全球应用加速

操作指南

腾讯云

文档目录

TLS安全策略组

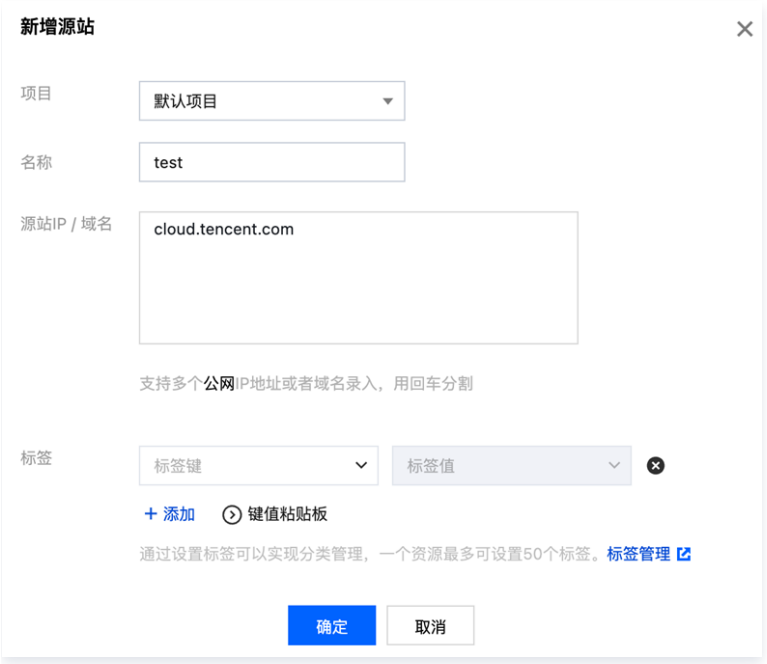
操作指南
全球应用加速

源站管理

最近更新时间：2024-11-06 12:17:42

增加源站

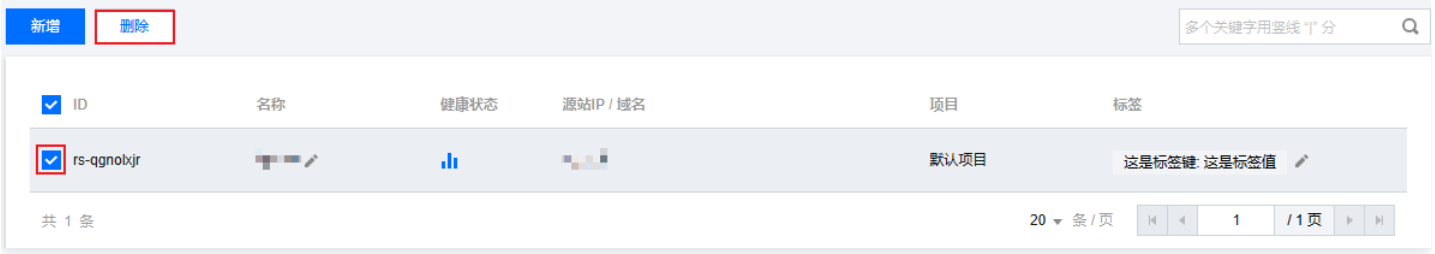
登录 全球应用加速控制台，在**源站管理**页面，单击**新增**，将所有要加速访问的服务器信息添加进来，可填写源站 IP 或域名，源站 IP 可支持 IPv4 和 IPv6，多个源站用回车分隔。



删除源站

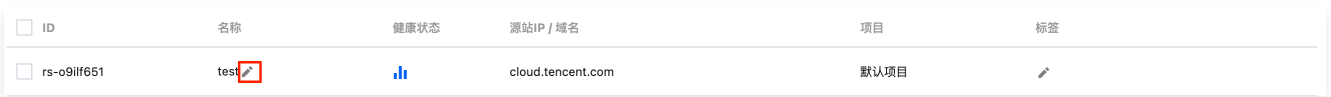
登录 全球应用加速控制台，在**源站管理**页面，选中待删除的源站，单击**删除**。

> **注意：**
> 如待删除源站已与现有通道进行绑定，请先进行解绑操作。



修改名称

1. 登录 全球应用加速控制台，在**源站管理**页面，单击源站名称旁的 ，对源站名称进行编辑。



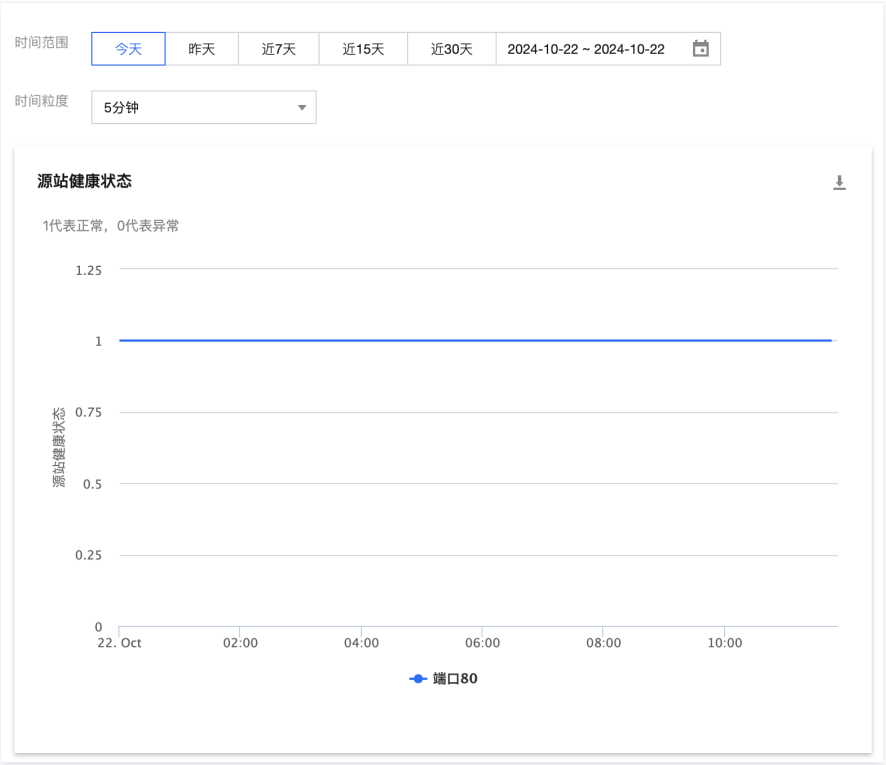
2. 在**修改名称**弹窗中，填写新的源站名称，单击**确定**即可完成修改。

