_int parameter to keep a balance between network jitter resistance and disaster recovery speed. If `advert_int` is too small, it is susceptible to frequent failovers and temporary **split-brain (dual primary)** situations due to network jitter until the network recovers. If `advert_int` is too large, it will result in slow primary-secondary failover (i.e., longer service downtime) when the primary server fails. **Please assess the impact of split-brain (dual primary) on your business beforehand.**
- Set the `interval` parameter in the specific execution item of `track_script` script (such as `checkhaproxy`) to a larger value, avoiding the `FAULT` status caused by script execution timeout.
- Optional: be aware of increased disk usage due to log printing. This can be solved using logrotate or other tools.

Instructions

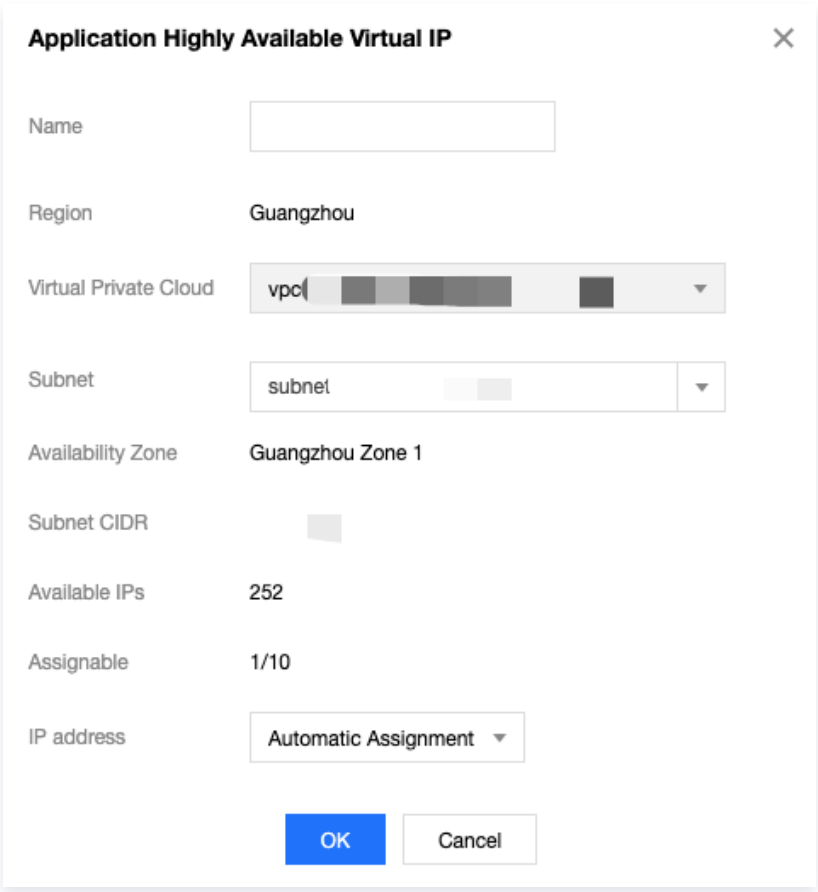
> **Reminder**
> This document uses the following environments as an example. Please replace with your actual configurations.
> - Primary CVM: HAVIP-01, 172.16.16.5
> - Secondary CVM: HAVIP-02, 172.16.16.6
> - HAVIP: 172.16.16.12
> - EIP: 81.71.14.118
> - Image: CentOS 7.6 64-bit

Step 1. Apply for an HAVIP

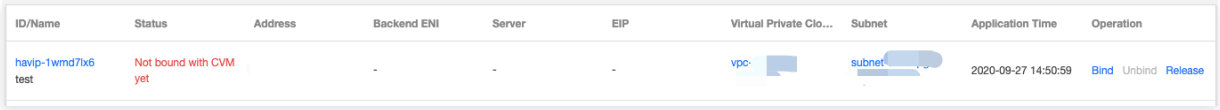
1. Log in to VPC console.
2. In the left sidebar, select **IP & Network Adapters** > **High Availability Virtual IP**.
3. Select the target region on the HAVIP management page and click **Apply**.
4. In the **Apply for HAVIP** window, enter the name, and select the VPC and subnet, and click **OK**.

