私有网络

实践教程

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践教程
- 基础网络迁移至私有网络
  - 迁移须知
  - 迁移方案
  - 公网 CLB 网络切换示例
  - 内网 CLB 业务迁移中的混访示例
- 安全组变更最佳实践
  - 安全组变更流程概述
  - 安全组变更示例
- 配置云服务器为公网网关
- 用 HAVIP+Keepalived 搭建高可用主备集群
- 用 HAVIP+Windows Server Failover Cluster 搭建高可用 DB
- 通过 EIP 实现云服务器访问 Internet
- 通过专线接入和 VPN 连接实现混合云主备冗余通信
- 通过云联网和 VPN 连接实现混合云主备冗余通信
- 自动更新安全组规则中放通登录的公网 IP

版权所有：腾讯云计算（北京）有限责任公司

实践教程
基础网络迁移至私有网络
迁移须知

最近更新时间：2024-09-10 11:00:21

基础网络是早期腾讯云云上网络，随着用户规模和更多业务的扩增，在基础网络上演进出具备自主可控、安全性更高的私有网络。私有网络作为主流云网络，能够为您提供更优质的网络体验。本文为您解答网络迁移前可能存在的疑问，以便您更好理解网络迁移操作。

为什么要迁移至私有网络?

私有网络是用户在腾讯云上建立的一块逻辑隔离的网络空间，具有以下优势：

- 可以自由定义网段划分、IP 地址和路由策略。
- 支持弹性网卡/网络 ACL 以及跨地域通信等更加复杂的场景。
- 容灾能力与可用性大幅提升。
- 支持多种 CVM 新机型。

综上所示，私有网络相比基础网络更适合有网络自定义配置需求的场景。且为了给您提供更优质的服务，腾讯云将进行基础网络的全面升级，已于**2022年3月31日**停止全部基础网络产品的创建，基础网络产品整体也将于**2022年12月31日**正式下线，下线后基础网络将不再提供服务，届时将由私有网络 VPC 下对应产品提供服务。

2022年3月31日停止基础网络产品创建后，对运行中的业务是否有影响?

目前暂时无影响，但由于基础网络产品整体将于**2022年12月31日**正式下线，我们建议您尽快完成业务迁移。

不迁移会有什么影响?

基础网络下线前，如不迁移您的存量业务不会受到影响，但基础网络下线后，基础网络相关服务将无法使用，请您务必于**2022年12月31日**前完成基础网络迁移。

如何判断是否需要进行基础网络迁移?

1. 如您是**2017年6月13日零点**后新注册的腾讯云账号，由于新注册的账号已不支持基础网络，那么您所购买的云资源网络属性均为私有网络，您无需关注网络迁移相关通知，如不是请进行下一步判断。
2. 您可进入 腾讯云控制台，进入**费用** > **我的订单**，查看您是否购买过如下资源：
   基础网络包含资源如下：

- 基础网络云服务器：云服务器、GPU 云服务器、FPGA 云服务器。
- 基础网络数据库：云数据库 MySQL、Redis、SQL Server、PostgreSQL、MongoDB、TDSQL MySQL。
- 基础网络负载均衡：传统型负载均衡、应用型负载均衡。
- 其他：基础网络文件存储 CFS、基础网络消息队列 CKafka。

> **说明：**
> 轻量应用服务器不属于基础网络资源，其本身网络属性即为私有网络，无需迁移。

3. 如购买过，请登录对应资源控制台查看该资源是否为基础网络属性。如未购买过，您可忽略网络迁移相关通知。
   以云服务器为例，进入 云服务器控制台 查看，如云资源的网络属性为**基础网络**，则需要进行网络迁移，如为**私有网络**可忽略网络迁移相关通知。

> **说明：**

> 如您资源较多，不方便一一判断，可 提交工单 获得更多帮助。

实例切换至私有网络后，计费是否发生变化？

计费不变。

实例切换至私有网络后，配置是否发生变化？

- 公网 IP 不变，不影响域名访问。
- MAC 地址（Media Access Control Address）不变，但内网 IP 会变。

> **说明：**
> 如果需要迁移的 VPC 中包含实例原有 IP 地址，可以保持内网 IP 不变（迁移时指定新 IP 为实例原 IP），若 VPC 内不包含实例原有 IP，则 IP 会产生变化。

网络迁移过程中，业务会发生中断么？

与具体业务有关：

- 云服务器迁移中需要重启，业务会发生短暂中断，建议选择业务闲时切换。
- 云数据库等产品，因迁移过程中支持双 IP 访问，可确保业务不中断。
- CLB 不支持直接迁移，可重建相同配置的实例，将业务流量逐步迁移。

实例迁移至私有网络后，能否再迁回去？

不能，基础网络切换至私有网络操作不可逆。

基础网络实例在上海，必须迁移至上海地域的私有网络么？

是的，实例切换网络需同地域同可用区。

网络迁移是腾讯云统一处理么？

不是，需要您手动执行迁移操作，若您有任何疑问，可 提交工单 获得更多帮助。

迁移方案

最近更新时间：2022-08-16 11:46:29

本文主要介绍 基础网络 迁移到 私有网络 的网络切换方案。

> **说明**
> - 为帮助您更好的理解本次网络迁移，您可以查看 迁移须知 解答您的疑虑。
> - 网络切换前，需提前创建好与待迁移基础网络实例同地域的私有网络，以及同可用区的子网，具体请参见 创建私有网络 。

目前腾讯云提供两种方案，两种方案可独立使用，也可结合实际业务迁移场景组合使用：
- 单实例网络迁移：如果您仅持有云服务器、云数据库其中一个实例，或您可接受实例单独迁移，可采用此方案。
- 多实例网络迁移：当您的业务比较复杂，例如同时包含 CVM、CLB 及云数据库时，推荐使用该方案，该方案可以保留 IP 迁移，需要在停服窗口操作。

单实例网络迁移

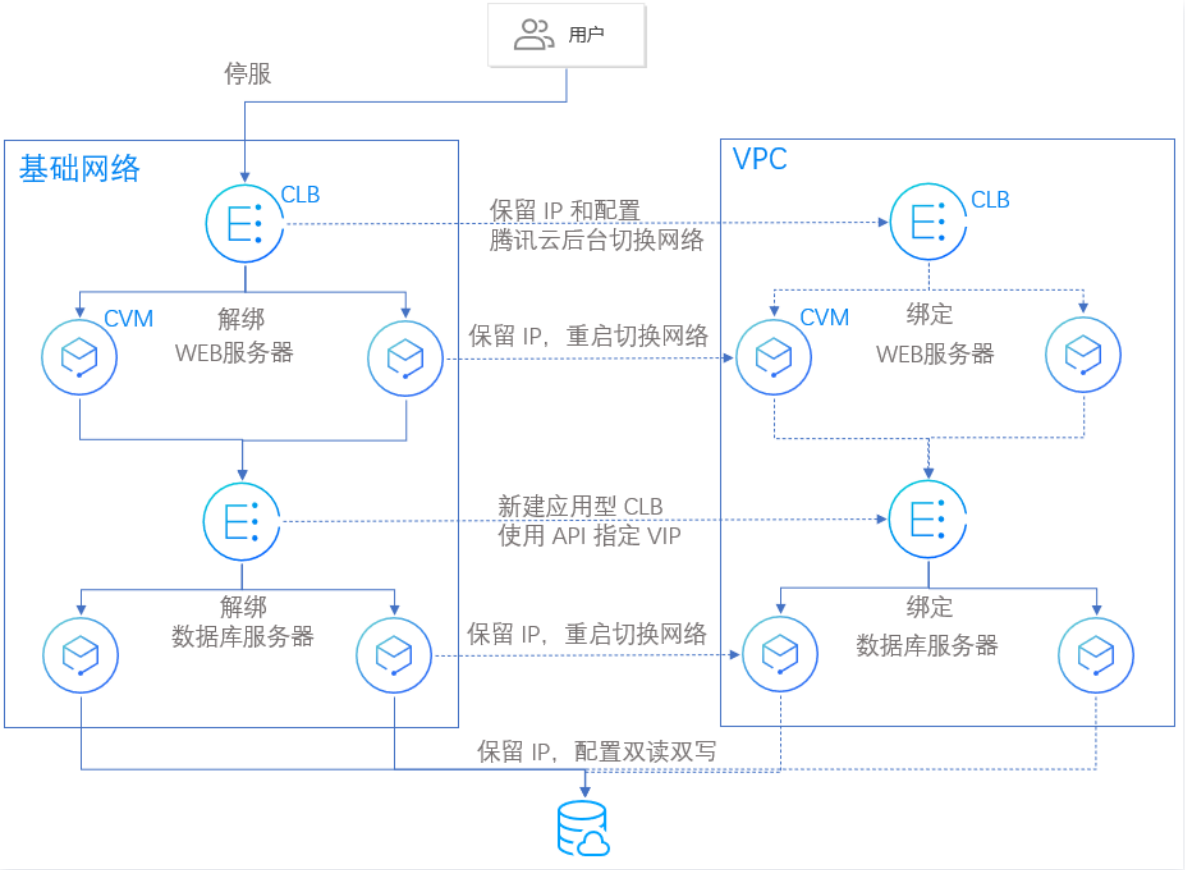
腾讯云支持基础网络内单实例一键迁移至私有网络，详情请单击以下实例链接。

实例	特点
云服务器 CVM	• 需重启实例 • 基础网络 IP 立即变更为私有网络 IP，无保留时间 • 如云服务器有公网 IP，网络切换后公网 IP 地址不变，不会影响域名访问
云数据库 MySQL 云数据库 SQL Server 云数据库 MariaDB 云数据库 TDSQL 云数据库 Redis 云数据库 MongoDB	在一定时间内保持双 IP 访问，原基础网络 IP 保持时间如下： • MySQL：默认保持24小时（1天），最长可以保持168小时（7天） • SQL Server：保持24小时（1天），可自定义回收时长，范围是0-168小时 • MariaDB：保持24小时（1天） • TDSQL：保持24小时（1天） • Redis：可选择立即失效、1天后释放、2天后释放、3天后释放、或7天后释放 • MongoDB：4.0以下版本可选择立即失效、1天后释放、2天后释放、3天后释放、或7天后释放。4.0及以上版本只支持 API 方式保留基础网络 VIP，务必保证 VPC IP 保留基础网络 IP，如果 IP 变化基础网络的访问则会不通
云数据库 PostgreSQL	一个实例最多同时存在两套网络配置，且两套网络可同时进行业务访问，不同网络的 IP 可以相同

> **说明**
> 如需网络切换后云资源 IP 地址不变，可尝试创建包含基础网络 IP 的 VPC。若因客观因素无法实现时，请参考如下方案：
> - 自建内网 DNS 服务并做域名化改造，待迁移至私有网络后，可使用腾讯云 私有域解析 Private DNS 。
> - 使用公网 IP 访问。

多实例网络迁移（保留 IP，需停服）

整体迁移图解



具体迁移步骤

1. 参见 创建私有网络 创建 VPC 网络环境。

> **说明**
> 子网需覆盖基础网络 IP，可使用辅助 CIDR 功能增加 VPC 的网段 编辑 IPv4 CIDR 。

2. 在 VPC 内新建内网 CLB（**注意：当前新建的 CLB 都是应用型与传统型有所差异，详情请参见 传统型负载均衡升级公告**），通过 API 的方式可以指定内网 CLB 的 VIP（可参见 购买负载均衡实例 ）并配置好对应的监听器和规则。
3. 业务停服。
4. 数据库指定 IP 迁移至 VPC 网络，可参考 **单实例网络迁移** 中对应的数据库迁移方法。
5. CLB 解绑后端 CVM。
   - 控制台操作请参见 管理后端服务器 。
   - API 操作请参见 批量解绑四七层后端服务 。
6. 外网 CLB 可联系 腾讯云售后支持 保留 VIP 及配置切换至 VPC。
7. CVM 指定 IP 切换至 VPC 网络，详情请参见 切换私有网络服务 。
8. 内外网 CLB 绑定后端 CVM
   - 控制台操作请 管理后端服务器 。
   - API 操作请参见 批量绑定虚拟主机或弹性网卡 。
9. 业务验证。

> **说明**
> 对方案有疑问或有其他特殊需求可联系 腾讯云售后支持 。

公网 CLB 网络切换示例

最近更新时间：2024-11-05 15:06:32

本文介绍如何将公网 CLB 业务从基础网络平滑迁移至私有网络。

> **说明：**
> 本示例仅供参考，实际迁移的场景可能较示例更复杂，请在迁移前仔细评估影响，谨慎地制定迁移方案。

迁移示例

假设用户基础网络业务使用如下产品：

- DNS 中配置域名解析地址为基础网络公网 CLB VIP。
- 公网 CLB 绑定了两台云服务器 CVM1 和 CVM2 作为后端服务器。
- CVM1 和 CVM2 上部署了应用服务，应用服务会访问后端的云数据库服务 Redis 和 MySQL。

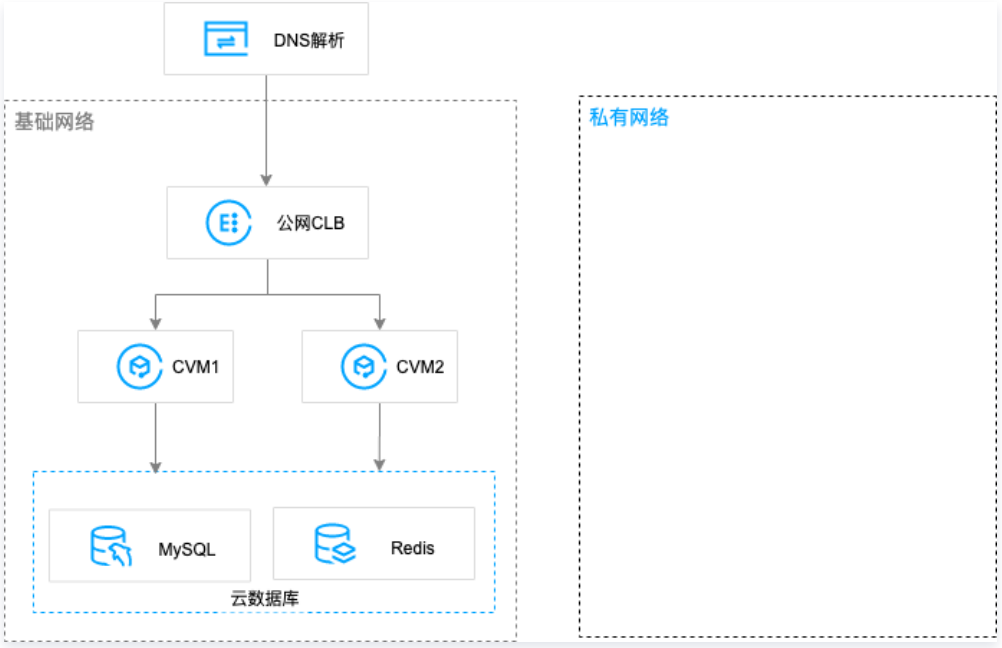
迁移要求：业务平滑切换至私有网络。

迁移流程

1. 准备私有网络环境
2. 切换云数据库网络
3. 新建 CVM 并部署应用
4. 新建公网 CLB 并绑定 CVM
5. 切换 DNS 域名解析 IP
6. 释放基础网络资源