查看源站健康状态

1. 登录 全球应用加速控制台，在**源站管理**页面，单击健康状态下的 图标。

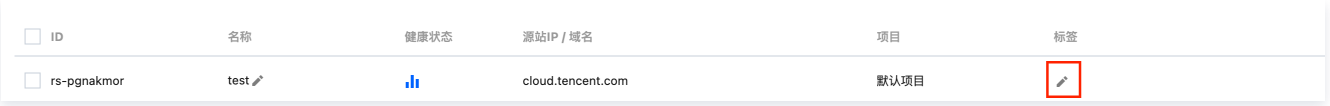
> **注意：**
> 如源站尚未绑定监听器，则无法使用该功能。

2. 在右上角的弹出框，查看不同时间范围/粒度的源站健康状态，1代表正常，0代表异常。



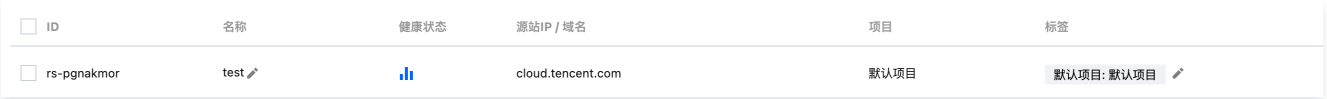
编辑标签

1. 登录 全球应用加速控制台，在**源站管理**页面，单击 。



2. 在**编辑标签**页面，选择资源下拉框中，选择**标签键**与**标签值**，用于从不同维度对源站进行分类管理，单击**确定**。

3. 添加完标签后，可以在**源站管理**页面查看标签情况。

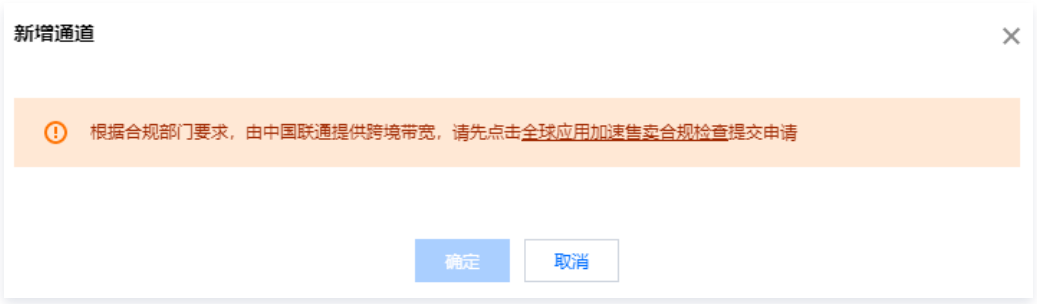


接入管理
通道管理（跨境通道）

最近更新时间：2024-10-25 10:56:22

新增通道

1. 登录 全球应用加速控制台，单击**接入管理 > 新增跨境通道。**
2. 初次使用需填写 全球应用加速合规售卖合规检查 审核资料，由联通侧审核确认(审查 SLA 2 – 3个工作日)，审核通过后才可进入下一步。



3. 审核通过之后，进入“接入管理”页面，单击**新增跨境通道**，在弹出的窗口中，填写通道信息。

> **注意：**
> 定制安全 EIP 功能尚未全量，如有需要，可通过 提交工单 联系我们。

- 项目：该通道所属项目（后续可以更换项目）。
- 通道名称：最多30个字符，支持中英文、常规符号。
- IP类型：选择普通公网IP或定制安全EIP，定制安全EIP会优先选用本地防护能力和全球覆盖较好的运营商线路，确保访问质量较好的同时，提供较大的防护能力。
- 接入节点：选择客户端所在区域或距离客户端最近区域的节点。

> **注意：**
> 若您需要提供中国香港的精品 BGP 网络接入，请在“接入节点”选择“香港”，选择“精品 BGP”。

- 回源节点：选择目标服务器所在区域或距离目标服务器最近区域的节点。
- 通道规则：选择通道的带宽上限，最大值2000Mbps；选择通道的最大并发连接数，最大值100万。
- 联通专线带宽：指专线带宽上限，根据通道规格带宽调整。
- 联通专线购买时长：由于采用月预付费方式，最小单位为1个月。
- 标签：非必填项，可通过设置标签实现对通道的分类管理。
- 费用：根据您选择的带宽与并发数，给出相应的通道费、带宽费、联通专线带宽费：
  a. 通道费：按日计算，直到通道被删除为止，请特别注意通道创建后未满一天删除也会按一天计费。
  b. 带宽费：按每日实际出入带宽峰值计费。
  c. 联通专线带宽费：月预付费，按选择的联通专线带宽上限计费。**(专线到期前删除跨境通道，联通专线带宽的费用需要在腾讯云费用中心取消云市场订单。)**

4. 单击**确定**，跳转到联通专线带宽费的付费页面，确认金额后单击**提交订单**，进行联通专线带宽费预付费。

付费完成后，返回 GAAP 控制台界面。单击**已支付完成**。

5. 在“接入管理”页面中，查看通道列表，单击 **ID/通道名**，可查看通道具体信息。

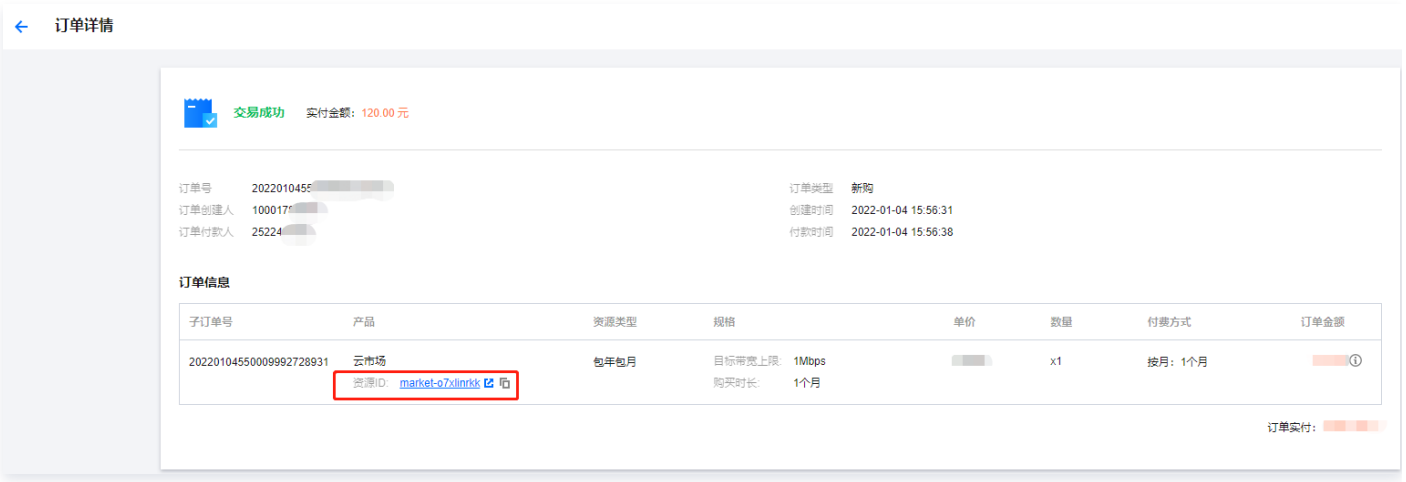
- ID/通道名：通道的 ID 和名字，其中通道名可以修改。
- VIP：用于客户端访问的接入 IP 地址。
- 域名：用于客户端访问的接入域名（系统分配，且自动绑定到 VIP）。
- 状态：仅“运行中”状态下，加速通道才可以正常使用。
- 续费：可在此为联通专线续费，选择联通专线有效时长进行预付费（1月、2月、1年等购买时长）。