Reminder

The IP address of the HAVIP can be automatically assigned or manually specified. If you choose to enter an IP address, make sure that the entered private IP address is within the subnet IP range and is not a reserved IP address of the system. For example, if the subnet IP range is `10.0.0.0/24`, the entered private IP address should be within `10.0.0.2 - 10.0.0.254`.



Then you can see the HAVIP you applied for.



Step 2: install Keepalived (version 1.2.24 or later) on primary and secondary CVMs

This document uses CentOS 7.6 as an example to install Keepalived.

1. Run the following command to verify whether the Keepalived version meets the requirements.

```
yum list keepalived
```

- If yes, proceed to Step 2.
- If no, proceed to Step 3.

2. Install the software package using the yum method.

```
yum install -y keepalived
```

3. Install the software package using the source code method.

```
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: configure Keepalived, and bind HAVIP to the primary and secondary CVMs.

1. Log in to the primary CVM HAVIP-01 and run `vim /etc/keepalived/keepalived.conf` to modify its configurations.

> **Reminder**
> In this example, HAVIP-01 and HAVIP-02 are configured with the same weight. Both are in the **BACKUP** status, with a priority of 100. This will reduce the number of switchovers caused by network jitter.

```
  ! 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"   # Check whether the service process runs normally. Replace
“do_sth.sh” with your actual script name. Run it as needed.
      interval 5
  }
  vrrp_instance VI_1 {
  Select proper parameters for the primary and secondary CVMs.
  state BACKUP           #Set the initial status to  Backup
     interface eth0         # The ENI such as  eth0  used to bind a VIP
     virtual_router_id 51       # The virtual_router_id  value for the cluster
     nopreempt             # Non-preempt mode,
     # preempt_delay 10      #Effective only when “state MASTER”
     priority 100          # Configure the same weight for the two devices
     advert_int 5
     authentication {
```

```
        auth_type PASS
        auth_pass 1111
    }
    unicast_src_ip 172.16.16.5  # Set the local private IP address
    unicast_peer {
        172.16.16.6             # IP address of the peer device
    }
    virtual_ipaddress {
        172.16.16.12            # HAVIP
    }
    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    # How long it will take before the ARP cache can be updated after the
CVM switches to the primary status
    garp_master_refresh 5   #Time interval between which the primary node sends ARP messages

    track_interface {
            eth0            # ENI that bound with VIP, such as  eth0
        }
    track_script {
        checkhaproxy
    }
}
```

2. Press "esc" to exit the editing mode, and enter `:wq!` to save and exit.
3. Log in to the secondary CVM HAVIP-02 and run `vim /etc/keepalived/keepalived.conf` to modify its configurations.

```
! 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 {
# Select proper parameters for the primary and secondary CVMs.
```

```
state BACKUP          #Set the initial status to Backup
    interface eth0        # The ENI such as eth0 used to bind a VIP
    virtual_router_id 51      # The virtual_router_id value for the cluster
    nopreempt               #Non-preempt mode
    # preempt_delay 10     #Effective only when “state MASTER”
    priority 100          # Configure the same weight for the two devices
    advert_int 5
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    unicast_src_ip 172.16.16.6   # Private IP of the local device
    unicast_peer {
        172.16.16.5             # IP address of the peer device
    }
    virtual_ipaddress {
        172.16.16.12            # HAVIP
    }
    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    # How long it will take before the ARP cache can be updated after the CVM
switches to the primary status
    garp_master_refresh 5   #Time interval between which the primary node sends ARP messages
    track_interface {
            eth0            # ENI that bound with VIP, such as eth0
        }
    track_script {
      checkhaproxy
    }
}
```

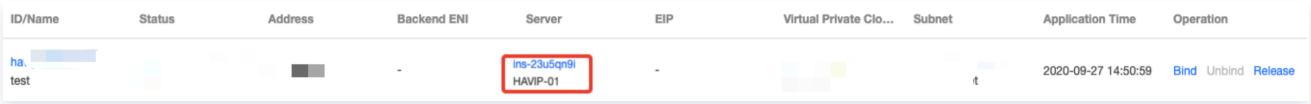
4. Press "esc" to exit the editing mode, and enter `:wq` to save and exit.
5. Restart Keepalived for the configuration to take effect.