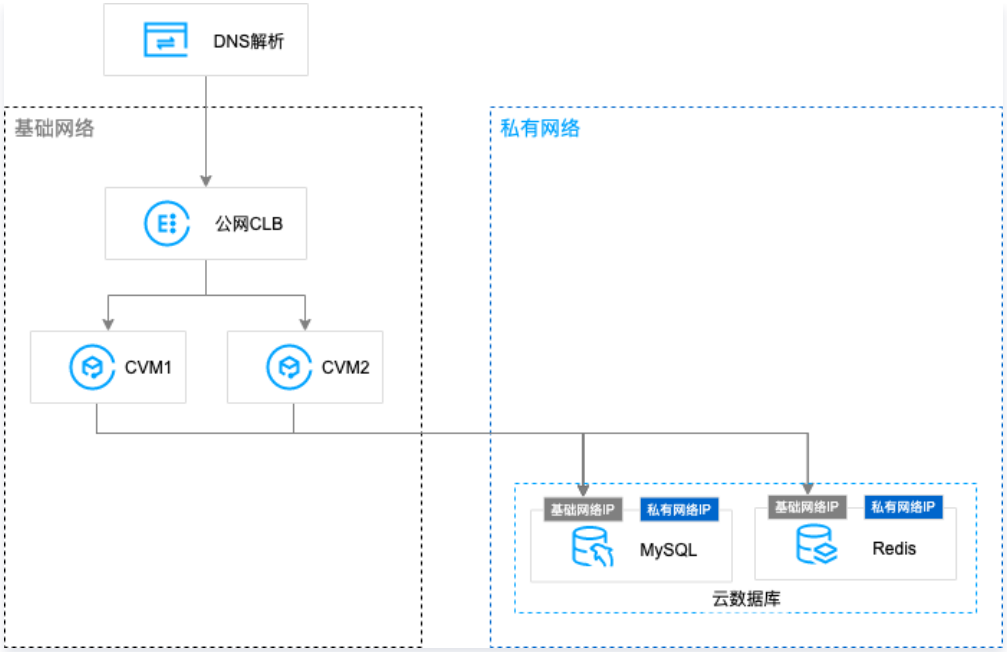
迁移步骤

1. 参见 创建私有网络 创建 VPC 网络环境。



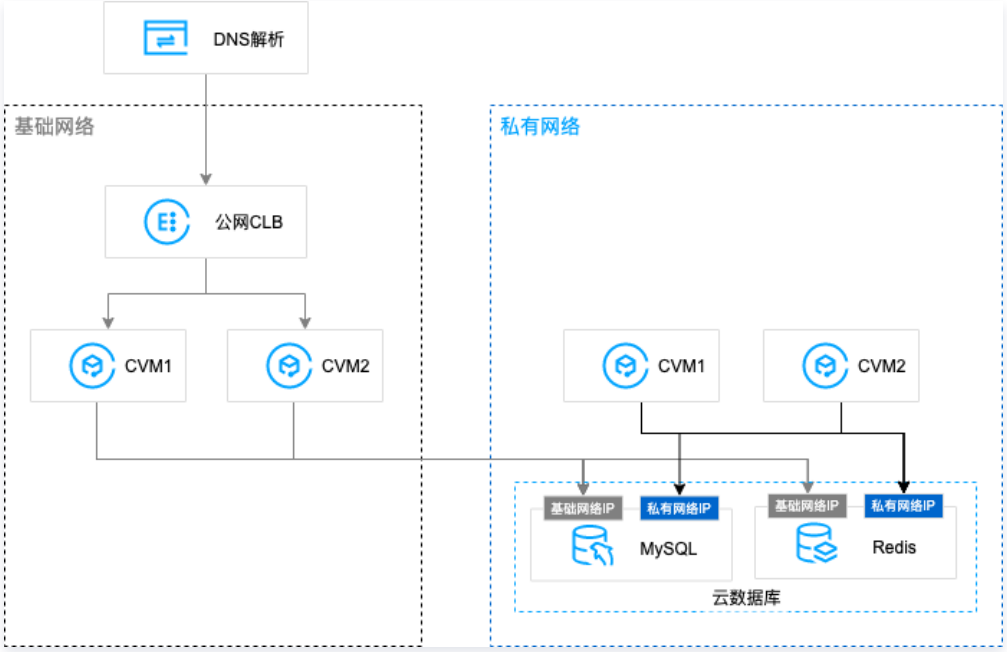
2. 参见 更换 MySQL 的网络 和 更换 Redis 的网络 切换数据库网络。

> **说明：**
> 云数据库在网络切换时连接不中断，且切换后同时保持原基础网络和私有网络 IP，可确保迁移过程中不停服。在原有基础网络访问的最长保持时间内，请完成其他产品的迁移。

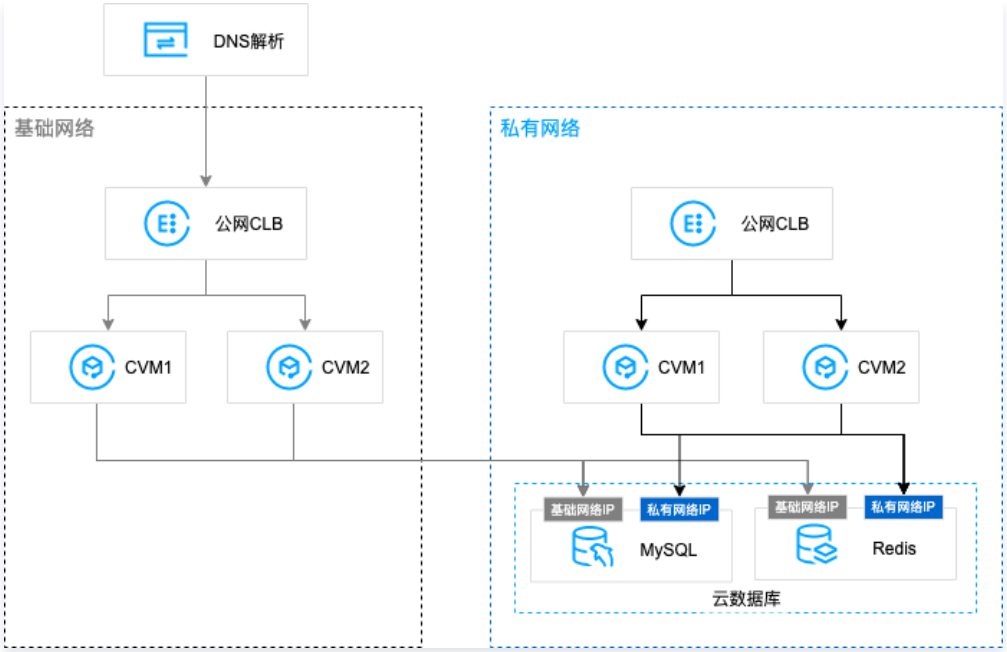


3. 参见 创建自定义镜像 在基础网络制作两台云服务器 CVM1 和 CVM2 的镜像，在私有网络内，参见 通过镜像创建实例 创建两个 CVM。完成后，测试 CVM 是否能正常访问云数据库。

> **说明：**
> 如您可以接受 CVM 切换时实例重启导致的业务停服，也可以选择在业务低峰时段直接切换 CVM 网络，具体请参见 云服务器切换私有网络服务 。

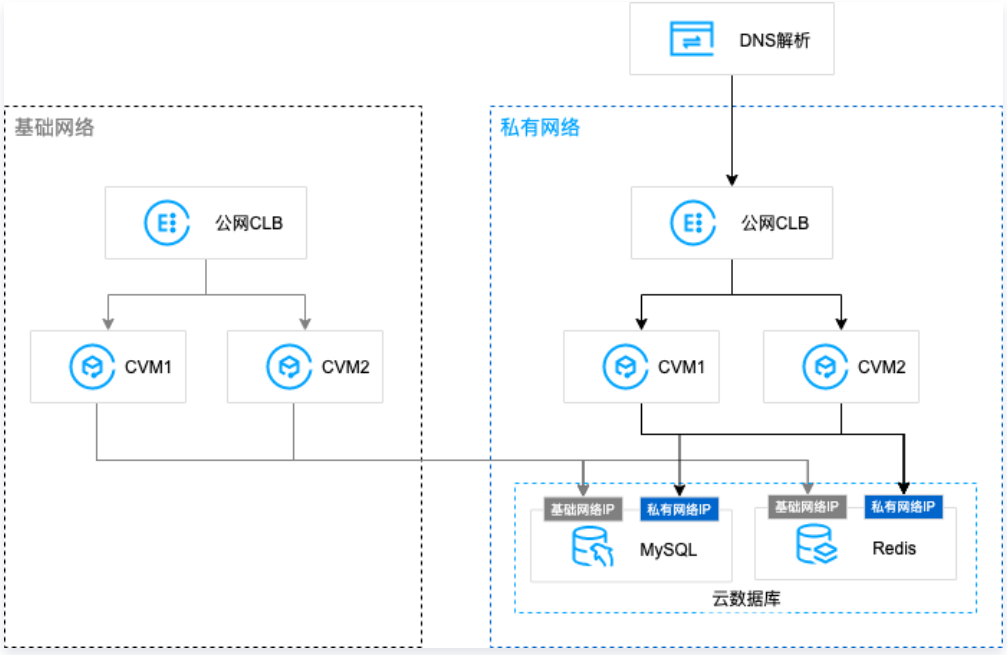


4. 参见 负载均衡快速入门，在私有网络内，新建一个公网 CLB，并绑定上述新建的两个 CVM，注意检查健康状态，避免因异常情况影响服务。



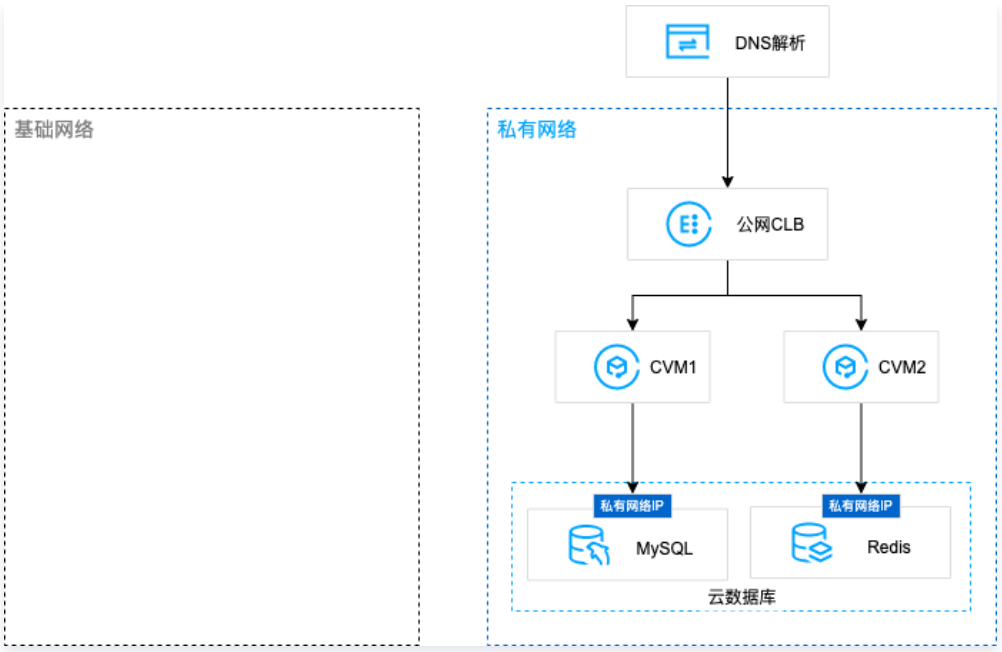
5. 切换 DNS 域名解析地址为 VPC 中公网 CLB 的 VIP。

> **说明：**
> 如您使用的是腾讯云 DNSPod，请参考 修改解析记录 。



6. 待私有网络运行正常后，释放基础网络下留存的公网 CLB、CVM 资源，结束迁移。

> **说明：**
> 云数据库的原基础网络 IP 过期后将自动释放。


DNS解析
基础网络
私有网络
公网CLB
CVM1
CVM2
私有网络IP
MySQL
私有网络IP
Redis
云数据库

内网 CLB 业务迁移中的混访示例

最近更新时间：2024-01-25 15:38:01

本文介绍在业务迁移过程中，有混访需求时的配置示例。

混访示例

假设用户基础网络业务使用如下产品：

- CVM 客户端访问内网 CLB。
- 内网 CLB 绑定了两台云服务器 CVM1 和 CVM2 作为后端服务器。
- CVM1 和 CVM2 上部署了应用服务，应用服务会访问后端的云数据库服务 MySQL。

迁移过程中的业务混访要求：

- 将基础网络的服务资源迁移至私有网络 VPC。
- 要求 VPC 客户端可以优先访问基础网络内网 CLB 服务。
- 基础网络客户端在网络切换后还能继续访问业务一个月。