> **注意：**
> 联通专线带宽费预付费用无法使用云账号余额支付，您可通过在线支付方式使用微信支付、QQ 钱包支付和网银支付等多种方式完成支付。

取消跨境专线订单

1. 单击**费用 > 订单管理**，根据订单创建时间、金额，找到对应的订单。

2. 单击进入订单**详情**，单击**产品：资源 ID**，进入订单详情。



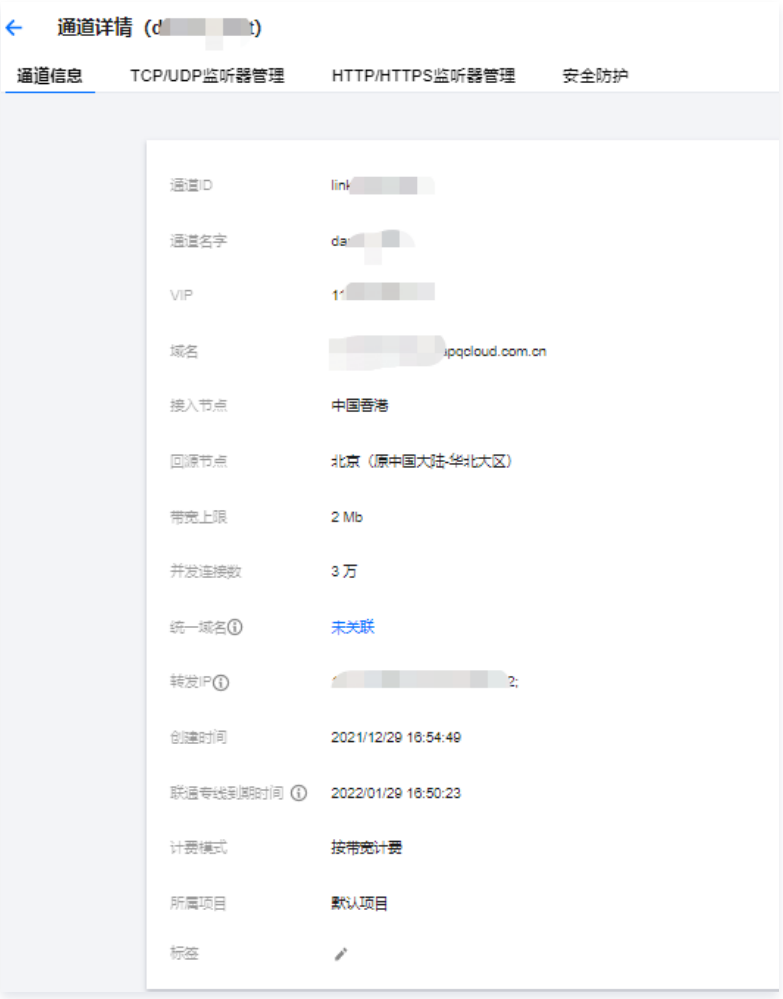
3. 单击**退款**，确认并提交退款信息。详细退款规则请查看 退款规则 。

查看通道信息

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击指定通道的 ID/通道名，进入下一级页面。

2. 单击修改配置，在修改通道信息弹窗中，当前只支持调整通道规格的并发上限，不支持带宽上限的扩容或者缩容，若您业务需要，请重新创建跨境通道，可单击复制进行监听器快速复制。

3. 在“通道信息”标签页，可以查看通道的详细信息。其中，“转发 IP”是指加速通道末端的转发节点 IP，该转发节点负责将加速通道的数据通过公网转发给源站。如果您希望多条通道使用同一个域名，可单击未关联跳转至 统一域名 页面进行配置。



通道管理（非跨境通道）

最近更新时间：2024-10-25 10:56:22

新增通道

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击**新增**。
2. 在弹出的窗口中，填写通道信息。

> **注意：**
> IPv6和定制安全EIP功能尚未全量，如有需要，可通过 工单联系 我们。

- 项目：该通道所属项目（后续可以更换项目）。
- 通道名称：最多30个字符，支持中英文、常规符号。
- IP 版本：可根据需要选择 IPv4 或 IPv6，其中 IPv6 暂时只支持国内区域。
- IP类型：选择普通公网IP或定制安全EIP，定制安全EIP会优先选用本地防护能力和全球覆盖较好的运营商线路，确保访问质量较好的同时，提供较大的防护能力。
- 接入节点：选择客户端所在区域或距离客户端最近区域的节点。

> **注意：**
> - 若您需要提供中国香港的精品 BGP 网络接入，请在“接入节点”选择“香港”，选择“精品BGP”。
> - 中国大陆提供三网节点网络，如有需要，可通过 提交工单 联系我们。

- 回源节点：选择目标服务器所在区域或距离目标服务器最近区域的节点。

> **注意：**

> 中国台湾无法与中国大陆进行直连。

- 通道规则：选择通道的带宽上限，最大值1000Mbps；选择通道的最大并发连接数，最大值100万。
- 标签：非必填项，可通过设置标签实现对通道的分类管理；
- 费用：根据您选择的带宽与并发数，下方会给出相应的通道费和带宽费。
  a. 通道费：按日计算，直到通道被删除为止，请特别注意通道创建后未满一天删除也会按一天计费；
  b. 带宽费：按每日实际出入带宽峰值计费。

3. 单击**确定**，完成新增通道。
4. 在“接入管理”页面中，查看通道列表信息。

- ID/通道名：通道的 ID 和名字，其中通道名可以修改。
- VIP：用于客户端访问的接入 IP 地址。
- 域名：用于客户端访问的接入域名（系统分配，且自动绑定到 VIP）。
- 状态：仅“运行中”状态下，加速通道才可以正常使用。

查看通道信息

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击指定通道的 **ID/通道名**，进入下一级页面。

2. 在“通道信息”标签页，可以查看通道的详细信息。其中，“转发 IP”是指加速通道末端的转发节点 IP，该转发节点负责将加速通道的数据通过公网转发给源站。如果您希望多条通道使用同一个域名，可单击**未关联**跳转至 统一域名 页面进行配置。


通道详情（zhuzi_test）
通道信息
TCP/UDP监听器管理
HTTP/HTTPS监听器管理
通道ID
通道名字
zhuzi_test
VIP
域名
qcloud.com
接入节点
上海（原中国大陆-华东大区）
接入节点网络类型
常规BGP
回源节点
美国西部硅谷
带宽上限
100 Mb
并发连接数
2 万
统一域名
未关联
转发IP
创建时间
2021/10/13 17:16:07
计费模式
按带宽计费
所属项目
默认项目
标签