```
systemctl restart keepalived
```

6. Check the primary/secondary status of the two CVMs, and confirm that both have HAVIP correctly bound.

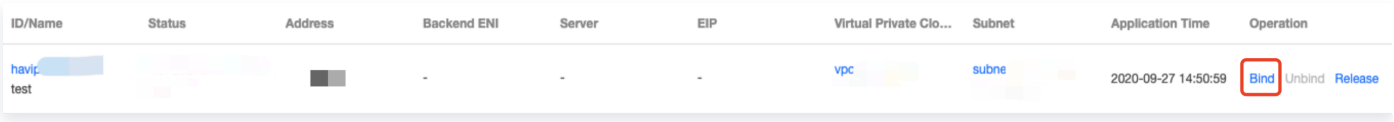
> **Reminder**
> In this example, HAVIP-01 starts the Keepalived first and will normally serve as the primary node.

Log in to the HAVIP console. You will see that HAVIP is bound to the primary CVM HAVIP-01, as shown below.

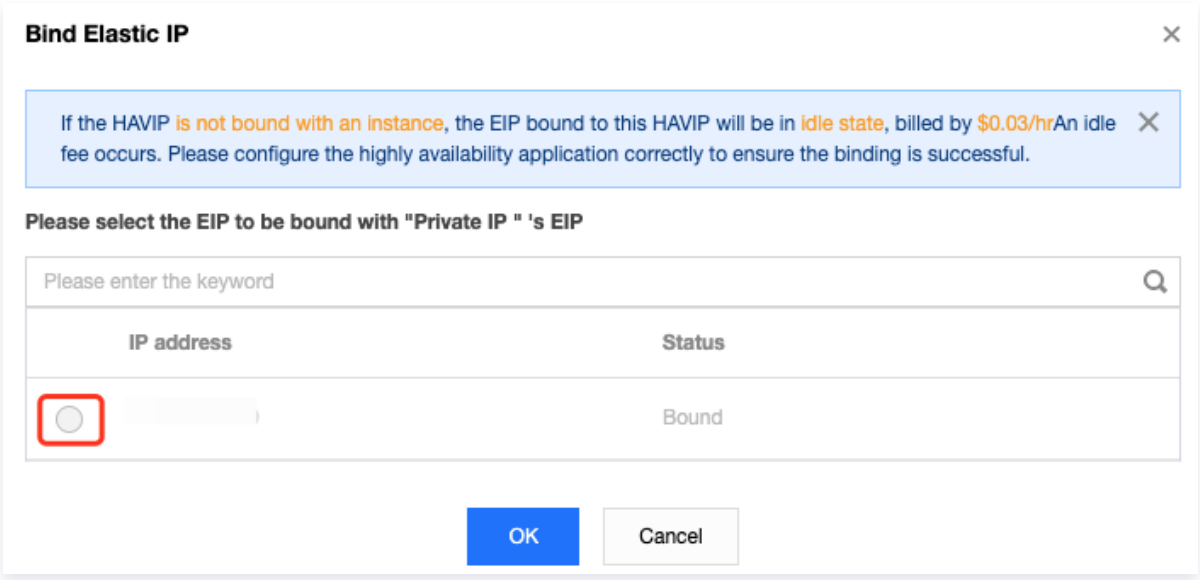


(Optional) Step 4. Bind an EIP with HAVIP

1. In the High Availability Virtual IP console, click **Bind** in the row of the HAVIP applied for in Step 1.



2. In the **Bind EIP** window, select the EIP to be bound and click **OK**. If there are no available EIPs, apply for one in the Elastic Public IP console first.



Step 5: use notify_action.sh for simple logging (optional)

The Keepalived’s main logs are still recorded in “/var/log/message”, and you can add the “notify” script for simple logging.

1. Log in to the CVM and run the `vim /etc/keepalived/notify_action.sh` command to add the following “notify_action.sh” script.