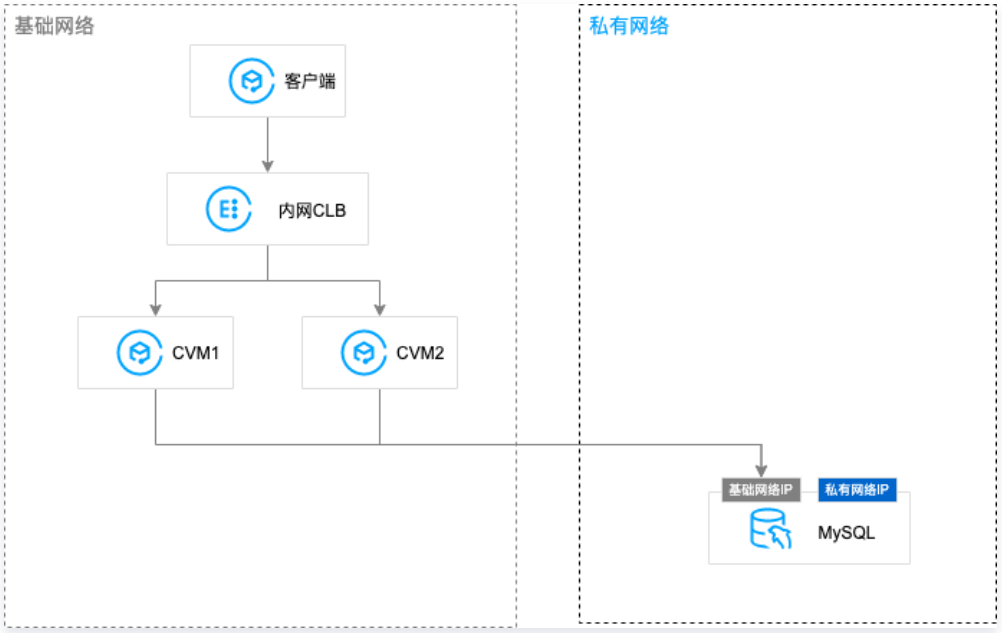
操作流程

1. 准备 VPC 环境
2. 切换云数据库网络
3. 配置终端连接
4. 创建内网 CLB 并配置后端服务
5. 配置基础网络互通
6. 释放基础网络资源

迁移步骤

1. 参见 创建私有网络 创建 VPC 网络。
2. 参见 更换 MySQL 的网络 切换云数据库网络。

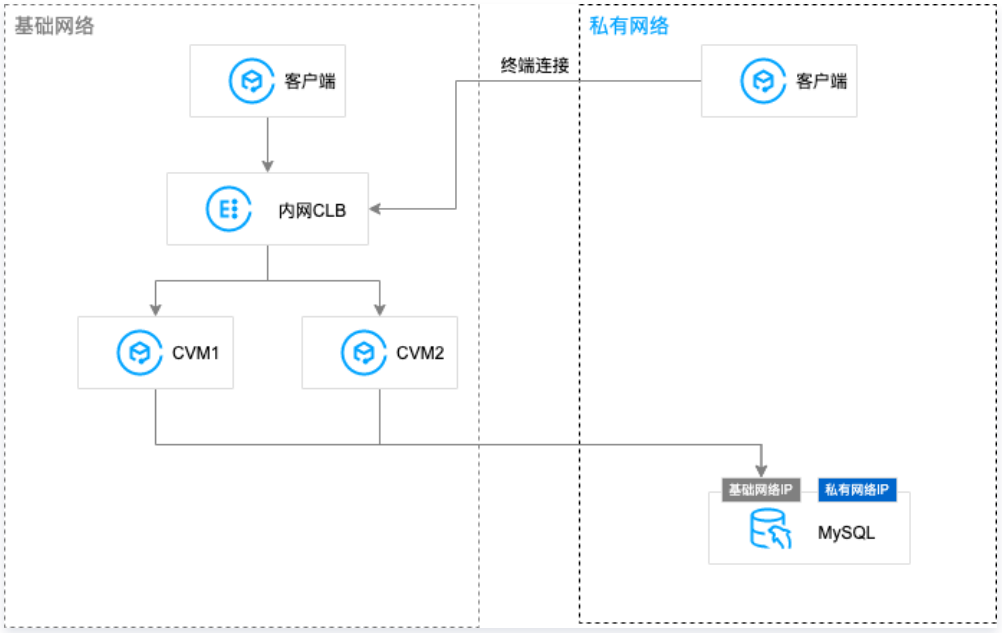
> **说明**
> 云数据库在网络切换时连接不中断，且切换后同时保持原基础网络和 VPC IP，可确保迁移过程中不停服。



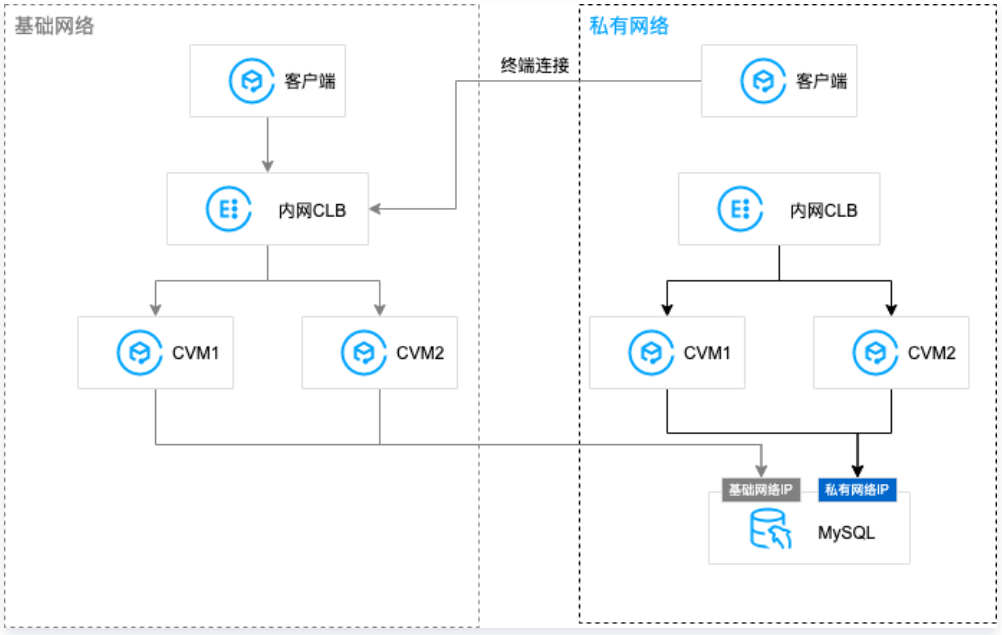
3. 配置终端连接服务，使得 VPC 内 CVM 客户端可以访问基础网络内网 CLB 服务。

> **说明**

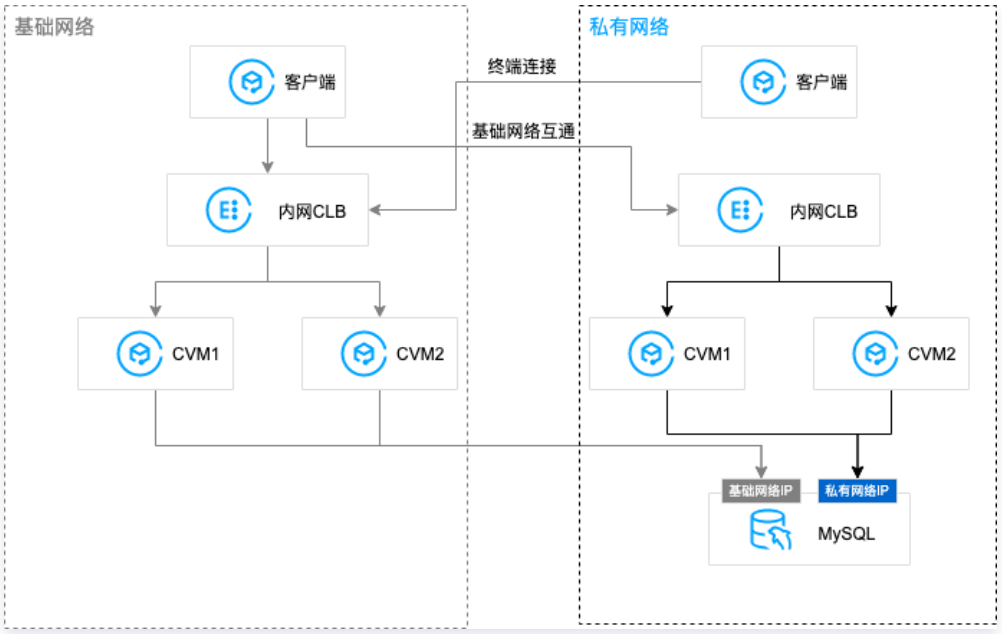
> 终端连接不支持跨地域、跨账号，如您有建立终端连接的需要，请 在线咨询 。



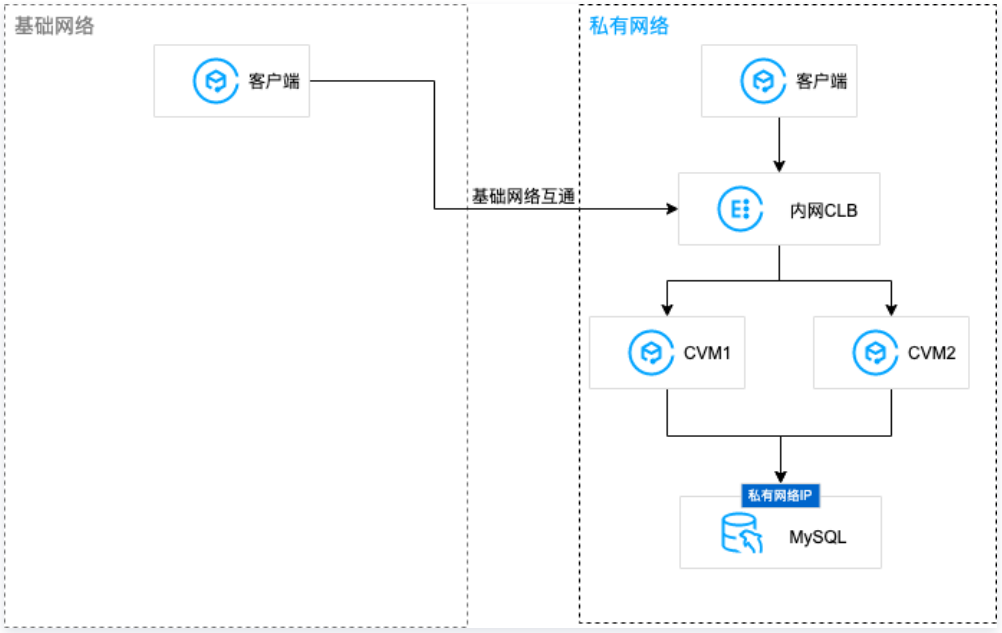
4. 在 VPC 内创建内网 CLB 以及后端 CVM，并配置相关业务。



5. 配置基础网络互通，使得基础网络内 CVM 客户端可以访问 VPC 内网 CLB，并验证 VPC 业务提供是否正常。



6. 待 VPC 业务验证正常后， VPC CVM 客户端开始访问 VPC 的内网 CLB 业务，删除终端连接，继续保持基础网络互通，释放基础网络服务资源。



安全组变更最佳实践
安全组变更流程概述

最近更新时间：2024-11-05 15:06:32

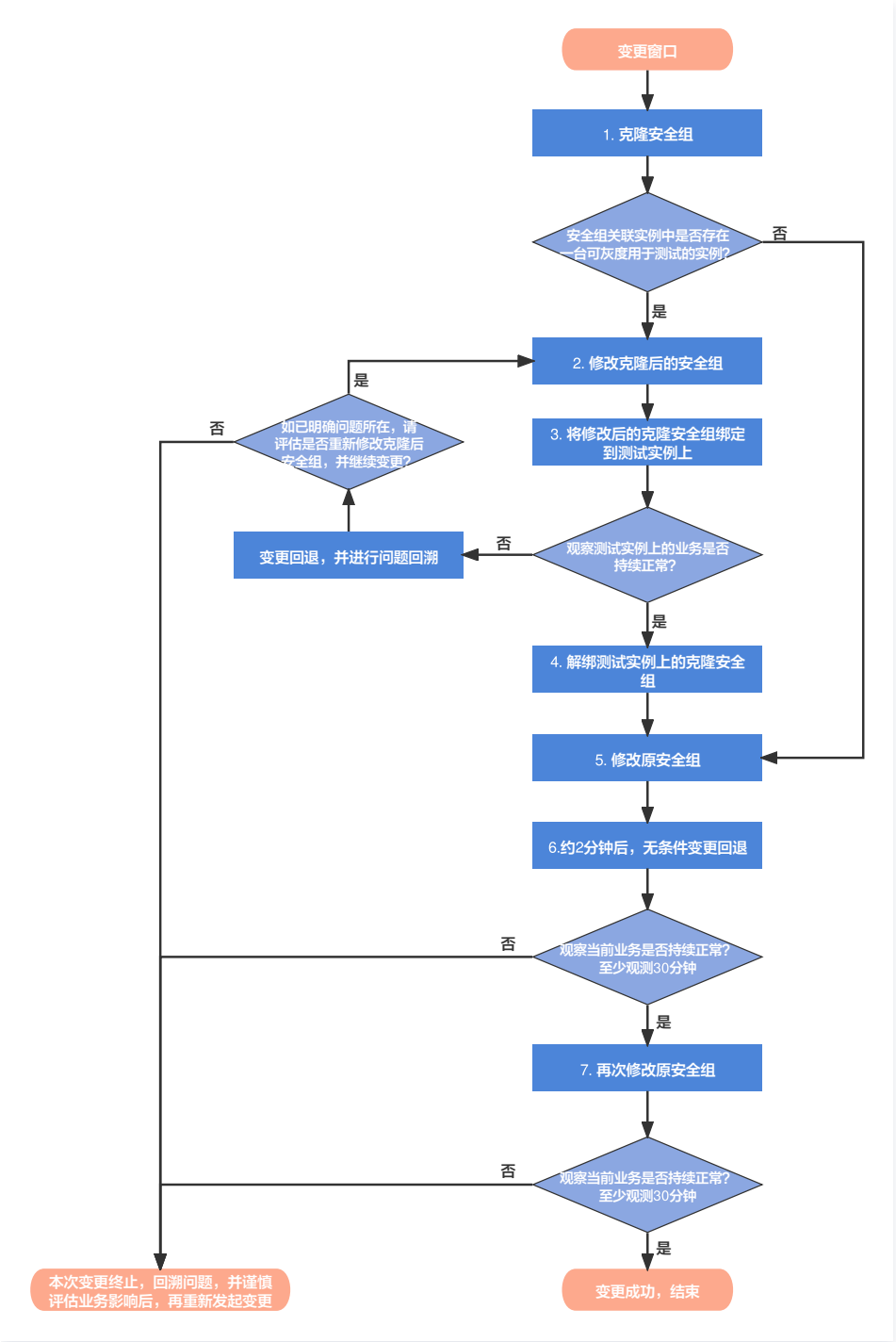
安全组是一种虚拟防火墙，具备有状态的数据包过滤功能，用于设置云服务器、负载均衡、云数据库等实例级别的流量访问控制，是重要的网络安全隔离手段。为满足业务需要，日常运维中可能会有安全组变更的需求，而变更安全组往往会对关联实例产生一定影响，为保证变更顺利进行，并将业务影响降至最低，本文提供安全组变更时的流程建议，及最佳实践案例供您参考。