TCP/UDP 监听器管理

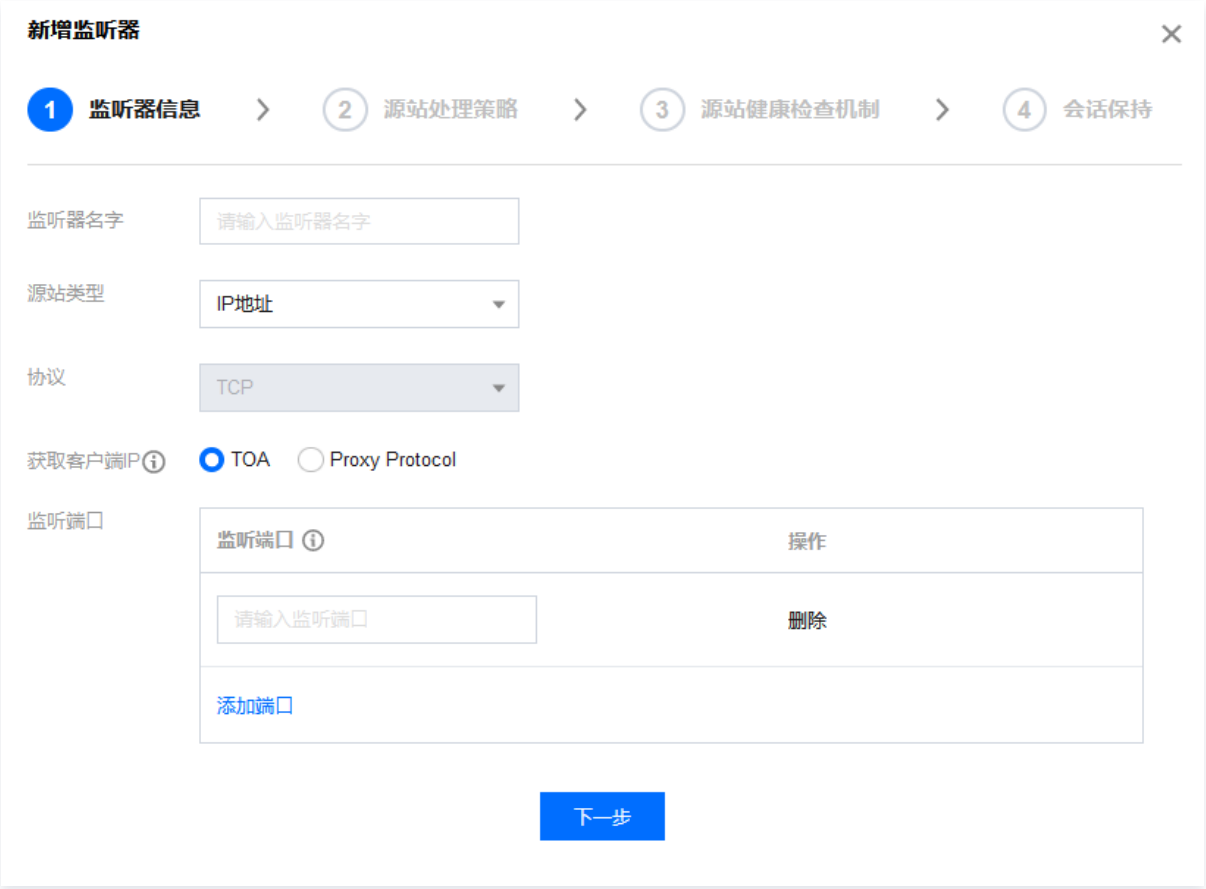
最近更新时间：2024-10-25 10:56:22

新增 TCP/UDP 监听器

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击指定通道的 ID/通道名。
2. 进入下一级页面，选择TCP /UDP 监听器管理 > 新建，具体配置如下：

2.1 配置监听器信息，用于设置加速协议和端口映射关系。

> ⚠ 注意：
> 会话保持功能尚未全量，如有需要，可通过 工单联系 我们。



- 源站类型：可以填写 IP 地址或域名，但同一个监听器只支持一种类型。（备注：IPv6通道暂不支持域名类型）
- 获取客户端IP：可选择 TOA 或 Proxy Protocol 两种获取客户端 IP 的方法获取真实用户 IP，具体介绍可参考 获取访问用户真实 IP 页面。
- 监听端口：指加速通道 VIP 的访问端口。端口规范：有效范围1 - 64999（21端口暂不开放）。支持单个端口或连续端口范围，但端口不能重复，一次最多添加的连续端口20个，如8000 - 8019。

2.2 配置源站处理策略。即在同一个监听器绑定多个源站的情况下，选择源站之间的调度策略。

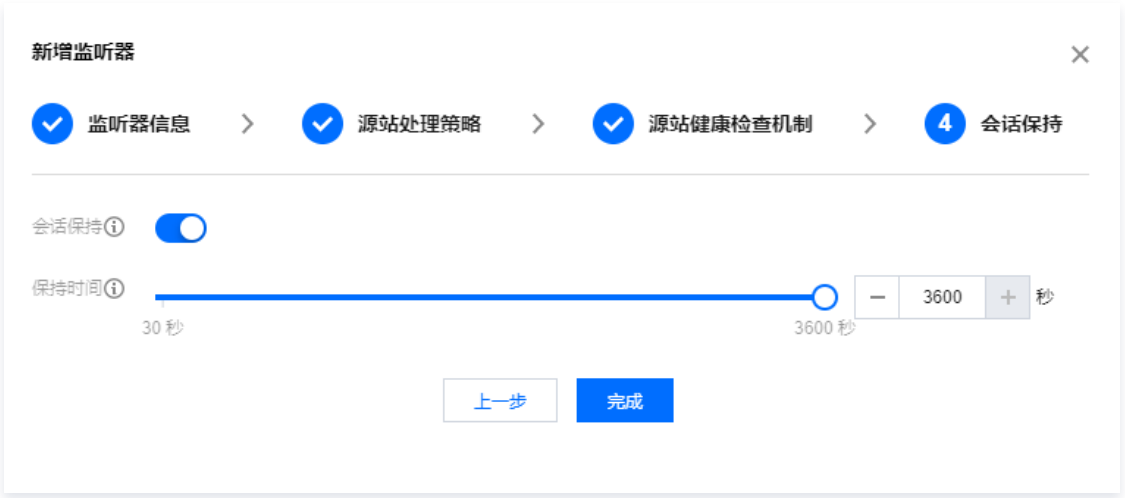
- 轮询：多个源站按轮询策略回源。
- 轮询加权：多个源站按权重比例回源（可以在绑定监听器时设置各源站的权重）。
- 最小连接数：在所有源站中选择连接数最小的源站优先进行调度。
- 最小时延：选择时延最小的源站优先进行调度。
- 备源：选择是否开启主备源切换（开启该功能会强制要求开启源站健康检查）。

> **注意：**
> 源站类型为域名的监听器，仅支持“轮询”及“最小连接数”两种调度策略，暂不支持备源。

2.3 如果使用 TCP 监听器，则可以选择配置健康检查机制，帮助您自动检查并移除异常的源站，启用备源则无法关闭健康检查。

- 响应超时时间：指源站响应的超时时间。
- 健康检查间隔：指前后两次健康检查的时间间隔。
- 不健康阈值：表示监听器连续检查失败多少次后确定源站不健康。当健康检查判断源站不健康时，将不再向该源站转发数据包，直至该源站健康检查状态恢复正常。
- 健康阈值：表示监听器连续检查成功多少次后确定源站健康。当健康检查判断源站健康时，将重新向该源站转发数据包。