```
#!/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
```

2. Run the `chmod a+x /etc/keepalived/notify_action.sh` command to modify the script permission.

Step 6: verify whether VIP and public IP are switched normally during primary/secondary switch

Simulate the CVM failure by restarting the Keepalived process or restarting the CVM to check whether the VIP can be migrated.

- If the primary/secondary switch succeeds, the secondary CVM will become the server bound with the HAVIP in the console.
- You can also ping a VIP from within the VPC to check the time lapse from network interruption to recovery. Each switch may cause an interruption for about 4 seconds. If you ping the EIP bound to HAVIP over a public network, the result will be the same.
- Run the `ip addr show` command to check whether the HAVIP is bound to the primary ENI.

Using HAVIP and Windows Server Failover Cluster to Build a High Availability DB

Last updated: 2024-01-12 15:03:27

1. **Create a HAVIP**
   Log in to the HAVIP console and create a HAVIP as instructed in Creating an HAVIP.
2. **Binding and Configuration**
   The configuration here is the same as in the traditional mode, where the backend machines declare and negotiate which device binds the created HAVIP. Simply specify the virtual IP as HAVIP in the corresponding configuration file.
   In the cluster manager, add the newly created HAVIP to the configuration.
3. **Verification**
   After the configuration is complete, switch nodes directly for testing.
   Under normal circumstances, you will see a brief interruption followed by network connectivity resuming (if the switch is fast, you may not even notice the interruption), and the business remains unaffected.

CVM Access to Internet Through EIP

Last updated: 2024-01-12 15:03:35

An EIP is a region-level static public IP. It can connect a VPC-based CVM instance to the public network. This document describes how to bind an EIP to a CVM instance for public network access.

Scenario

A VPC-base CVM instance, if you do not allocate a public IP when purchasing it, it cannot access to the public network.
However you can bind an EIP to the CVM instance to access the public network.

Instructions

Step 1. Apply for an EIP

> **Note**
> If you already have an idle EIP, you can skip this step and proceed to Step 2.

1. Log in to the VPC console.
2. Click **IP and ENI** > **Public IP/EIP** to enter the public IP page.
3. At the top of the **Public IP** page, select the same region as the CVM, and then click **Apply**.
4. In the pop-up **Apply for EIP** page, configure the parameters according to your needs and click **OK**. For descriptions of parameters, see Applying for EIP.

Step 2. Bind an EIP with the CVM

1. In the **Public IP** page, select **More** > **Bind** on the right side of the EIP.
2. In the **Bind resource** window, select **CVM instance**, select the CVM ID, and click **OK**.

Step 3. Verify the public network access through the EIP

1. Go to the CVM console, click **Login** on the right of the CVM instance, and enter the password to access the CVM UI.
2. Run `ping www.qq.com` to test the data connectivity. If data is returned, the CVM instance can access the public network.