观看视频

安全组变更流程

通过安全组变更流程规范，不仅能够在变更过程中及时发现问题，更能在变更过程中最大限度地降低业务影响。

> **说明：**
> 实际操作时建议您按照流程规范操作，保证变更顺利进行；如已明确无业务影响，也可根据实际情况适当简化操作。

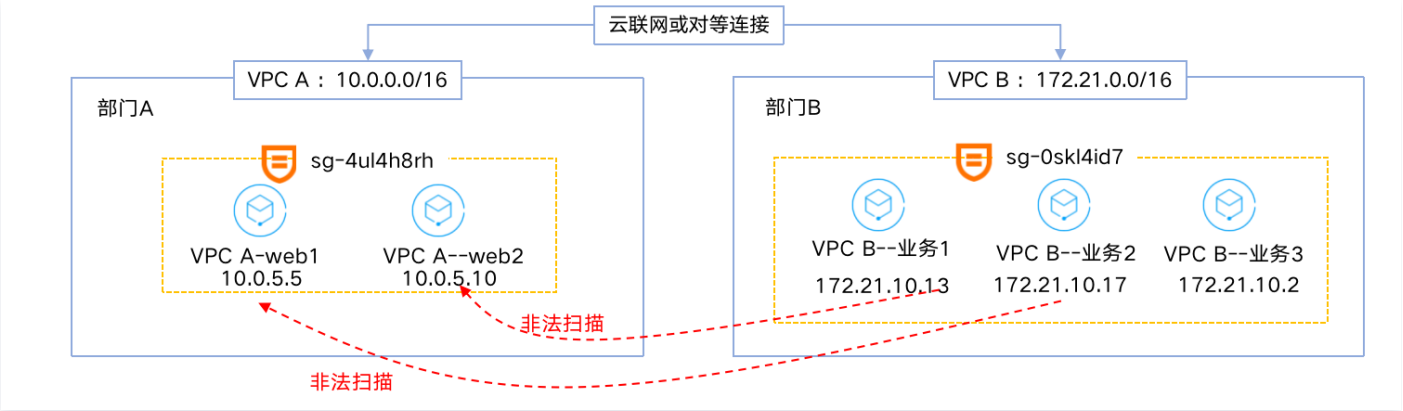

变更窗口
1. 克隆安全组
安全组关联实例中是否存在一台可灰度用于测试的实例?
否
是
2. 修改克隆后的安全组
是
如已明确问题所在，请评估是否重新修改克隆后安全组，并继续变更?
否
3. 将修改后的克隆安全组绑定到测试实例上
变更回退，并进行问题回溯
否
观察测试实例上的业务是否持续正常?
是
4. 解绑测试实例上的克隆安全组
5. 修改原安全组
6.约2分钟后，无条件变更回退
否
观察当前业务是否持续正常?至少观测30分钟
是
7. 再次修改原安全组
否
观察当前业务是否持续正常?至少观测30分钟
是
本次变更终止，回溯问题，并谨慎评估业务影响后，再重新发起变更
变更成功，结束

安全组变更示例

最近更新时间：2024-09-10 11:00:21

场景描述

- 部门 A 的 VPC A 和部门 B 的 VPC B 已经通过云联网或对等连接打通，VPC B 的云服务器可以访问 VPC A 中云服务器上的80端口服务。
- 部门 A 安全部门在巡检中发现，IP 地址为 `172.21.10.13` 和 `172.21.10.17` 的云服务器持续扫描 VPC A 中的非开放端口，造成了攻击式影响。
- 部门 A 业务运维人员接安全部门通知后，决定在安全组规则中针对来源 IP 为 `172.21.10.13` 和 `172.21.10.17` 的访问进行封禁。



操作步骤

> **注意：**
> 建议在业务停服或业务低峰时进行操作。

步骤一：克隆安全组

1. 登录 私有网络控制台，在左侧导航中选择**安全** > **安全组**。
2. 找到部门 A 受访问攻击的实例关联的安全组，如本例中 sg-4ul4h8rh，单击**更多** > **克隆**。

3. 输入克隆安全组的新名称，并单击**确定**。

步骤二：修改克隆后的安全组

> **说明：**
> 本例中安全组关联了多个实例，经评估，可将 VPC A 中 web1 实例灰度作为测试实例，稍后用于绑定修改后的克隆安全组进行业务验证。

1. 单击**克隆安全组 ID**，进入安全组规则详情页。
2. 在入站规则页签，单击**添加规则**。
3. 在弹出的“添加入站规则”对话框中，添加攻击方来源 IP（例如 `172.21.10.13` 和 `172.21.10.17` ）的“拒绝”策略，单击**完成**。

> **注意：**
> 此处经常出现的错误如下：
> - **策略填写错误**：例如本例是要禁用攻击方 IP 的流量，那么策略应该是**拒绝**，如果手误填为**允许**，将导致无法按照预期禁用流量。
> - **填写网段范围太大**：例如本例中只需要禁用 VPC B 中业务1和业务2两台云服务器，而业务3仍需要与 VPC A 通信，如果将入站规则中“来源”填为“ `172.21.10.0/24` ”网段，将导致 VPC B 中业务3也无法访问 VPC A 中的云服务器，不符合预期，因此，如果涉及实例不多，请尽量填写具体 IP 地址，如涉及较多，请尽可能精确填写网段范围，减少不必要的影响。

步骤三：将修改后的克隆安全组绑定到测试实例上

> **说明：**
> 当实例绑定多个安全组时，安全组将由上至下顺序匹配，本例请将克隆安全组移至测试实例原有安全组的最上面，确保克隆安全组优先匹配生效。

1. 单击克隆安全组右侧操作列的**管理实例**，进入“关联实例”界面。
2. 单击**新增关联**，选择需要关联的实例，如本例中的测试实例 web1。

3. 单击**确定**。
4. 观测测试实例业务是否持续正常。
   - 如业务正常，则执行下一步。
   - 如业务异常，则变更回退，进行问题回溯，如在变更窗口内已明确问题，请评估是否继续修改克隆安全组并继续变更，如是，则重复 步骤二；如否，则本次变更终止，结束。

   > **说明：**
   > 经观测本例中测试实例上的业务持续正常，说明修改克隆安全组符合预期。

步骤四：解绑测试实例上的克隆安全组

> **说明：**
> - 将测试实例上的克隆安全组进行解绑，稍后修改原安全组。
> - 请注意，如果一个实例只关联了一个安全组，则无法进行安全组解绑操作。

1. 单击克隆安全组右侧操作列的**管理实例**，进入**关联实例**界面。

2. 在需要解绑安全组的实例（如本例中的测试实例web1）右侧，单击**移出安全组**完成该实例安全组的解绑。

步骤五：修改原安全组

1. 单击**原安全组 ID**，本例中为 VPC A 中 web1 和 web2 绑定的安全组 ID，进入安全组规则详情页。
2. 在入站规则页签，单击**添加规则**。
3. 在弹出的**添加入站规则**对话框中，添加攻击方来源 IP（例如 `172.21.10.13` 和 `172.21.10.17` ）的**拒绝**策略，请确保规则中来源、策略等填写无误后，单击**完成**。

步骤六：2分钟后无条件变更回退

> **说明：**
> 无条件变更回退，目的在于能够及时发现安全组变更中是否存在对周边业务的短暂影响，以便及时做出应对决策，降低业务影响程度。

1. 修改完原安全组规则后，等待约2分钟，进行无条件回退，即删除 步骤五 中对安全组的修改。
2. 观察相关业务是否均持续正常，至少观测30分钟。

- 如所有业务均正常，则执行下一步。
- 如有部分业务反馈异常，建议立即终止本次变更，待问题回溯并评估业务影响后，再重新发起变更。

> **说明：**
> 本例中业务观测30分钟后，业务均正常。

步骤七：再次修改原安全组

1. 再次执行 步骤五 的操作，即单击**原安全组 ID**，本例中为 VPC A 中 web1 和 web2 绑定的安全组 ID，进入安全组规则详情页。
2. 在入站规则页签，单击**添加规则**。
3. 在弹出的**添加入站规则**对话框中，添加攻击方来源 IP（例如 `172.21.10.13` 和 `172.21.10.17` ）的“拒绝”策略，请确保规则中来源、策略等填写无误后，单击**完成**。
4. 观察相关业务是否均持续正常，至少观测30分钟。
   - 如所有业务均正常，则变更成功，结束。
   - 如有部分业务反馈异常，建议立即终止本次变更，待问题回溯并评估业务影响后，再重新发起变更。

> **说明：**
> 本例中业务运行正常，本次变更成功，结束。

配置云服务器为公网网关

最近更新时间：2025-06-18 16:10:12

> **警告**
> **使用单台云服务器 CVM 作为公网网关存在单点故障风险，生产环境建议使用 NAT 网关 。**

2019年12月06日后，腾讯云不支持在云服务器购买页勾选配置公网网关。如果您有需要，请按照本文所示方法自行配置。

操作场景

当您在腾讯云 VPC 中的部分云服务器没有普通公网 IP，但需要访问公网时，可以利用带有公网 IP（普通公网 IP 或弹性公网 IP ）的云服务器访问公网。公网网关云服务器将对出网流量进行源地址转换，所有其他云服务器访问公网的流量经过公网网关云服务器后，源 IP 都被转换为公网网关云服务器的公网 IP 地址，如下图所示：

前提条件

- 已登录 云服务器控制台 。
- 公网网关云服务器只能转发非所在子网的路由转发请求，因此，公网网关云服务器不能与需要借助公网网关访问公网的云服务器处于同一个子网下。
- 公网网关云服务器必须为 Linux 云服务器，Windows 云服务器无法作为公网网关使用。

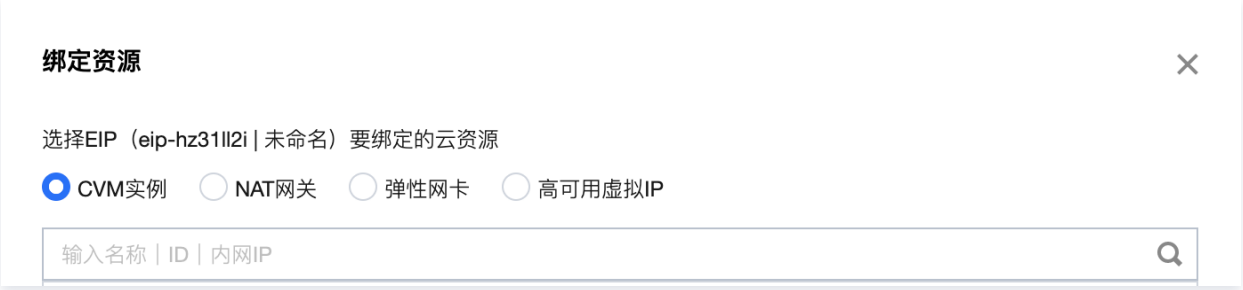
操作步骤

步骤1：绑定弹性公网 IP（可选）

> **说明：**
> 如果用作公网网关的云服务器已经有公网 IP 地址，请跳过此步骤，完成后续步骤。

1. 登录 云服务器控制台 在左侧导航栏中，单击 弹性公网 IP ，进入弹性公网 IP 管理页面。
2. 在需要绑定实例的弹性公网 IP 的操作栏下，选择**更多** > **绑定**。

3. 在**绑定资源**弹框中，选择一个被选做公网网关的 CVM 实例进行绑定。



步骤2：配置网关所在子网路由表

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

1. 创建自定义路由表。
2. 创建后会提示关联子网操作，直接关联公网网关服务器所在子网即可。

步骤3：配置普通子网路由表

配置普通子网的路由表，配置默认路由走公网网关云服务器，使得普通子网内的云服务器能通过公网网关的路由转发能力访问公网。
在普通云服务器所在子网的路由表中，新增如下路由策略：

- 目的端：您要访问的公网地址。
- 下一跳类型：云服务器。
- 下一跳：步骤1中绑定弹性公网 IP 的云服务器实例的内网 IP。
  具体操作请参见 配置路由策略。

步骤4：配置公网网关

1. 登录 公网网关云服务器，执行如下操作开启网络转发及NAT 代理功能。
   1.1 执行如下命令，在 `usr/local/sbin` 目录下新建脚本 `vpcGateway.sh`。

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

   1.2 按 i 切换至编辑模式，将如下代码写入脚本中。

```
#!/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 -o lo -m addrtype --src-type LOCAL -j ACCEPT
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 按 **Esc**，输入 **:wq**，保存文件并返回。

1.4 执行如下命令，设置脚本文件权限。

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

2. 设置公网网关的 rps：

2.1 执行如下命令，在 `usr/local/sbin` 目录下新建脚本 `set_rps.sh` 。

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

2.2 按 i 切换至编辑模式，将如下代码写入脚本中。

```
# !/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 按 **Esc**，输入 **:wq**，保存文件并返回。

2.4 执行如下命令，设置脚本文件权限。

```
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. 完成上述配置后，重启公网网关云服务器使配置生效，并在无公网 IP 的云服务器上，测试是否能够成功访问公网。

用 HAVIP+Keepalived 搭建高可用主备集群

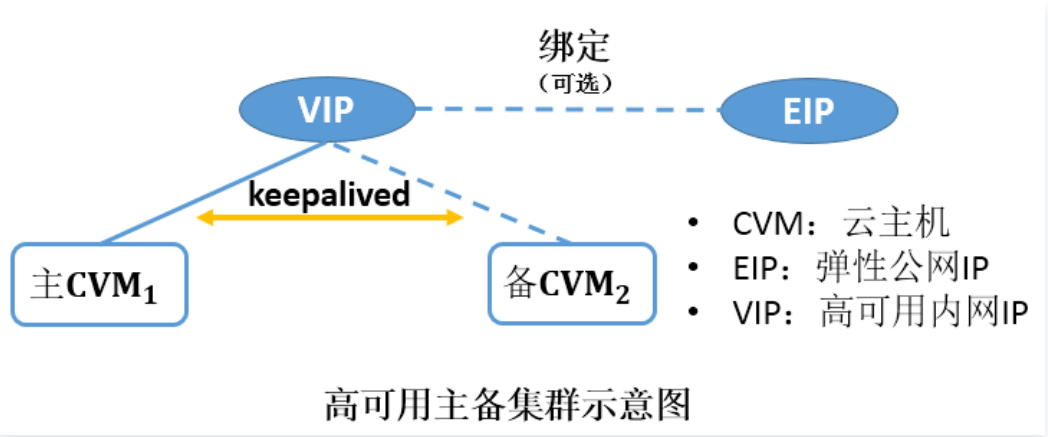
最近更新时间：2025-03-27 11:22:02

本文将介绍如何在腾讯云 VPC 内通过 keepalived 软件 + 高可用虚拟 IP (HAVIP) 搭建高可用主备集群。

> **说明**
> 目前 HAVIP 产品处于灰度优化中，切换的时延在10s左右，如有需要，请提交 工单申请 。

基本原理

通常高可用主备集群包含2台服务器，一台主服务器处于某种业务的激活状态（即 Active 状态），另一台备服务器处于该业务的备用状态（即 Standby 状态），它们共享同一个 VIP（Virtual IP）。同一时刻，VIP 只在一台主设备上生效，当主服务器出现问题时，备用服务器接管 VIP 继续提供服务。高可用主备模式有着广泛的应用，例如，MySQL 主备切换、Nginx Web 接入。
在 VPC 的云服务器间可以通过部署 Keepalived 来实现高可用主备集群。Keepalived 是基于 vrrp 协议的一款高可用软件，Keepalived 配置通过 keepalived.conf 文件完成。



- 在传统的物理网络中，可以通过 keepalived 的 VRRP 协议协商主备状态，其原理是：主设备周期性发送免费 ARP 报文刷新上联交换机的 MAC 表或终端 ARP 表，触发 VIP 迁移到主设备上。
- 在腾讯云 VPC 中，支持部署 keepalived 来搭建主备高可用集群。与物理网络相比，主要区别是：
  - 使用的 VIP 必须是从腾讯云申请的 高可用虚拟 IP (HAVIP) 概述 。
  - VIP 有子网属性，只能在同一个子网下的机器间宣告绑定。

注意事项

- 推荐使用单播方式进行 VRRP 通信。

> **说明**
> 本文演示配置为单播模式，如果使用组播方式进行 VRRP 通信，需提交 组播内测申请，待内测申请通过后参考 开启或关闭组播功能 打开 VPC 组播开关；同时在 keepalived 配置文件中无需配置对端设备的 IP 地址，即**不配置** “unicast_peer” 参数。

- 推荐使用 Keepalived（**1.2.24版本及以上**）。
- 确保已经配置以下 garp 相关参数。因为 keepalived 依赖 ARP 报文更新 IP 信息，如果缺少以下参数，会导致某些场景下，主设备不发送 ARP 导致通信异常。