2.4 选择是否开启会话保持功能。



- 会话保持：来自同一IP的用户请求保持访问相同源站。
- 保持时间：会话保持的时间。当监听器无请求的持续时间超过保持时间，将会自动断开会话保持。

3. 单击**完成**，成功新建 TCP/UDP 监听器。

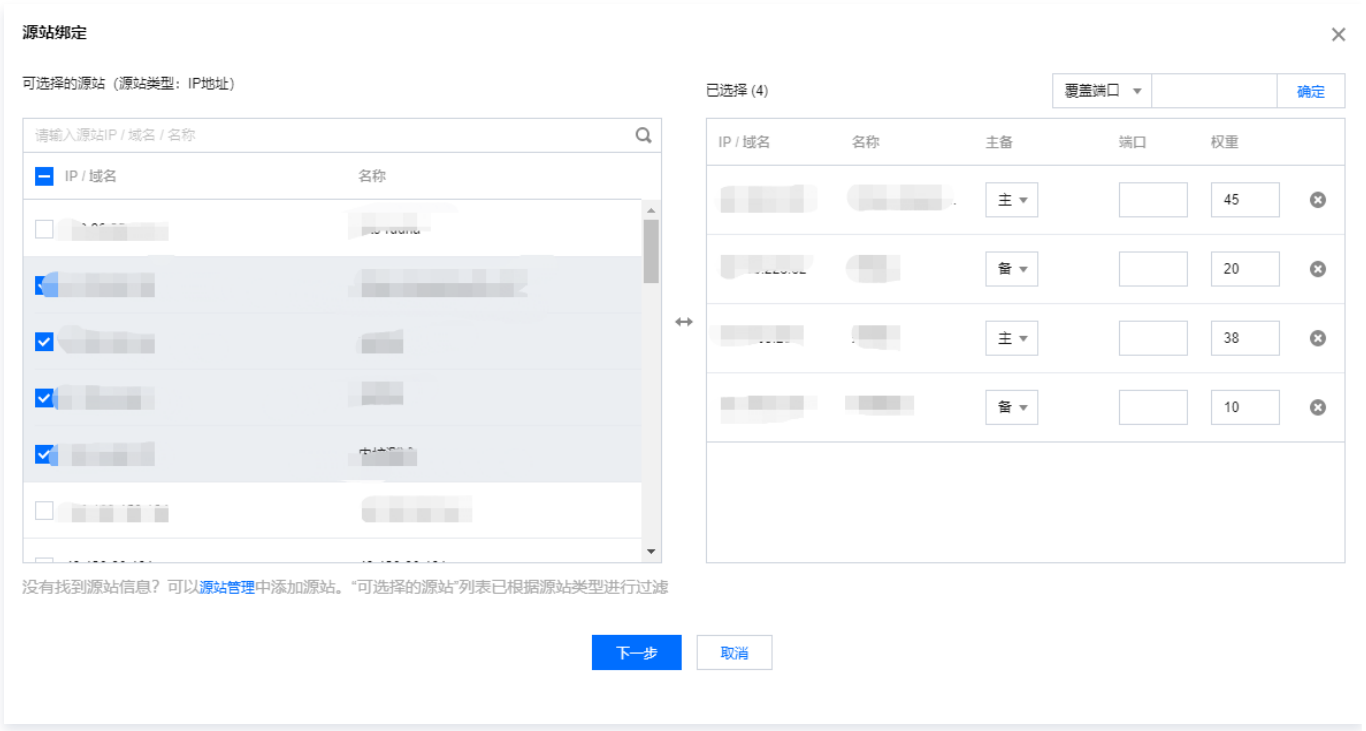
设置TCP/UDP 监听器

单击 **TCP/UDP 监听器管理**标签页，在操作栏单击**设置**可以修改监听器名字、调度策略和健康检查参数等。

源站绑定

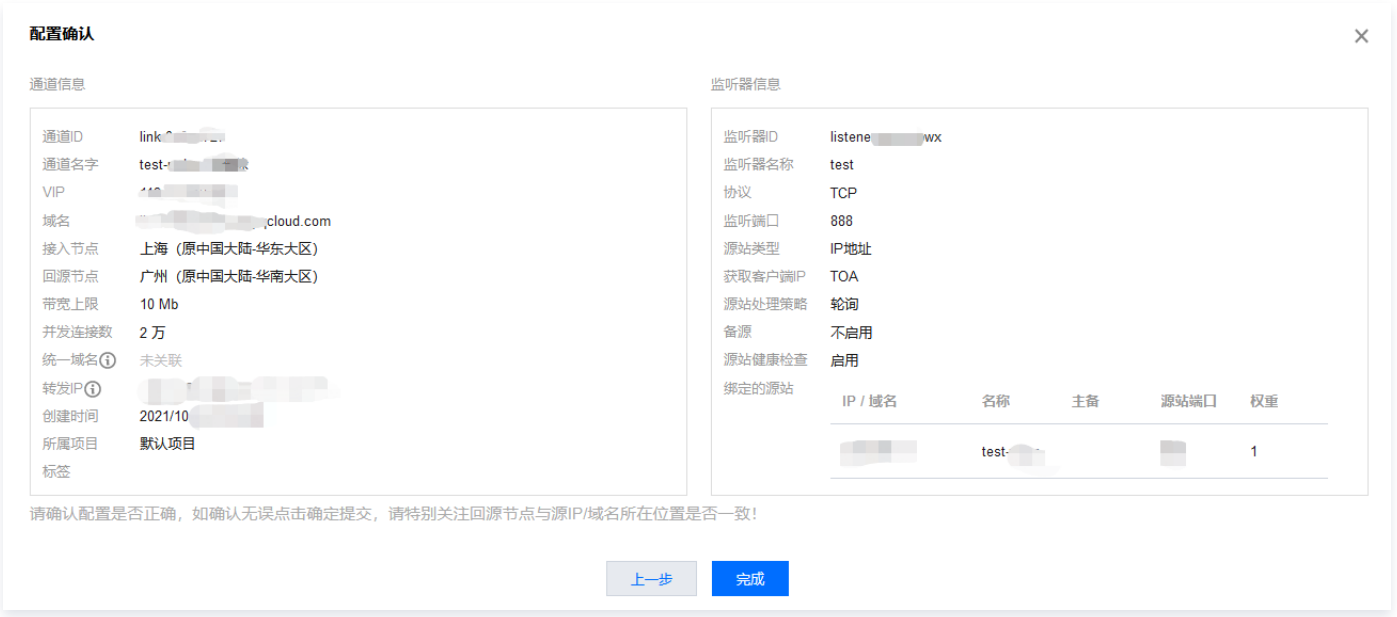
1. 单击 **TCP/UDP 监听器管理**标签页，选择已建立的 “TCP/UDP监听器” 。在操作栏单击**源站绑定**，您可选择多个源站进行绑定或解绑。如果控制台显示未找到源站信息，可能是由于源站类型不匹配或源站未添加在 源站管理 中。

2. 选择源站，并配置回源端口。
   - 如果监听器开启主备轮询选项，则需要在源站绑定页面设置“主源站”与“备源站”。
   - 如果您希望对多个源站的端口进行设定，可使用右上角“覆盖端口/补齐端口”功能。无论您之前设定的源站端口为多少，“覆盖端口”功能都会将您选择的目标源站统一设定为您输入的端口。如果选定的目标源站中存在未设定端口的情况，您可使用“补齐端口”功能进行统一设定，减少您的重复工作量。
   - 如果监听器策略为“加权轮询”，则可以在绑定的同时设置源站的权重，范围1 - 100，按权重值占总权重的比例进行调度，如源站1的权重为60，源站2的权重为80，则源站1的调度比例为 60 / (60 + 80) = 42.8%，源站2的调度比例为57.2%。



- 如果您开启了健康检查，绑定源站后，健康检查即开始启动。可以通过监听器的状态来判断源站是否正常，加速通道只会向正常状态的源站进行数据包转发，异常的源站将不再转发数据包，直至该异常源站健康检查状态恢复正常后才重新转发。
- 如果您未开启健康检查或使用UDP 协议监听器，那么不论源站的状态如何，加速通道将始终做数据包转发。

3. 配置确认
   源站配置完成后点击下一步到配置确认页面，用户可以查看当前配置的归属通道信息以及监听器详细信息。



4. 单击**完成**，成功绑定源站。

删除 TCP/UDP 监听器

单击 TCP/UDP 监听器管理标签页，在操作栏单击**删除**，可以删除指定的监听器，若监听器已绑定源站，则需要选中“允许强制删除绑定有源站的监听器”后，才能删除。删除后，该监听器的端口将停止加速。