```
[root@VM-0-13-centos ~]# ping www.qq.com
PING a.https.qq.com (121.51.18.68) 56(84) bytes of data.
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=1 ttl=55 time=3.40 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=2 ttl=55 time=3.42 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=3 ttl=55 time=3.46 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=4 ttl=55 time=3.42 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=5 ttl=55 time=3.43 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=6 ttl=55 time=3.34 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=7 ttl=55 time=3.47 ms
64 bytes from 121.51.18.68 (121.51.18.68): icmp_seq=8 ttl=55 time=3.32 ms
```

Hybrid Cloud Primary/Secondary Communication (DC and VPN)

Last updated: 2024-01-12 15:03:44

If your business is deployed in both a local IDC and a Tencent Cloud VPC, you can connect them via Direct Connect or VPN. To improve the business availability, you set up both DC and VPN connections and configure them as the primary and secondary linkage for redundant communication. This document guides you through how to configure the DC and VPN connection as primary/secondary linkages to connect your IDC to the cloud.

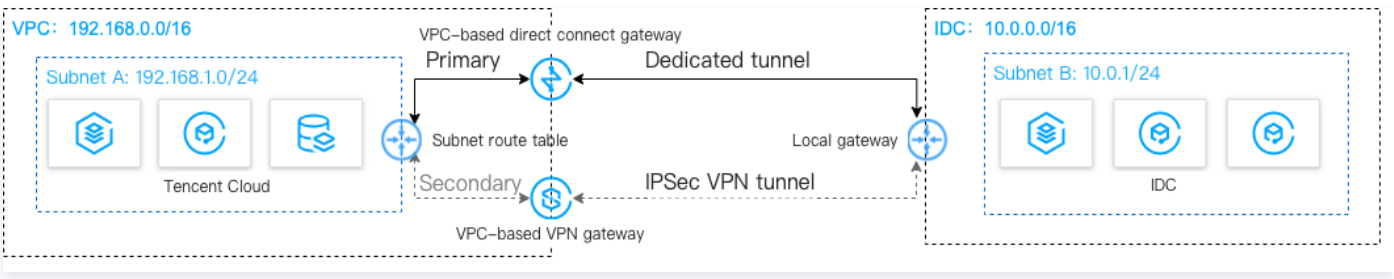
Note

- The route priority feature is in beta testing. To try it out, please contact us.
- The next hop type determines the route priority in the VPC route table. The default route priority sequence from high to low is CCN, direct connect gateway, VPN gateway, and others.
- The route priority can not be changed in the console. To change the priority, please contact us.

Scenarios

Suppose you have deployed your business in both Tencent Cloud VPC and an IDC. To interconnect them, you need to configure network connection services for high-availability communications as follows:

- Direct Connect (primary): Connects the local IDC to a VPC-based direct connect gateway through a connection. When the connection linkage is normal, all data traffic between the IDC and the VPC is forwarded through the connection.
- VPN connection (secondary): Establishes an IPsec VPN tunnel to interconnect the local IDC and the Tencent Cloud VPC. When the connection linkage fails, traffic will be forwarded using this linkage to ensure the business availability.



Preparations

- Your local IDC gateway device supports IPsec VPN and can act as a customer gateway to create a VPN tunnel with the VPN gateway.
- Configure a static IP for the IDC gateway
- Sample data and configuration:

Configuration item			Sample value
Network	VPC information	Subnet CIDR block	192.168.1.0/24

Configuration		Public IP of the VPN gateway	203.xx.xx.82
	IDC	Subnet CIDR block	10.0.1.0/24
		Public IP of the gateway	202.xx.xx.5

Instructions

Step 1. Connect the IDC to VPC via Direct Connect

1. Log in to the Direct Connect console and click **Connections** on the left sidebar to create a connection.
2. Click on **Direct Connect gateway** in the left sidebar, and create a Direct Connect gateway. In this example, we choose to connect to a VPC with a standard Direct Connect gateway. If there are IP range conflicts between the IDC and VPC, you can also choose the NAT type.
3. Click <1>Tunnels<1/> on the left sidebar and create a tunnel. Enter a tunnel name and select the connection type and the direct connect gateway instance that is created. Configure the IP addresses on the Tencent Cloud and IDC sides, select the static route, and enter the IDC IP range. After the configuration is complete, download the configuration guide and complete the IDC device configurations as instructed in the guide.
4. In the route table associated with the VPC subnet for communication, configure a routing policy with the direct connect gateway as the next hop and IDC IP range as the destination.

> **Note**
> For detailed configurations, see Getting Started.

Step 2. Connect IDC to VPC through a VPN connection

1. Log in to the VPN Gateway console and click **Create** to create a VPN gateway. In this example, we select VPC for the Associated network.
2. Click on **Cusotmer gateway** in the left sidebar, and configure the customer gateway (i.e., the logical object of the VPN gateway on the IDC side). Enter the public IP address of the VPN gateway on the IDC side, for example, 202.xx.xx.5.
3. Click **VPN Tunnel** on the left sidebar and then configure the SPD policy, IKE, and IPsec.