```
garp_master_delay 1
garp_master_refresh 5
```

- 确保同一 VPC 下的每个主备集群需要配置不同的 vrrp router id。
- 确定没有采用 strict 模式，即需要删除“vrrp_strict” 配置。
- 控制单个网卡上配置的 VIP 数量，建议目前在单个网卡绑定的高可用虚拟 IP 数量不超过5个。如果需要使用多个虚拟 IP，建议在 keepalived 配置文件的 global_defs 段落添加或修改配置 “vrrp_garp_master_repeat 1”。

- 通过调节 advert_int 参数的大小，在抗网络抖动及灾害恢复速度进行平衡取舍。当 advert_int 参数过小，容易受网络抖动影响发生频繁倒换和暂时 **双主（脑裂）** 直到网络恢复。当 advert_int 参数过大，会导致主机器故障后，主备倒换慢（即服务暂停时间长）。**请充分评估双主（脑裂）对业务的影响！**
- track_script 脚本的具体执行项（如 checkhaproxy ）中的 interval 参数请适当提高，避免脚本执行超时导致 FAULT 状态的发生。
- 可选：注意日志打印导致的磁盘使用量上涨，可以通过 logrotate 等工具解决。

操作步骤

> **注意**
> 本文操作步骤均以如下环境条件为例，实际操作时，请您务必使用实际环境参数进行替换。
> - 主节点云服务器：HAVIP-01，172.16.16.5
> - 备节点云服务器：HAVIP-02，172.16.16.6
> - 高可用HAVIP：172.16.16.12
> - 弹性公网IP：81.71.14.118
> - 镜像版本：CentOS 7.6 64位

步骤1：申请 VIP

1. 登录 私有网络控制台 。
2. 在左侧导航栏中，选择 **IP 与网卡** > **高可用虚拟 IP**。
3. 在 HAVIP 管理页面，选择所在地域，单击**申请**。
4. 在弹出的**申请高可用虚拟 IP** 对话框中输入名称，选择 HAVIP 所在的私有网络和子网等信息，单击**确定**即可。

> **说明**
> HAVIP 的 IP 地址可以自动分配，也可以手动填写。如果您选择手动填写，请确认填写内网 IP 在所属子网网段内，且不属于系统保留 IP。例如，所属子网网段为：10.0.0.0/24，则可填的内网 IP 范围 为：10.0.0.2 – 10.0.0.254。

申请成功的 HAVIP 如下图所示。

在详情页选择绑定的实例，填写 HAVIP 可切换的实例范围。

步骤2：在主服务器和备服务器上安装 keepalived 软件（推荐1.2.24版本及以上）

本文以 CentOS 7.6镜像类型服务器为例提供 keepalived 的安装方法。

1. 查看 keepalived 软件包版本号是否符合要求。

```
yum list keepalived
```

- 是 = 执行2
- 否 = 执行3

2. 使用 yum 方式安装软件包。

```
yum install -y keepalived
```

3. 使用源码方式安装软件包。

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

步骤3：配置 keepalived，绑定高可用 VIP 到主备云服务器

1. 登录主节点云服务器 HAVIP-01，执行 `vim /etc/keepalived/keepalived.conf`，修改相关配置。

> **说明**
> HAVIP-01 和 HAVIP-02 在本例中将被配置成“等权重节点”，即 state 均为 BACKUP，priority 均为 100。优点是可以减少抖动造成的倒换次数。

```
! 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"    # 检测业务进程是否运行正常。其中“do_sth.sh”文件为用户自定义的业务进程检测脚本，请根据业务需要来执行，执行时“do_sth.sh”更换为实际的脚本名称。
        interval 5
   }
   vrrp_instance VI_1 {
   # 注意主备参数选择
   state BACKUP                # 设置初始状态均为“备“
       interface eth0            # 设置绑定 VIP 的网卡 例如 eth0
       virtual_router_id 51      # 配置集群 virtual_router_id 值
       nopreempt                 # 设置非抢占模式，
       # preempt_delay 10        # 仅 state MASTER 时生效
       priority 100              # 两设备是相同值的等权重节点
       advert_int 5
       authentication {
           auth_type PASS
           auth_pass 1111
       }
       unicast_src_ip 172.16.16.5  # 设置本机内网IP地址
       unicast_peer {
           172.16.16.6             # 对端设备的 IP 地址
       }
       virtual_ipaddress {
           172.16.16.12           # 设置高可用虚拟 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     # 设置当切为主状态后多久更新 ARP 缓存
       garp_master_refresh 5   # 设置主节点发送 ARP 报文的时间间隔

       track_interface {
                   eth0                # 使用绑定 VIP 的网卡 例如 eth0
           }
       track_script {
          checkhaproxy
       }
   }
```

2. 按“esc”退出编辑状态，输入 `:wq!` 保存并退出。
3. 登录备节点云服务器 HAVIP-02，执行 `vim /etc/keepalived/keepalived.conf`，修改相关配置。

```
! Configuration File for keepalived
global_defs {
   notification_email {
     acassen@firewall.loc
     failover@firewall.loc
     sysadmin@firewall.loc
   }
```

```
   notification_email_from Alexandre.Cassen@firewall.loc
   smtp_server 192.168.200.1
   smtp_connect_timeout 30
   router_id LVS_DEVEL
   vrrp_skip_check_adv_addr
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}
vrrp_script checkhaproxy
{
    script "/etc/keepalived/do_sth.sh"
     interval 5
}
vrrp_instance VI_1 {
# 注意主备参数选择
state BACKUP            # 设置初始状态均为“备“
    interface eth0          # 设置绑定 VIP 的网卡 例如 eth0
    virtual_router_id 51    # 配置集群 virtual_router_id 值
    nopreempt               # 设置非抢占模式
    # preempt_delay 10      # 仅 state MASTER 时生效
    priority 100            # 两设备是相同值的等权重节点
    advert_int 5
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    unicast_src_ip 172.16.16.6  # 设置本机内网 IP 地址
    unicast_peer {
        172.16.16.5             # 对端设备的 IP 地址
    }
    virtual_ipaddress {
        172.16.16.12           # 设置高可用虚拟 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    # 设置当切为主状态后多久更新 ARP 缓存
    garp_master_refresh 5   # 设置主节点发送ARP报文的时间间隔
    track_interface {
                eth0               # 使用绑定 VIP 的网卡 例如 eth0
        }
    track_script {
       checkhaproxy
    }
}
```

4. 按“esc”退出编辑状态，输入 `:wq` 保存并退出。
5. 重启 keepalived 进程使配置生效。

```
systemctl restart keepalived
```

6. 检查两台云服务器的主备状态，并确认 HAVIP 已经正确的绑定到主备服务器。

> **说明**
> 此示例中 HAVIP-01 先启动 keepalived 服务，所以正常情况下，HAVIP-01 将被选择为主节点。

登录 高可用虚拟 IP 控制台，可以看到 HAVIP 绑定的云服务器为主节点云 HAVIP-01，如下图所示。

步骤4：VIP 绑定弹性公网 IP（可选）

1. 在 高可用虚拟 IP 控制台，单击 步骤1 中申请的 HAVIP 所在行的**绑定**。

2. 在弹出的**绑定弹性公网 IP** 对话框中选择待绑定的 EIP，并单击**确定**。如果没有可用的 EIP，请先在 弹性公网 IP 控制台申请。

步骤5：使用 notify_action.sh 进行简单的日志记录（可选）

keepalived 主要日志仍然记录在“/var/log/message”中，可以通过添加 notify 的脚本来进行简单的日志记录。

1. 登录云服务器，执行 `vim /etc/keepalived/notify_action.sh` 命令添加脚本“notify_action.sh”，脚本内容如下：

```
#!/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. 执行 `chmod a+x /etc/keepalived/notify_action.sh` 修改脚本权限。

步骤6：验证主备倒换时 VIP 及外网 IP 是否正常切换

通过重启 keepalived 进程、重启子机等方式模拟主机故障，检测 VIP 是否能正常迁移。

- 如果完成了主备切换，则可以看到控制台的绑定主机已经切换为 backup 云服务器。
- 另外，也可以从 VPC 内 ping VIP 的方式，查看网络中断到恢复的时间间隔，每切换一次，ping 中断的时间大约为4秒。从公网侧 ping HAVIP 绑定的 EIP，可以查看网络中断到恢复的时间间隔，每切换一次，ping 中断的时间大致为4秒。
- 使用 ip addr show 检查 havip 是否出现主设备网卡上。

用 HAVIP+Windows Server Failover Cluster 搭建高可用 DB

最近更新时间：2023-07-24 18:33:43

1. **创建 HAVIP**
   登录 HAVIP 控制台，创建一个 HAVIP，具体方法请参见 创建高可用虚拟 IP 。
2. **绑定和配置**
   此处与传统模式配置一样，由后端机器声明和协商哪一设备绑定创建的 HAVIP。您只要在对应的配置文件中指定 virtual IP为 HAVIP。在群集管理器里，将刚才创建的 HAVIP 配置进去。

3. **验证**
   等待配置完成后，直接切换节点进行测试。

正常情况下会看到只有短暂中断后网络又通了（若切换较快甚至看不到中断），业务不受影响。

```
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
请求超时。
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
来自 172.18.0.121 的回复: 字节=32 时间<1ms TTL=128
```

版权所有：腾讯云计算（北京）有限责任公司

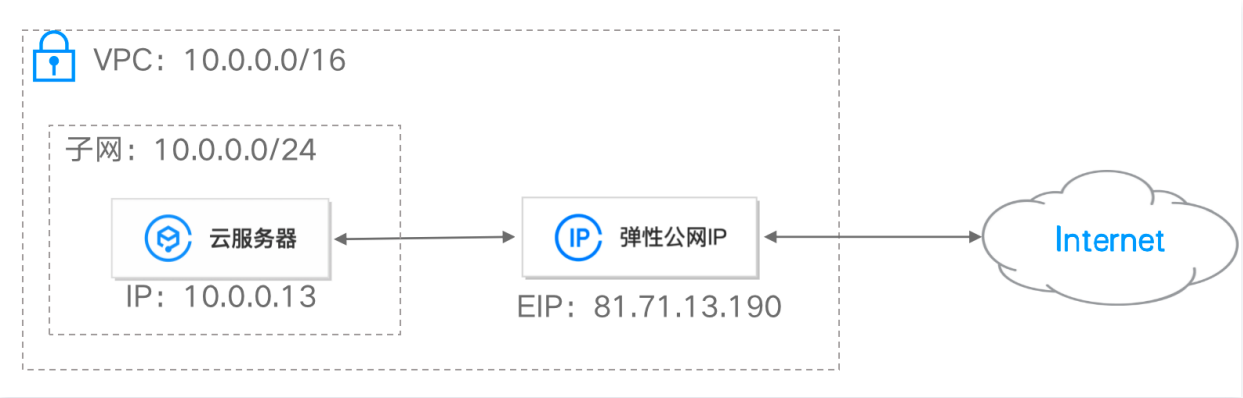
通过 EIP 实现云服务器访问 Internet

最近更新时间：2024-12-02 14:47:01

弹性公网 IP（EIP）是您可以独立购买和持有，且在某个地域下固定不变的公网 IP 地址，能够为私有网络的单台云服务器提供与公网交互的能力。本章节介绍单台云服务器通过绑定 EIP 来访问公网的详细操作。

操作场景

您 VPC 中的一台云服务器在购买时未分配普通公网 IP，不具备公网交互的能力，现因业务需要，需与公网通信。
为解决该场景的通信需求，您可以为云服务器绑定弹性公网 EIP。



操作步骤

步骤一：申请EIP

> **说明**
> 如已有闲置的 EIP，可跳过该步骤，直接执行 步骤二 。

1. 登录 私有网络控制台 。
2. 单击 **IP 与网卡** > **公网 IP**，进入公网 IP 界面。
3. 在“公网 IP”页面顶部，选择与云服务器相同的地域，然后单击**申请**。

4. 在弹出的“申请 EIP”界面，按实际需要配置参数，并单击**确定**。更多参数说明可参考 申请 EIP 。

步骤二：为云服务器绑定EIP

1. 在公网 IP 界面，选择 EIP 右侧的**更多** > **绑定**。

2. 在弹出的**绑定资源**窗口中，选择 **CVM 实例**，并勾选您的云服务器实例 ID，然后单击**确定**。

步骤三：验证通过EIP访问公网

1. 进入 云服务器控制台，单击云服务器右侧的**登录**，输入密码等信息，进入云服务器界面。
2. 执行如 `ping www.qq.com` 测试数据连通性，可看到有数据返回，表示该 CVM 可以访问公网。

```
[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
```