HTTP/HTTPS 监听器管理

最近更新时间：2024-10-25 10:56:22

新增 HTTP/HTTPS 监听器

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击指定通道的 ID/通道名。
2. 进入到下一级页面，选择 **HTTP/HTTPS 监听器管理** > **新增**，可选的协议有 HTTP 和 HTTPS（备注：IPv6 通道当前不支持 HTTP/HTTPS 监听器配置）。
3. 具体配置如下：

3.1 当选中 HTTP 时，仅需要输入监听端口即可，监听器会默认按照 HTTP 协议进行转发。

3.2 当选中 HTTPS 时，则需要额外配置证书和其他信息，如下图：

- “监听器与源站之间使用 HTTP 协议”，指客户端到加速通道 VIP 之间使用 HTTPS 协议，而 VIP 到源站之间使用 HTTP 协议，需要源站开通 HTTP 协议端口。
  “监听器与源站之间使用 HTTPS 协议”，指客户端到源站之间全程使用 HTTPS 协议，需要源站开通 HTTPS 协议端口。
- SSL 解析方式：支持单向认证、双向认证。
- 服务器证书/客户端证书：需要在全球应用加速控制台的**证书管理**上传/更新，然后在新建/修改 HTTPS 监听器时从下拉列表中选择对应的证书，详见 证书管理 。

设置 HTTP/HTTPS 监听器

单击 **HTTP/HTTPS 监听器管理**标签页，在操作栏单击**设置规则**，可以进入下一级页面，进行域名和 URL 管理。

添加域名

> **注意：**
> 添加的域名需要提前备案，相关备案操作，您可参考 如何快速备案您的网站或 APP 。

1. 为HTTP监听器添加域名只需直接输入域名即可，但须符合域名格式要求，且只支持精确匹配。监听器支持的字符集有：`a-z、0-9、.、-`，**长度**`3 - 80` 。

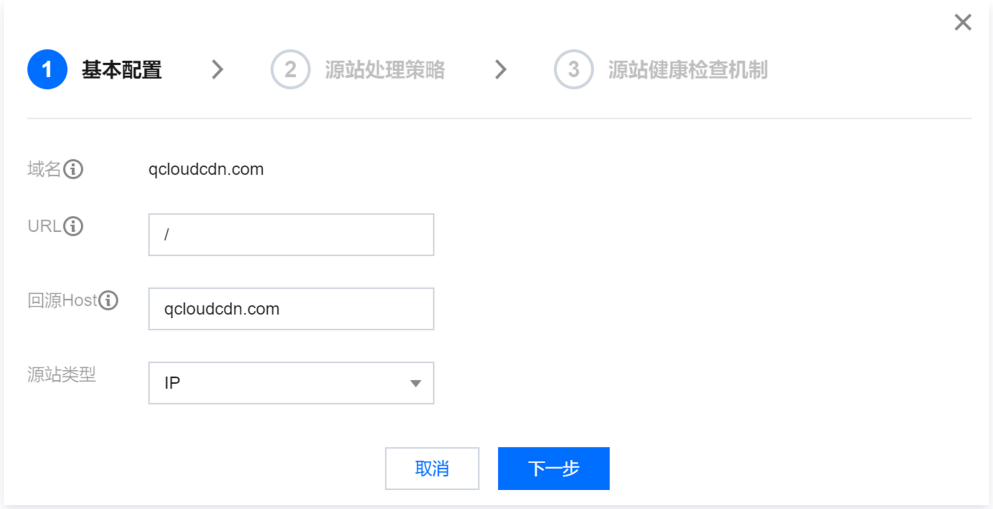
2. 为 HTTPS 监听器添加域名需输入域名并选择对应服务器证书

- 域名：需要符合域名的格式要求，只支持精确匹配，支持字符集如下，长度3-80个字符：a-z 0-9 . -
- 服务器证书：默认使用创建监听器时选择的证书。如您在此处重新上传证书，则该域名将使用新证书进行认证。
- HTTP3 传输：点击**开启**后，支持通过 HTTP3（QUIC）访问，若客户端不支持 HTTP3，规则自动降级为 HTTP2.0 及以下协议访问。

添加规则

完成“添加域名”操作后，单击**添加规则**，可以添加对应URL及选择源站类型。同一个域名下可以添加最多20条 URL 规则，具体如下：

1. 基本配置：

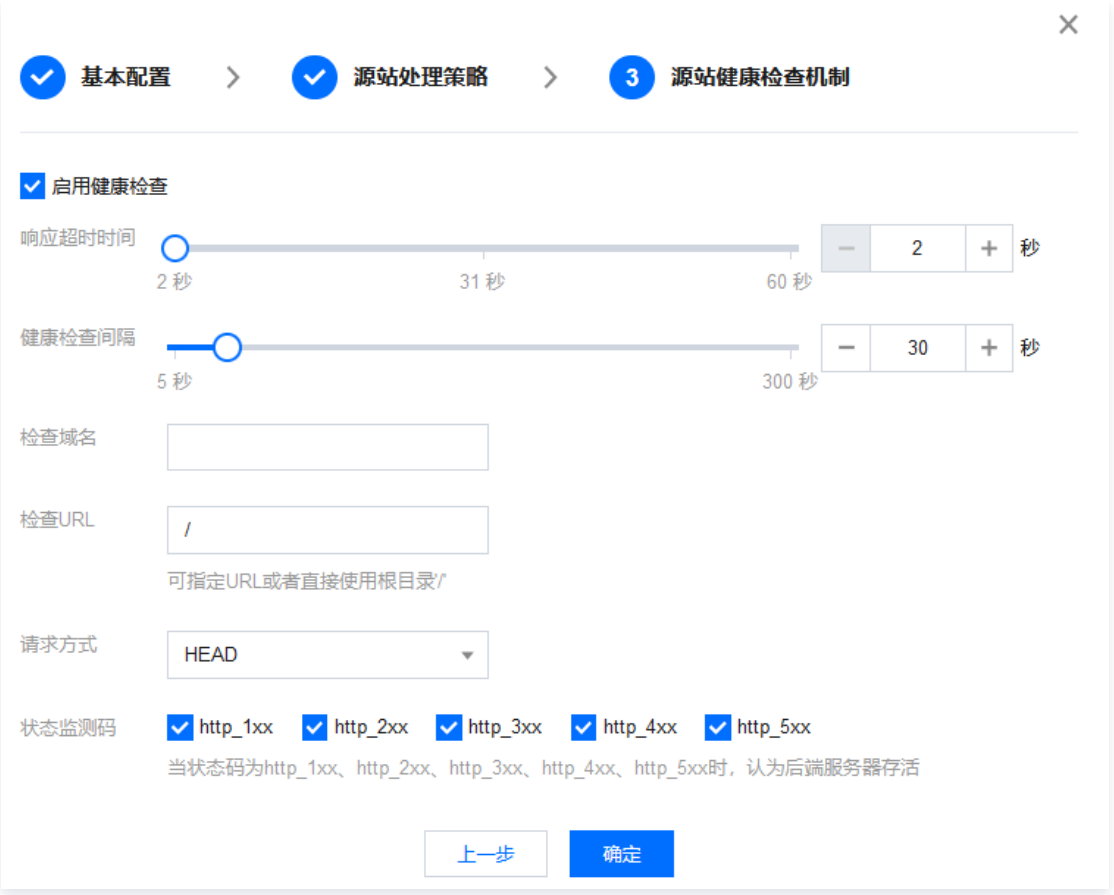


- URL：支持字符集如下，`a-z、A-Z、0-9、_ 、.、- 、/`，长度`1 - 80`。
- 回源 Host：支持修改回源请求中的 HOST 字段。
- 源站类型：支持 IP 和域名两种类型，但同一个监听器只支持一种类型。