4. Configure the same VPN tunnel as the step 3 on the local gateway device of the IDC to ensure a normal connection.
5. In the route table associated with the VPC subnet for communication, configure a routing policy with the VPN gateway as the next hop and IDC IP range as the destination.

> **Note**
> For detailed directions, see Connecting VPC to IDC (Route Table).

Step 3. Configure network probes

> **Note**
> After the first two steps, there are two VPC routes to IDC. That is, both direct connect gateway and VPN gateway act as the next hop. By default, the direct connect gateway route has a higher priority, making it the primary path and the VPN gateway the secondary path.

To stay on top of the primary/secondary connection quality, configure two network probes separately to monitor the key metrics such as latency and packet loss rate and check the availability of primary/secondary routes.

1. Go to the Network Probe page in the VPC console.
2. Click **Create** to create a network probe. Fill in the network probe name, select the VPC, subnet, and probe destination IP. Specify the source-side next-hop route, such as a Direct Connect gateway.
3. Perform Step 2 again, specifying the source next hop route as the VPN gateway. After configuration, you can view the network latency and packet loss rate for the primary and secondary paths of Direct Connect and VPN connections.

> **Note**
> For detailed configurations, see Network Probe.

Step 4. Configure the alarm policy

You can configure an alarm policy for linkage exceptions. When a linkage exception occurs, notifications are sent automatically via emails and SMS messages.

1. Log in to the Alarm Policy Console under Tencent Cloud Observability Platform.
2. Click **Create**. Enter the policy name, select **VPC/Network Probe** for the policy type, specify the network probe instances as the alarm object, and configure trigger conditions, alarm notifications, and other information. Then click **Complete**.

Step 5. Switch between primary and secondary routes

After receiving the exception alarms about the direct connect gateway, you need to manually disable the primary route, and forward traffic to the secondary route VPN gateway.

1. Log in to the VPC console and go to the Route Tables page.
2. Click on the associated route table ID of the subnet to access the route details page. Click  to disable the primary route with the next hop to the Direct Connect gateway. At this point, the VPC traffic to IDC will switch from the Direct Connect gateway to the VPN gateway.

Hybrid Cloud Primary/Secondary Communication (CCN and VPN)

Last updated: 2024-01-12 15:03:55

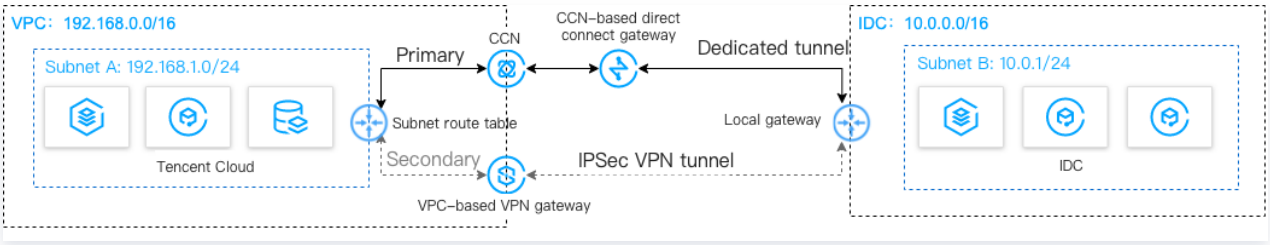
If your business is deployed in both a local IDC and a Tencent Cloud VPC, you can connect them via Cloud Connect Network (CCN) or VPN. To improve the business availability, you set up both CCN and VPN connections and configure them as the primary and secondary linkage for redundant communication. This document guides you through how to configure the CCN and VPN connection as primary/secondary linkages to connect your IDC to the cloud.

> **Note**
> The route priority feature is in beta testing. To try it out, please contact us.

Scenarios

Suppose you have deployed your business in both Tencent Cloud VPC and an IDC. To interconnect them, you need to configure network connection services for high-availability communications as follows:

- CCN (primary): connects the local IDC to a CCN-based direct connect gateway through a physical connection, and adds both the direct connect gateway and the VPC to a CCN to enable interconnection. When the connection linkage is normal, all data traffic between the IDC and the VPC are forwarded over CCN through the physical connection.
- VPN connection (secondary): establishes an IPsec VPN tunnel to interconnect the local IDC and the Tencent Cloud VPC. When the connection linkage fails, traffic will be forwarded using this linkage to ensure the business availability.



Preparations

- + Your local IDC gateway device should support the IPsec VPN feature and can act as a customer gateway to create a VPN tunnel with the VPN gateway.
- + The IDC gateway device has configured with a static IP address.
- + Sample data and configuration:

Configuration item			Sample value
Network Configuration	VPC information	Subnet CIDR block	192.168.1.0/24
		Public IP of the VPN gateway	203.xx.xx.82