版权所有：腾讯云计算（北京）有限责任公司

通过专线接入和 VPN 连接实现混合云主备冗余通信

最近更新时间：2023-08-09 10:27:43

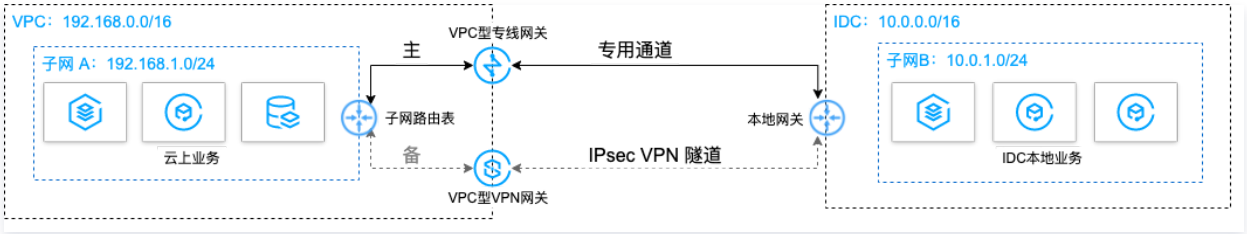
当用户业务分别部署于云下数据中心和云上 VPC 中时，可通过专线接入或 VPN 连接实现云上云下业务互通，为提升业务高可用性，可同时创建专线接入和 VPN 连接服务，结合 VPC 路由优先级功能，配置两条链路为主备链路，来实现冗余通信。本文指导您如何配置专线和 VPC 主备链路来实现云上云下混合通信。

> **说明**
> - 路由优先级功能目前处于内测中，如有需要，请 在线咨询 。
> - VPC 路由表中根据不同的下一跳类型定义了不同的优先级，目前默认路由优先级为：云联网 > 专线网关 > VPN 网关 > 其他。
> - 暂不支持控制台修改路由优先级，如需调整，请 在线咨询 。

业务场景

如下图所示，用户在 VPC 和 IDC 中部署了业务，为了实现云上与云下业务交互，用户需要部署网络连接服务来实现业务互通，为实现高可用通信，部署方案如下：

- 专线接入（主）：本地 IDC 通过物理专线，接入 VPC 的专线网关实现云下云上业务通信。在物理专线链路正常时，本地 IDC 与 VPC 之间所有的通信流量都通过物理专线进行转发。
- VPN 连接（备）：本地 IDC 与云上 VPC 通过建立 VPN 安全隧道来实现云上云下业务通信，当专线链路出现异常时，可将流量切换至该链路，确保业务可用性。



前提条件

- 用户本地 IDC 网关设备具有 IPsec VPN 功能，可同时作为用户侧 VPN 网关设备，与 VPC 侧 VPN 设备建立 IPsec 隧道通信。
- 用户 IDC 侧网关设备已配置静态 IP。
- 数据准备如下：

配置项			示例值
网络配置	VPC 信息	子网 CIDR	192.168.1.0/24
		VPN 网关公网 IP	203.xx.xx.82
	IDC 信息	子网 CIDR	10.0.1.0/24
		网关公网 IP	202.xx.xx.5

操作步骤

步骤一： 配置 IDC 通过专线接入上云

1. 登录 专线接入控制台 ，单击左侧导航栏的**物理专线**创建物理专线。
2. 单击左侧导航栏的**专线网关**创建专线网关，本例选择接入私有网络，标准型的专线网关，如果 IDC 和 VPC 通信网段冲突也可以选择 NAT 型。
3. 单击左侧导航栏的**专用通道**创建专用通道，此处需要配置通道名称、选择专线类型、已创建的专线网关、腾讯云侧和用户侧的互联 IP、路由方式选择静态路由、填写 IDC 通信网段等，配置完成后下载配置指引并在 IDC 设备完成配置。
4. 在 VPC 通信子网关联的路由表中配置下一跳为专线网关、目的端为 IDC 通信网段的路由策略。

> **说明**
> 更多详细配置可参考 专线接入快速入门 。

步骤二：配置 IDC 通过 VPN 连接上云

1. 登录 VPN 网关控制台 ，单击**新建**创建 VPN 网关，本例关联网络选择私有网络。
2. 单击左侧导航栏的**对端网关**，配置对端网关（即 IDC 侧 VPN 网关的逻辑对象），填写 IDC 侧 VPN 网关的公网 IP 地址，例如202.xx.xx.5。
3. 单击左侧导航栏的**VPN 通道**，请配置 SPD 策略、IKE、IPsec 等配置。
4. 在 IDC 本地网关设备上配置 VPN 通道信息，此处配置需要和步骤3中的 VPN 通道信息一致，否则 VPN 隧道无法正常连通。
5. 在 VPC 通信子网关联的路由表中配置下一跳为 VPN 网关、目的端为 IDC 通信网段的路由策略。

> **说明**
> 更多详细配置请参考 建立 VPC 到 IDC 的连接（路由表）。

步骤三：配置网络探测

> **说明**
> 如上两步配置完成后，VPC 去往 IDC 已经有两条路径，即下一跳为专线网关和 VPN 网关，根据路由默认优先级：专线网关 > VPN 网关，则专线网关为主路径，VPN 网关为备路径。

为了解主备路径的连接质量，需要分别配置两条路径的网络探测，实时监控到网络连接的时延、丢包率等关键指标，以探测主备路由的可用性。

1. 登录 网络探测控制台 。
2. 单击**新建**，创建网络探测，填写网络探测名称，选择私有网络、子网、探测目的IP，并指定源端下一跳路由，如专线网关。
3. 请再次执行 步骤2，指定源端下一跳路由为 VPN 网关。配置完成后，即可查看专线接入和VPN连接主备路径的网络探测时延和丢包率。

> **说明**
> 更多详细配置请参考 网络探测 。

步骤四：配置告警

为及时发现探测链路异常，可配置告警策略。当检测到链路异常时，告警信息将通过电子邮件和短信等形式发送到您，帮助您提前预警风险。

1. 登录腾讯云可观测平台下的 告警策略控制台 。
2. 单击**新建**，填写策略名称、策略类型选择**私有网络/网络探测**，告警对象选择具体的网络探测实例，配置触发条件和告警通知等信息，并单击**完成**即可。

步骤五：切换主备路由

当收到专线网关主路径的网络探测异常告警时，您需要手动禁用主路由，将流量切换至 VPN 网关备份路由上。

1. 登录 路由表控制台 。
2. 单击 VPC 通信子网关联路由表 ID，进入路由详情页，单击 禁用下一跳到专线网关的主路由，此时 VPC 去往 IDC 的流量将从专线网关切换至 VPN 网关。

通过云联网和 VPN 连接实现混合云主备冗余通信

最近更新时间：2023-08-09 10:27:44

当用户业务分别部署于云下数据中心和云上 VPC 中时，可通过云联网或 VPN 连接实现云上云下业务互通，为提升业务高可用性，可同时创建云联网和 VPN 连接服务，配置两条链路为主备链路，来实现冗余通信。本文指导您如何配置云联网和 VPN 主备链路来实现云上云下业务通信。

> **说明**
> 路由优先级功能目前处于内测中，如有需要，请 在线咨询 。

业务场景

如下图所示，用户在 VPC 和 IDC 中部署了业务，为了实现云上与云下业务交互，用户需要部署网络连接服务来实现业务互通，为实现高可用通信，部署方案如下：

- 云联网（主）：本地 IDC 通过物理专线，连接到云联网型专线网关，专线网关和 VPC 均接入云联网，从而实现云下云上全业务通信。在物理专线链路正常时，本地 IDC 与 VPC 之间所有的通信流量都通过云联网经物理专线进行转发。
- VPN 连接（备）：本地 IDC 与云上 VPC 通过建立 VPN 安全隧道来实现云上云下业务通信，当专线链路出现异常时，可将流量切换至该链路，确保业务可用性。

前提条件

- 用户本地 IDC 网关设备具有 IPsec VPN 功能，可同时作为用户侧 VPN 网关设备，与 VPC 侧 VPN 设备建立 IPsec 隧道通信。
- 用户 IDC 侧网关设备已配置静态 IP。
- 数据准备如下：

配置项			示例值
网络配置	VPC 信息	子网 CIDR	192.168.1.0/24
		VPN 网关公网 IP	203.xx.xx.82
	IDC 信息	子网 CIDR	10.0.1.0/24
		网关公网 IP	202.xx.xx.5

操作步骤

步骤一： 配置 IDC 通过云联网上云

1. 登录 专线接入控制台，单击左侧导航栏的**物理专线**创建物理专线。
2. 单击左侧导航栏的**专线网关**创建专线网关，本例选择接入云联网。
3. 单击云联网型专线网关 ID 进入详情页，在 **IDC 网关**中输入用户 IDC 网段，例如10.0.1.0/24。
4. 登录 云联网控制台，单击**新建**创建云联网实例。
5. 登录 专用通道控制台，单击**新建**创建专用通道连接云联网专线网关，此处配置通道名称、选择接入网络为云联网，选择已创建的云联网型专线网关、配置腾讯云侧和用户侧的互联 IP、路由方式选择 BGP 路由等，配置完成后下载配置指引并在 IDC 设备完成配置。
6. 将 VPC 和专线网关关联到云联网实例，即可实现 VPC 和 IDC 通过云联网、云联网专线网关进行互通。

> **说明**
> 更多详细配置请参考 IDC 通过云联网上云 。

步骤二：配置IDC通过VPN连接上云

1. 登录 VPN 网关控制台，单击**新建**创建 VPN 网关，本例关联网络选择私有网络。
2. 单击左侧导航栏的**对端网关**，配置对端网关（即 IDC 侧 VPN 网关的逻辑对象），填写 IDC 侧 VPN 网关的公网 IP 地址，例如202.xx.xx.5。
3. 单击左侧导航栏的 **VPN 通道**，请配置 SPD 策略、IKE、IPsec 等配置。
4. 在 IDC 本地网关设备上配置 VPN 通道信息，此处配置需要和 步骤3 中的 VPN 通道信息一致，否则 VPN 隧道无法正常连通。
5. 在 VPC 通信子网关联的路由表中配置下一跳为 VPN 网关、目的端为 IDC 通信网段的路由策略。

> **说明**
> 更多详细配置请参考：
> - 如果是1.0和2.0版本的 VPN 网关，请参考 建立 VPC 到 IDC 的连接（SPD 策略）。
> - 如果是3.0版本的 VPN 网关，请参考 建立 VPC 到 IDC 的连接（路由表）。

步骤三：配置网络探测

> **说明**
> 如上两步配置完成后，VPC 去往 IDC 已经有两条路径，即下一跳为云联网、VPN 网关，根据路由默认优先级：云联网 > VPN 网关，则云联网为主路径，VPN 网关为备路径。

为了解主备路径的连接质量，需要分别配置两条路径的网络探测，实时监控到网络连接的时延、丢包率等关键指标，以探测主备路由的可用性。

1. 登录 网络探测控制台。
2. 单击**新建**，创建网络探测，填写网络探测名称，选择私有网络、子网、探测目的 IP，并指定源端下一跳路由，如云联网。
3. 请再次执行 步骤2，指定源端下一跳路由为 VPN 网关。配置完成后，即可查看云联网和VPN连接主备路径的网络探测时延和丢包率。

> **说明**
> 更多详细配置请参考 网络探测。

步骤四：配置告警

为及时发现探测链路异常，可配置网络探测的告警策略，以便检测到链路异常时，可通过电子邮件和短信等及时获取到告警信息，帮助您提前预警风险。

1. 登录腾讯云可观测平台下的 告警策略控制台。
2. 单击**新建**，填写策略名称、策略类型选择**私有网络/网络探测**，告警对象选择具体的网络探测实例，配置触发条件和告警通知等信息，并单击**完成**即可。

步骤五：切换主备路由

当收到云联网主路径的网络探测异常告警时，您需要手动禁用主路由，将流量切换至 VPN 网关备份路由上。

1. 登录 路由表控制台。
2. 单击 VPC 通信子网关联路由表 ID，进入路由详情页，单击 禁用下一跳为云联网的主路由，此时 VPC 去往 IDC 的流量将从云联网切换至 VPN 网关。

自动更新安全组规则中放通登录的公网 IP

最近更新时间：2024-10-23 17:57:32

操作场景

家庭宽带的公网 IP 是由运营商分配的，运营商可能会不定时更换这些 IP 地址。如果您使用家庭宽带远程登录云服务器，并且安全组设置为只允许特定的公网 IP 登录，那么每当 IP 地址被更换时，您都需要对应地更新安全组规则中的 IP 地址，以便正常登录云服务器。部分办公网络也面临同样的问题。
本文将指导您在本地终端中通过调用接口自动获取本地出口的公网 IP，并将其更新到安全组规则里，减轻运营商更换公网 IP 所带来的影响。

前提条件

- 完成 TCCLI 工具安装，安装方式参考 安装TCCLI 。
- 完成 TCCLI 工具配置，配置方式参考 配置TCCLI ，通过浏览器登录腾讯云账号并授权以获取凭证。

方案说明

基本原理

CreateSecurityGroupPolicies 接口的 SecurityGroupPolicySet 中源地址（SourceCidr）支持传入字符串 MY_PUBLIC_IP，表示当前访问 API 的公网 IP 地址。您可通过 TCCLI 工具快速将本地公网出口 IP 添加到安全组规则中，创建放通对应 IP 的安全组规则，并添加备注，以便以后根据备注识别和删除旧的安全组规则。
主要操作指令如下：
1. 指定安全组所在地域，地域列表可通过接口 DescribeRegions 获取。

```
tccli configure set region ap-guangzhou  # ap-guangzhou 表示安全组所在地域，可按实际情况替换
```

2. 添加安全组规则，并添加备注以便后面识别和删除。

```
tccli vpc CreateSecurityGroupPolicies --SecurityGroupId sg-mdw2qvp3 --SecurityGroupPolicySet
'{"Ingress":
[{"Protocol":"tcp","Port":"22","CidrBlock":"MY_PUBLIC_IP","Action":"ACCEPT","PolicyIndex":"0","PolicyDe
scription":""}]}'

# 录入信息说明：
# sg-mdw2qvp3 表示需要修改的安全组ID
# Ingress表示入添加入站规则
# tcp表示协议类型
# Port表示远程登录需要放通端口，常见端口包括22,3389等
# CidrBlock表示需要放通的源地址，此处输入MY_PUBLIC_IP，表示当前访问云API的通往IP
# Action访问动过选择ACCEPT表示允许访问
# PolicyIndex输入0，表示将该规则添加到安全组第一条
# PolicyDescription表示备注信息，用户可根据实际情况自行填入。若不输入，则通过"MY_PUBLIC_IP"入参创建的安全组，默认备注
内容为“Replaced-From-MY_PUBLIC_IP”
```

3. 查询自动创建的安全组规则，并删除旧的规则。

```
tccli vpc DescribeSecurityGroupPolicies --SecurityGroupId sg-mdw2qvp3 --Filters
'[{"Name":"description","Values":["Replaced-From-MY_PUBLIC_IP"]}]'
# sg-mdw2qvp3 表示需要修改的安全组ID，通过Filters字段，填入备注中的"Replaced-From-MY_PUBLIC_IP"，快速查看规则添加
情况，获取到安全组规则的PolicyIndex值

tccli vpc DeleteSecurityGroupPolicies --SecurityGroupId sg-mdw2qvp3 --SecurityGroupPolicySet
'{"Ingress":[{"PolicyIndex":"N"}]}'
# sg-mdw2qvp3 表示要修改的安全组ID。PolicyIndex值"N"要替换成要删除的安全组规则的实际PolicyIndex值，
如"PolicyIndex":"2"
```