2. 源站处理策略：
设置源站的转发处理规则，即在同一个监听器绑定多个源站的情况下，选择源站之间的调度策略。

- 轮询：多个源站按轮询策略回源。
- 轮询加权：多个源站按权重比例回源（源站类型为域名时不支持配置）。
- 最小连接数：在所有源站中选择连接数最小的源站优先进行调度。
- 回源 SNI：与源站建立SSL连接之前先发送 SNI，源站根据 SNI 值返回对应的证书。

3. 源站健康检查机制：
您可以选择针对当前域名启用监控检查机制。可以设置独立的检查 URL，请求方式可以支持 HEAD 及 GET，检查状态码可支持 http_1xx，http_2xx，http_3xx，http_4xx，http_5xx，状态码可单选也可多选，即当检测到指定的状态码时，监听器认为后端源站属于正常状态。如果未检测到任何状态码时，监听器认为后端源站异常。



修改域名

完成“添加域名”操作后，单击修改域名，可以对域名进行修改。

删除域名

完成“添加域名”操作后，单击删除可删除域名。如果域名下已有规则绑定源站，则需要勾选“强制删除绑定有源站的规则”。

HTTP3 配置

支持变更对应域名是否开启 HTTP3 传输（当前仅支持在 HTTPS 监听器配置，且创建通道时 HTTP3 特性状态为开启）。

修改规则

参考上文添加规则，主要差别在于域名和源站类型无法修改。

绑定源站

详情请参见绑定源站，可以对不同源站绑定不同的端口。有关“覆盖端口”及“补齐端口”功能，请参见TCP/UDP 监听器源站绑定。

> **注意：**
> 一个规则绑定的源站总数最多为100个。

删除规则

完成“添加规则”操作后，单击**删除**，可删除规则，如果规则下有绑定的源站，需要先勾选“强制删除绑定有源站的规则”。

配置回源请求头

1. 完成“添加规则”操作后，在规则的操作栏选择**更多**，单击**配置回源请求头**。

2. 单击**新增参数**，添加请求头的名称参数及取值；如需要携带用户真实IP的头部，其变量值为 $remote_addr（默认已经有 X-Forwarded-For 头部携带客户 IP 回源），携带用户真实端口的变量值为 $remote_por；其余带$变量默认不支持，如有需求，可 提交工单 联系我们。

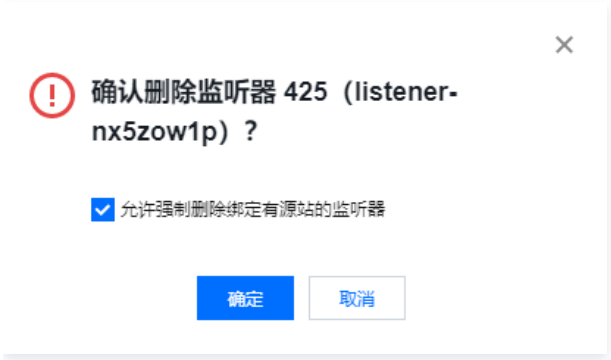
> **注意：**
> 1. HTTP 头部的名称 Key 值长度默认为1 - 100个字符，由数字0 - 9、字符a - z、A - Z，及特殊符 - _ : 空格 组成。Value 长度为1 - 100个字符，不支持中文；
> 2. 每条规则最多可配置10条回源 HTTP 请求头；
> 3. 部分标准头部不支持自助设置/增加/删除，具体清单请参见以下列表。

www-authenticate	authorization	proxy-authenticate	proxy-authorization
age	cache-control	clear-site-data	expires
pragma	warning	accept-ch	accept-ch-lifetime
early-data	content-dpr	dpr	device-memory
save-data	viewport-width	width	last-modified
etag	if-match	if-none-match	if-modified-since
if-unmodified-since	vary	connection	keep-alive
accept	accept-charset	expect	max-forwards
access-control-allow-origin	access-control-max-age	access-control-allow-headers	access-control-allow-methods
access-control-expose-headers	access-control-allow-credentials	access-control-request-headers	access-control-request-method
origin	timing-allow-origin	dnt	tk
content-disposition	content-length	content-type	content-encoding
content-language	content-location	forwarded	x-forwarded-host
x-forwarded-proto	via	from	host
referer-policy	allow	server	accept-ranges
range	if-range	content-range	cross-origin-embedder-policy
cross-origin-opener-policy	cross-origin-resource-policy	content-security-policy	content-security-policy-report-only
expect-ct	feature-policy	strict-transport-security	upgrade-insecure-requests
x-content-type-options	x-download-options	x-frame-options(xfo)	x-permitted-cross-domain-policies
x-powered-by	x-xss-protection	public-key-pins	public-key-pins-report-only
sec-fetch-site	sec-fetch-mode	sec-fetch-user	sec-fetch-dest
last-event-id	nel	ping-from	ping-to
report-to	transfer-encoding	te	trailer
sec-websocket-key	sec-websocket-extensions	sec-websocket-accept	sec-websocket-protocol
sec-websocket-version	accept-push-policy	accept-signature	alt-svc

date	large-allocation	link	push-policy
retry-after	signature	signed-headers	server-timing
service-worker-allowed	sourcemap	upgrade	x-dns-prefetch-control
x-firefox-spdy	x-pingback	x-requested-with	x-robots-tag
x-ua-compatible	max-age		

删除 HTTP/HTTPS 监听器

单击 HTTP/HTTPS 监听器管理标签页，在操作栏单击**删除**，可以删除指定的监听器，若监听器已绑定源站，则需要选中“允许强制删除绑定有源站的监听器”后，才能删除。删除后，该监听器的端口将停止加速。



TLS 版本及密码套件说明

最近更新时间：2024-10-25 10:56:22

本文介绍了 GAAP 对 TLS 握手时允许使用的协议版本和密码套件的支持情况。

什么是 TLS 协议版本?

TLS（Transport Layer Security）协议是一种用于加密网络通信的安全协议，它是 SSL（Secure Sockets Layer）协议的继任者，允许客户端/服务器应用程序之间进行加密通信。 TLS 协议有多个版本，包括 TLS 1、TLS 1.1、TLS 1.2 和 TLS 1.3，TLS 1.3是最新的版本，提供了更安全、更高效的加密机制。

什么是密码套件?