	IDC	Subnet CIDR block	10.0.1.0/24

		Public IP of the gateway	202.xx.xx.5

Instructions

Step 1. Connect IDC to VPC through CCN

1. Log in to the Direct Connect console and click **Connections** on the left sidebar to create a connection.
2. Click **Direct Connect Gateway** in the left sidebar to create a Direct Connect gateway, and in this example, choose to connect to the Cloud Connect Network.
3. Click on the CCN-based Direct Connect Gateway ID to enter the details page. In the **IDC Gateway** section, enter the user's IDC IP range, such as 10.0.1.0/24.
4. Log in to the CCN console and click **Create** to create a CCN instance.
5. Log in to the Dedicated Tunnel console and click **Create** to set up a dedicated tunnel connecting to the CCN-based Direct Connect Gateway. Configure the tunnel name, select CCN as the access network, choose the created CCN Direct Connect Gateway, set up the interconnection IPs for both Tencent Cloud and user sides, and select BGP routing as the routing method. After completing the configuration, download the configuration guide and finish the setup on the IDC device.
6. Add the VPC and Direct Connect gateway with the CCN instance to interconnect the VPC and the IDC.

> **Note**
> For detailed directions, see Migrating IDC to the Cloud Through CCN.

Step 2. Connect IDC to VPC through a VPN connection

1. Log in to the VPN Gateway Console and click **Create** to set up a VPN gateway. In this example, select Virtual Private Cloud as the associated network.
2. Click on **Customer Gateway** in the left sidebar to configure the customer gateway (i.e., the logical object of the VPN gateway on the IDC side). Enter the public IP of the VPN gateway on the IDC side, for example, 202.xx.xx.5.
3. Click **VPN Tunnel** on the left sidebar and then configure the SPD policy, IKE, IPsec, and other settings.
4. Configure the same VPN tunnel as the step 3 on the local gateway device of the IDC to ensure a normal connection.
5. In the route table associated with the VPC subnet for communication, configure a routing policy with the VPN gateway as the next hop and IDC IP range as the destination.

> **Note**
> For detailed configurations of VPN gateways in different versions,
> - For a VPN gateway v1.0 and v2.0, see Connecting VPC to IDC (SPD Policy).
> - For a VPN gateway v3.0, see Connecting VPC to IDC (Route Table).

Step 3. Configure network probes

> **Note**
> After the first two steps, there are two VPC routes to IDC. That is, both CCN and VPN gateway act as the next hop. The CCN route has a higher priority, making it the primary path and the VPN gateway the secondary path.

To stay on top of the primary/secondary connection quality, configure two network probes separately to monitor the key metrics such as latency and packet loss rate and check the availability of primary/secondary routes.

1. Go to the Network Probe page on the VPC console.
2. Click **Create** to set up a network probe. Enter the probe name, select the VPC, subnet, and probe destination IP, and specify the next-hop route for the source, such as Cloud Connect Network.
3. Please perform Step 2 again, specifying the source-side next-hop route as the VPN gateway. After configuration, you can view the network latency and packet loss rate for the primary and secondary paths of the CCN and VPN connections.

> **Note**
> For detailed configurations, see Network Probe.

Step 4. Configure the alarm policy

You can configure an alarm policy for linkage exceptions. When a linkage has an exception, notifications are sent automatically via emails and SMS message.

1. Log in to the Alarm Policy Console under Tencent Cloud Observability Platform.
2. Click **Create**. Enter the policy name, select **VPC/Network Probe** for the policy type, specify the network probe instances as the alarm object, and configure trigger conditions, alarm notifications, and other information. Then click **Complete**.

Step 5. Switch between primary and secondary routes

After receiving a CCN network exception alarm, you need to manually disable the primary route, and forward traffic to the secondary route VPN gateway.

1. Log in to the VPC console and go to the Route Tables page.
2. Click the associated route table ID of the VPC communication subnet to enter the route details page. Click  to disable the primary route with CCN as the next hop. At this point, the VPC traffic to IDC will switch from CCN to the VPN gateway.