> **注意：**

> 安全组规则变更会影响服务器访问情况，配置前请仔细核对信息是否准确，避免影响业务正常运行。

操作步骤

信息确认

通过 安全组控制台，您可以获取到需要自动化配置的安全组 ID 以及地域信息。后续安全组 ID 和地域信息将作为本地脚本的必填参数使用。

创建本地脚本

创建脚本文件

```
vi CreateSecurityGroupPolicies.py   # CreateSecurityGroupPolicies.py表示脚本名称
```

> **说明：**
> 您可以参考 方案说明 中的步骤，编写脚本或使用示例脚本以自动更新安全组规则配置。

示例脚本

> **说明：**
> 1. **我们提供示例脚本供您参考，绝大部分使用场景可直接使用。**
> 2. **考虑到使用场景多样和公网环境复杂，我们针对典型的特殊场景提供了对应的参数供您调整。**
> - 当同时有多个潜在登录终端时，会出现多个终端先后使用本脚本修改同一个安全组的情况。为避免最后执行此脚本的终端修改了安全组对其他终端的放通规则，本脚本支持通过 HOSTNAME 参数区分不同终端，不同终端的规则互不影响。该参数默认为 `Auto-Created-For-Hostname-$` 主机名。
> - 当单个公网宽带出口同时有多个相互切换的 IP 时，本脚本支持通过 MAX_RESERVED_NUM 调整安全组对单终端放通的 IP 数量上限，即保留最近 MAX_RESERVED_NUM 次放通的 IP。脚本默认单终端所在的宽带出口只有1个公网 IP，即 MAX_RESERVED_NUM = 1。

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import json
import optparse
import socket
import subprocess
import sys

# 默认配置，可直接在此处修改，也可通过传参来调整
HOSTNAME = socket.gethostname()  # 主机名，用于区分不同客户端的放通策略
MAX_RESERVED_NUM = 1  # 自动放通策略的最大保留数量，用于清理过期的策略，存在多个出口IP时，可以按需调整
POLICY_PROTOCOL = 'TCP'  # 要放通的协议，通常是: TCP
POLICY_PORT = '22'  # 要放通的端口，通常是: 22(Linux/Unix)、3389(Windows)


def get_policy_description():
```

```
    return 'Auto-Created-For-Hostname-%s' % HOSTNAME


def get_describe_filters():
    return [{
        'Name': 'direction',
        'Values': ['INBOUND']
    }, {
        'Name': 'poly',
        'Values': ['ACCEPT']
    }, {
        'Name': 'protocol-type',
        'Values': [POLICY_PROTOCOL]
    }, {
        'Name': 'port',
        'Values': [POLICY_PORT]
    }, {
        'Name': 'description',
        'Values': [get_policy_description()]
    }]


def get_create_policy():
    return {'Ingress': [{
        'PolicyIndex': 0,
        'Action': 'ACCEPT',
        'Protocol': POLICY_PROTOCOL,
        'Port': POLICY_PORT,
        'CidrBlock': 'MY_PUBLIC_IP',  # MY_PUBLIC_IP 会自动替换为本地公网IP
        'PolicyDescription': get_policy_description(),
    }]}


def get_delete_filters(policies):
    _registry = {}
    delete_index_list = []
    for item in policies:
        if len(_registry) < MAX_RESERVED_NUM \
                and item['CidrBlock'] not in _registry:
            _registry[item['CidrBlock']] = None
        else:
            delete_index_list.append(item['PolicyIndex'])

    delete_filters = {'Ingress': []}
    for index in delete_index_list:
        delete_filters['Ingress'].append({
            'PolicyIndex': index,
            'Action': 'ACCEPT',
            'Protocol': POLICY_PROTOCOL,
            'Port': POLICY_PORT,
            'PolicyDescription': get_policy_description(),
        })

    return delete_filters


def tccli_cmd(cmd):
    p = subprocess.Popen('tccli %s' % cmd, shell=True,
                         stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    std_out, std_err = p.communicate()
```

```
    if p.returncode:
        sys.stderr.write(std_err.decode('utf-8'))
        sys.exit(p.returncode)

    if std_out:
        return json.loads(std_out.decode('utf-8'))


def main(SecurityGroupId):
    # 添加放通本地公网IP的规则
    tccli_cmd('vpc CreateSecurityGroupPolicies '
              '--SecurityGroupId %r '
              '--SecurityGroupPolicySet %r '
              % (SecurityGroupId, json.dumps(get_create_policy())))

    # 查询匹配特定条件的规则
    r = tccli_cmd('vpc DescribeSecurityGroupPolicies '
                  '--SecurityGroupId %r '
                  '--Filters %r '
                  '--filter "SecurityGroupPolicySet.Ingress" '
                  '--output json'
                  % (SecurityGroupId, json.dumps(get_describe_filters())))

    # 删除最大保留量以外的规则
    delete_filters = get_delete_filters(r)
    if delete_filters['Ingress']:
        tccli_cmd('vpc DeleteSecurityGroupPolicies '
                  '--SecurityGroupId %r '
                  '--SecurityGroupPolicySet %r '
                  % (SecurityGroupId, json.dumps(delete_filters)))

    print('done.')


if __name__ == '__main__':
    usage = 'usage: %prog [options] SecurityGroupId [SecurityGroupId ...]'
    parser = optparse.OptionParser(usage=usage)

    parser.add_option('-r', '--region')
    parser.add_option('-n', '--max-reserved-num', type=int,
                      help='default %r' % MAX_RESERVED_NUM)
    parser.add_option('-N', '--hostname', help='default %r' % HOSTNAME)
    parser.add_option('-p', '--port', help='default %r' % POLICY_PORT)
    parser.add_option('-P', '--protocol', help='default %r' % POLICY_PROTOCOL)

    options, args = parser.parse_args()

    if options.hostname is not None:
        HOSTNAME = options.hostname

    if options.max_reserved_num is not None:
        MAX_RESERVED_NUM = options.max_reserved_num

    if options.port is not None:
        POLICY_PORT = options.port

    if options.protocol is not None:
        POLICY_PROTOCOL = options.protocol

    # 设置 TCCLI 工作地域
    if options.region:
```

```
        tccli_cmd('configure set region %s' % options.region)

    for SecurityGroupId in set(args):
        main(SecurityGroupId)
```

运行示例脚本

情况一：直接使用，覆盖大部分场景。

```
python CreateSecurityGroupPolicies.py -r ap-guangzhou  sg-xxxx -p 22
# ap-guangzhou表示安全组所在地域，sg-xxxx表示安全组ID，22表示需要放通的端口
```

情况二：针对典型的特殊场景，您可以根据实例脚本说明调整对应的参数。

```
python CreateSecurityGroupPolicies.py -N MY_MAC -n 15 -r ap-guangzhou sg-xxxx -p 22
# MY_MAC表示HOSTNAME名字，15表示MAX_RESERVED_NUM数量，ap-guangzhou表示安全组所在地域，sg-xxxx表示安全组ID，22表示需
要放通的端口
```

将脚本设置为开机运行或定时运行

> **说明：**
> 脚本设置为开机运行或定时运行，您可以根据运营商更换公网 IP 的频率，合理设置时间间隔，以便正常登录云服务器。

将脚本设置为开机运行

MacOS

通过 crontab 工具设置开机运行

1. 打开终端，并执行命令 crontab -e。

```
crontab -e
```

2. 编辑定时任务文件，设置开机运行。

```
@reboot /path/to/Python /path/to/your/CreateSecurityGroupPolicies.py  -r  ap-guangzhou sg-xxxx
# 设置开机运行，ap-guangzhou表示安全组所在地域，sg-xxxx表示安全组ID
```

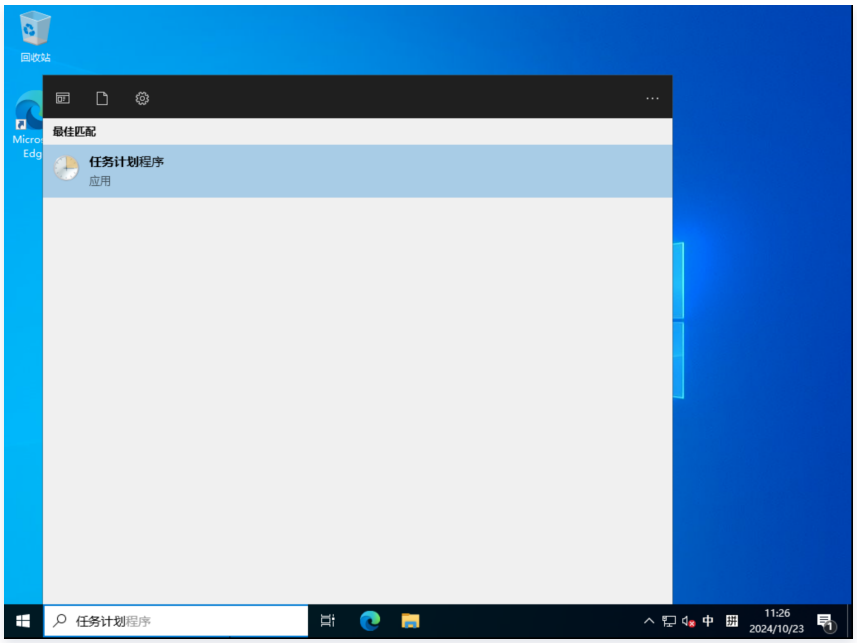
3. 保存并关闭文件。

```
: wq
```

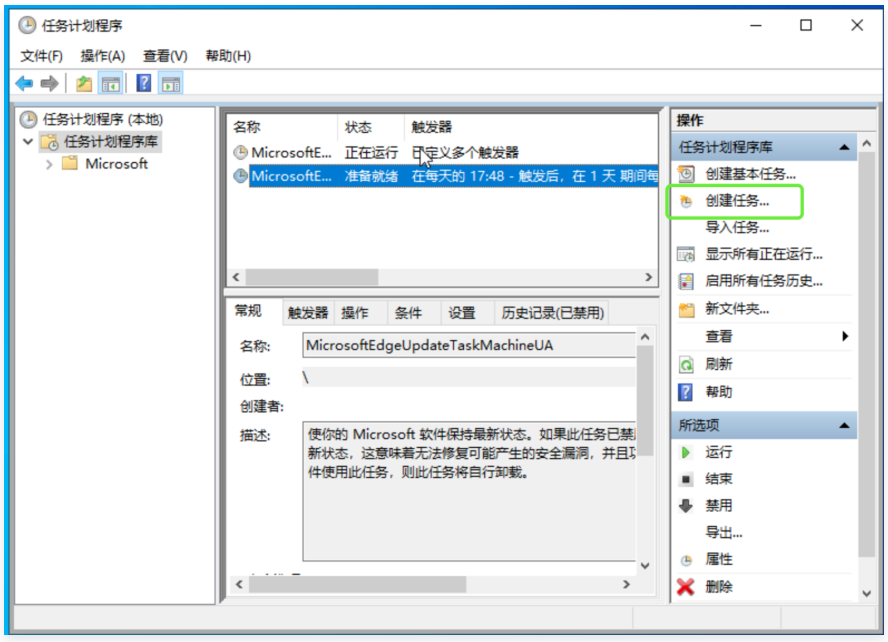
Windows 系统

通过 windows 任务计划程序设置开机运行

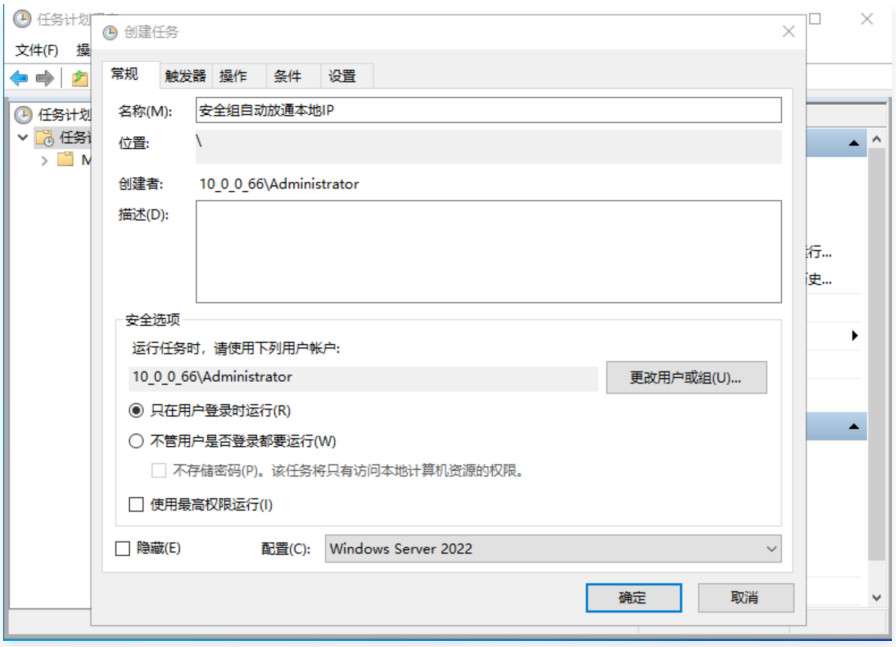
1. 打开任务计划程序。



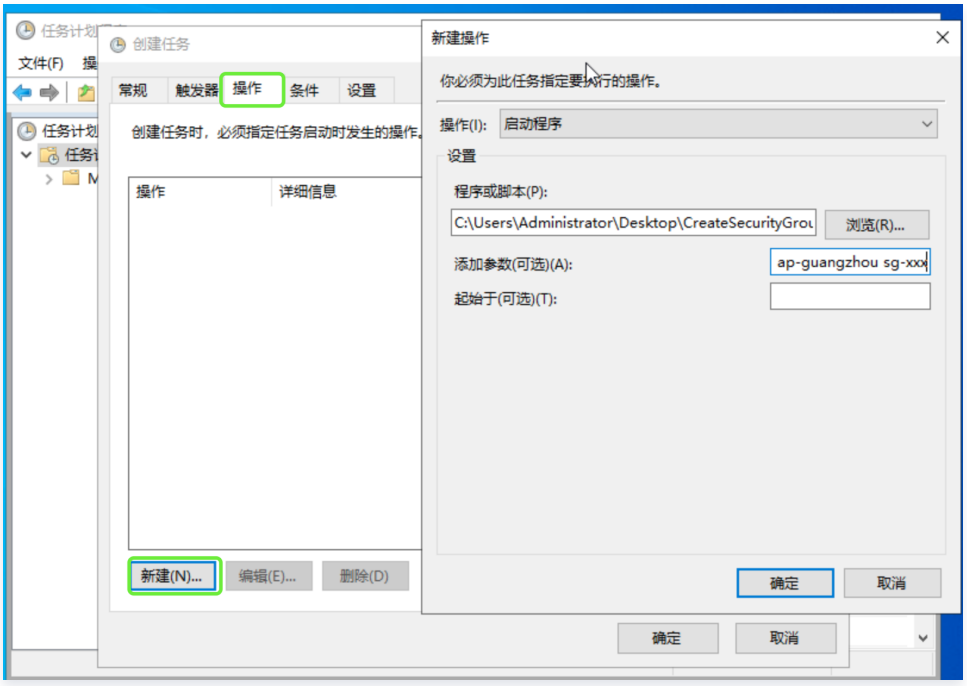
2. 添加计划任务，选择**创建任务**。



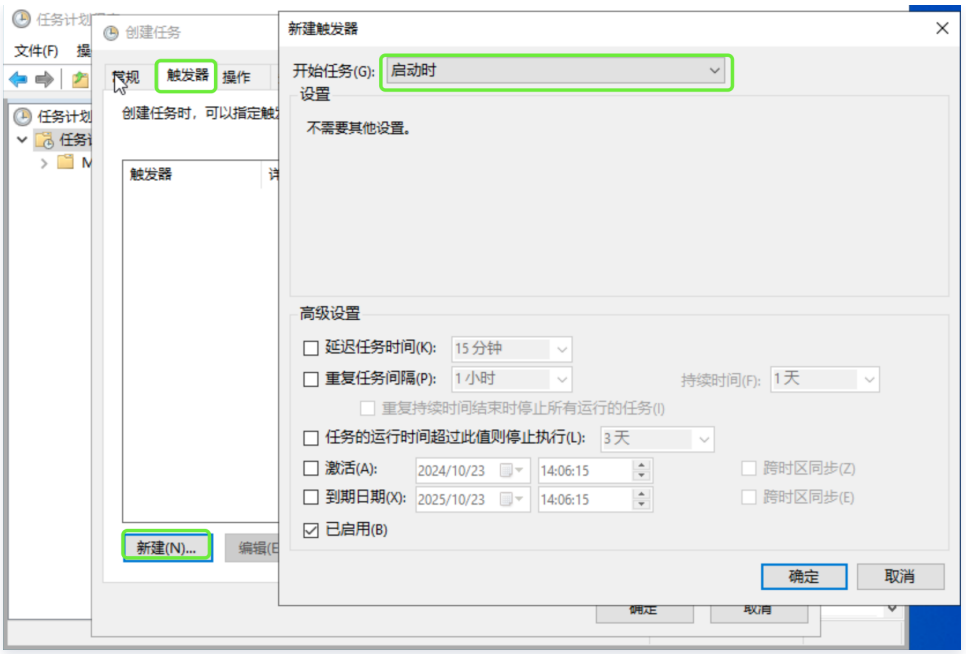
2.1 填写计划任务信息以及选择运行账户。



2.2 “程序或脚本”选择需要运行的脚本，“添加参数（可选）”填写脚本参数。



2.3 配置**触发器**，选择**启动时**运行。



3.单击**确定**保存。

将脚本设置为定时运行

MacOS

通过crontab工具设置定时运行

1. 打开终端，并执行命令 crontab -e。

```
crontab -e
```

2. 编辑定时任务文件，每行表示一个定时任务，可根据需要设置合理的运行时间。

```
0 0 * * * /path/to/Python /path/to/CreateSecurityGroupPolicies.py  -r  ap-guangzhou sg-xxxx
#表示每天的0点执行脚本。ap-guangzhou表示安全组所在地域，sg-xxxx表示安全组ID
```

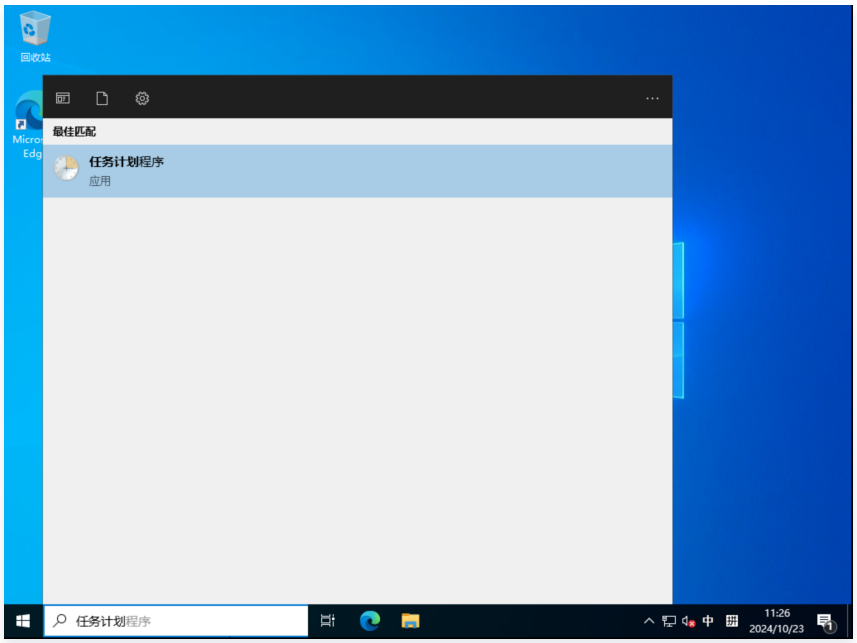
3. 保存。

```
: wq
```

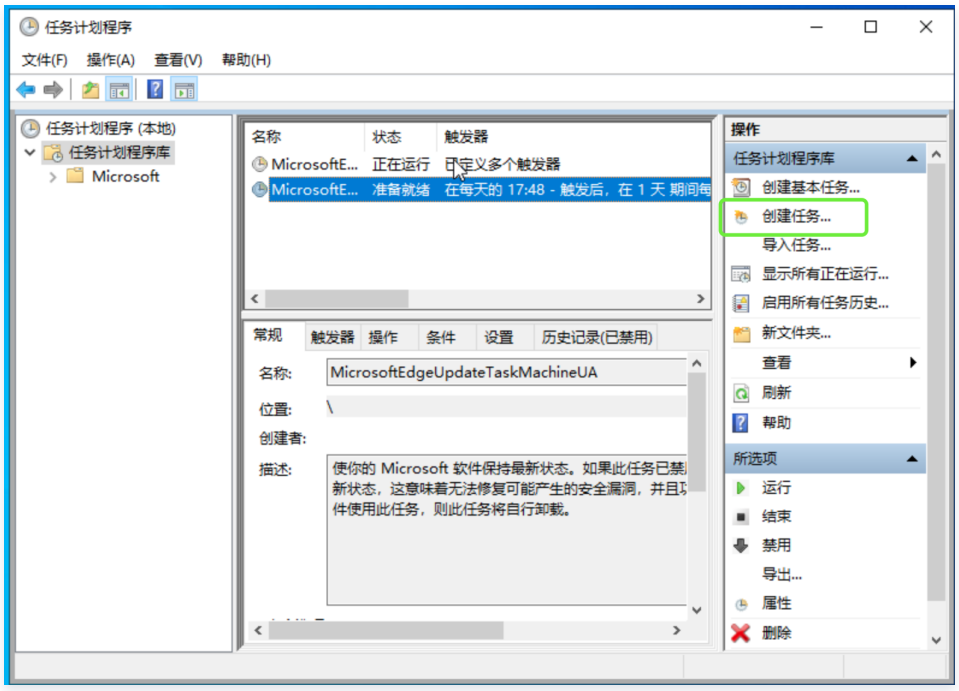
Windows 系统

通过 windows 任务计划程序设置定时运行

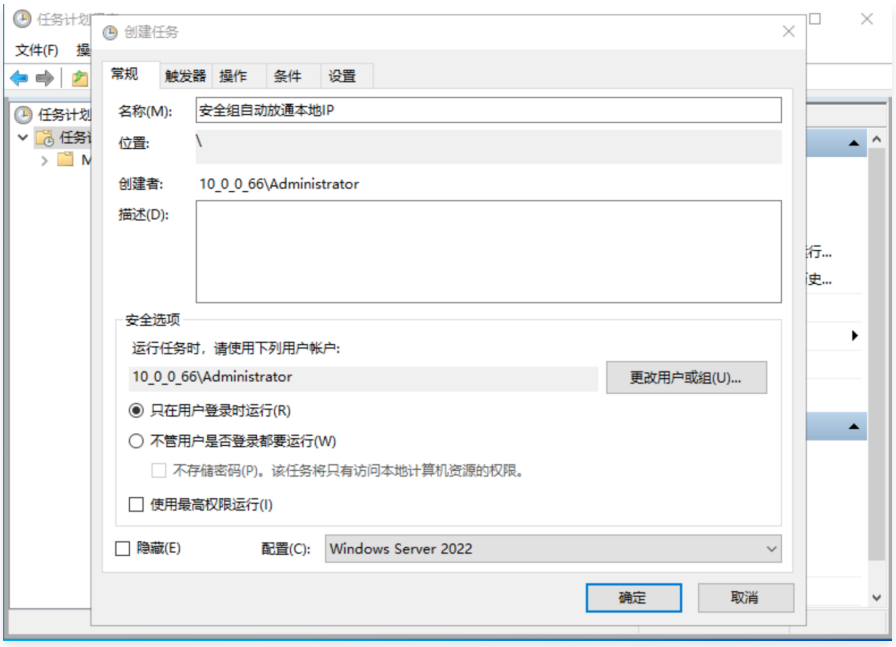
1. 打开任务计划程序。



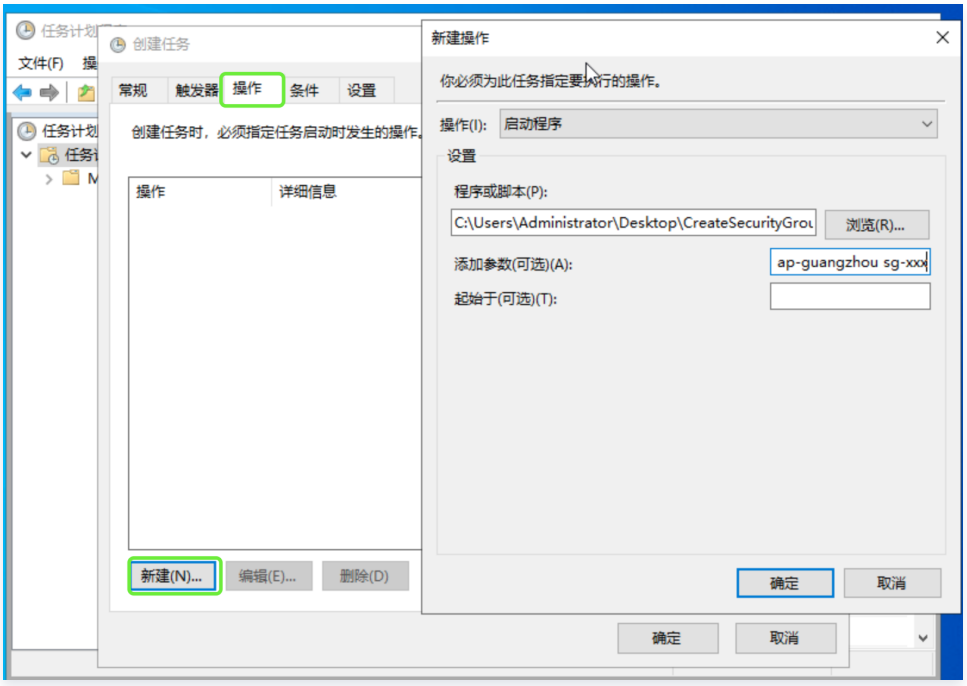
2. 添加计划任务，选择**创建任务**。



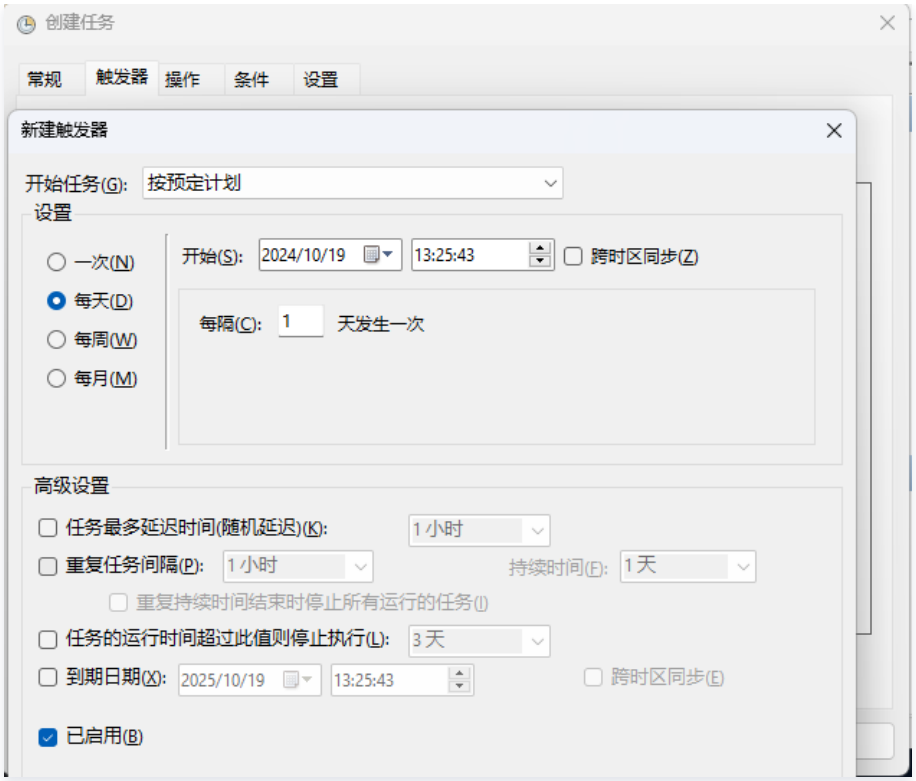
2.1 填写计划任务信息以及选择运行账户。



2.2 “程序或脚本”选择需要运行的脚本，“添加参数（可选）”填写脚本参数。



2.3 设置定时运行。


创建任务
常规
触发器
操作
条件
设置
新建触发器
开始任务(G):
按预定计划
设置
一次(N)
每天(D)
每周(W)
每月(M)
开始(S):
2024/10/19
13:25:43
跨时区同步(Z)
每隔(C):
1
天发生一次
高级设置
任务最多延迟时间(随机延迟)(K):
1 小时
重复任务间隔(P):
1 小时
持续时间(F):
1 天
重复持续时间结束时停止所有运行的任务(I)
任务的运行时间超过此值则停止执行(L):
3 天
到期日期(X):
2025/10/19
13:25:43
跨时区同步(E)
已启用(B)

3. 确认保存。