密码套件是一组加密算法，用于安全传输层协议（TLS）中的安全连接。TLS 密码套件由认证，加密和消息认证码（MAC）三个部分组成，它们提供安全性和可靠性，保护传输中的数据免受第三方窃取。在 TLS 握手过程中，客户端和服务器会协商一个可以使用的密码套件（客户端和服务器会根据它们支持的密码套件列表来确定使用哪个密码套件），以便客户端和服务器之间的通信可以使用该密码套件进行加密。

使用场景

GAAP 默认启用所有 TLS 版本，密码套件为 GAAP_TLS_CIPHERS_WIDE，可以满足大部分客户需求，若您对安全性有更高要求，可自定义调整使用的密码套件包：

业务场景	密码套件
注重兼容旧版浏览器，对安全性要求可适当放宽。	GAAP_TLS_CIPHERS_WIDE
需兼顾浏览器的兼容性和安全性，安全性和兼容性均为适中	GAAP_TLS_CIPHERS_GENERAL
安全性要求高，可降低浏览器兼容性，需屏蔽所有可能存在安全漏洞的 TLS 版本和密码套件	GAAP_TLS_CIPHERS_STRICT

GAAP 支持的 TLS 协议版本及密码套件

GAAP 支持的 TLS 版本如下：

TLS 1、TLS 1.1、TLS 1.2、TLS 1.3

GAAP 支持在 TLS 协议版本的基础之上，为用户提供不同的强度的密码套件选择：

- GAAP_TLS_CIPHERS_STRICT：安全性要求高，禁用所有不安全的密码套件。
- GAAP_TLS_CIPHERS_GENERAL：需兼顾浏览器的兼容性和安全性，安全性和兼容性均为适中。
- GAAP_TLS_CIPHERS_WIDE（默认）：注重兼容旧版浏览器，对安全性要求可适当放宽。

OpenSSL 密码套件	GAAP_TLS_CIPHERS_STRICT	GAAP_TLS_CIPHERS_GENERAL	GAAP_TLS_CIPHERS_WIDE
HIGH（nginx预定义高强度密码套件）	✓	✓	✓
MEDIUM（nginx预定义中等强度密码套件）	-	✓	✓
DEFAULT（nginx预定义默认强度密码套件）	-	-	✓
ECDHE-ECDSA-AES256-GCM-SHA384	✓	✓	✓
ECDHE-ECDSA-AES128-GCM-SHA256	✓	✓	✓
ECDHE-RSA-AES256-GCM-SHA384	✓	✓	✓

ECDHE-RSA-AES128-GCM-SHA256	✓	✓	✓
ECDHE-ECDSA-CHACHA20-POLY1305	✓	✓	✓
ECDHE-RSA-CHACHA20-POLY1305	✓	✓	✓
ECDHE-RSA-AES256-SHA	-	-	✓
ECDHE-RSA-AES128-SHA	-	-	✓
AES256-GCM-SHA384	-	-	✓
AES128-GCM-SHA256	-	-	✓
AES256-SHA	-	-	✓
AES128-SHA	-	-	✓

您可根据自身业务的安全和兼容性需求配置 TLS 版本及密码套件，最终支持的 OpenSSL 密码套件取 TLS 版本和密码套件选项对应内容的交集，例如：TLS 版本开启 TLS 1.2，且密码套件选项选择GAAP_TLS_CIPHERS_WIDE，则最终支持的 OpenSSL 密码套件为 TLS 1.2 与 GAAP_TLS_CIPHERS_WIDE 支持的交集：ECDHE-ECDSA-AES256-GCM-SHA384 等密码套件包。

安全防护

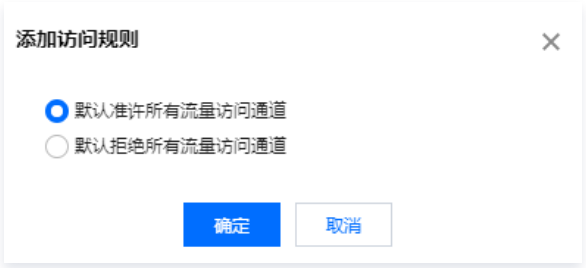
最近更新时间：2024-09-13 14:55:51

GAAP 默认提供基础安全防护策略（普通用户是 2Gbps 带宽，VIP 用户是 10Gbps 带宽）。如需升级到高级防护策略，可在 DDoS 高防包控制台-云资产升级防护。

另外 GAAP 控制台提供安全防护可支持配置黑白名单。详细配置方法如下：

1. 登录 全球应用加速控制台，进入“接入管理”页面，单击指定通道的 **ID/通道名**。
2. 进入到下一级页面，选择**安全防护** > **添加规则**，可进入向导，具体配置如下：

2.1 添加访问规则，选择默认准许/拒绝所有流量访问通道。



2.2 添加来源IP，选择协议并添加协议端口。之后选择“允许”/“拒绝”该IP的访问。

> **说明：**
> 可添加的访问规则最多为100个。

3. 单击**确定**。

通道组管理

最近更新时间：2024-12-02 14:54:03

新增通道组

当用户需要加速多个区域，源站区域相同，且监听器配置相同时，可以通过通道组实现批量配置管理，减少管理单通道时的重复工作。

1. 登录 全球应用加速控制台，进入“通道组管理”页面，单击**新增**。
2. 在弹出的窗口中，填写通道组信息。

- 项目：该通道组所属项目（后续可以更换项目）。
- 通道组名称：最多30个字符，支持中文。
- IP 版本：可根据需要选择 IPv4 或 IPv6，其中 IPv6 暂时只支持国内接入节点。
- HTTP3 特性：启用后，通道支持HTTP3（QUIC）协议传输，仅支持配置HTTP/HTTPS监听器（通道组创建成功后，暂不支持变更**启用/关闭**）。
- 接入节点：选择客户端所在区域或距离客户端最近区域的节点，支持多选。

  注意：
  - 若您需要提供中国香港的精品 BGP 网络接入，请在“接入节点”选择“香港”，选择“精品 BGP”。
  - 中国大陆提供三网节点网络，如有需要，可 提交工单 联系我们。

- 源站区域：选择目标服务器所在区域或距离目标服务器最近区域的节点。

  注意：
  中国台湾无法与中国大陆进行直连。

- 通道规格：选择各通道的带宽上限及最大并发数。
- 带宽上限：通道的带宽上限，最大值10000Mbps（部分通道最大值为1000Mbps）。

- 并发上限：通道的最大并发连接数，最大值100万（部分通道最大值为30万）。

> ⚠ **注意：**
> 一个通道组下的通道数量不能超过20个。

- 标签：非必填项，可通过设置标签实现对通道的分类管理；
- 费用：根据您选择的带宽与并发数，下方会给出相应的通道费和带宽费。
  a. 通道费：按日计算，直到通道被删除为止，请特别注意通道创建后未满一天删除也会按一天计费；
  b. 带宽费：按每日实际出入带宽峰值计费。

3. 单击**确定**，完成新增通道组。
4. 在 通道组管理 页面中，查看通道组列表信息，可根据实际需求，选择对同一通道组下的不同通道进行管理，监控不同通道的实时运行状态。

- ID/通道组名：通道的 ID 和名字，其中通道名可以修改。
- VIP：用于客户端访问的接入 IP 地址。
- 域名：用于客户端访问的接入域名（系统分配，且自动绑定到 VIP）。
- 状态：仅“运行中”状态下，加速通道才可以正常使用。

查看通道组信息

1. 登录 全球应用加速控制台，进入“通道组管理”页面，单击指定通道组的 **ID/通道名**，进入下一级页面。

2. 在“通道组信息”标签页，可以查看各通道的详细信息。其中，“转发 IP”是指加速通道末端的转发节点 IP，该转发节点负责将加速通道的数据通过公网转发给源站。如果您希望多条通道使用同一个域名，可单击**统一域名**选项，可直接跳转至“统一域名”页面进行配置，通道组下不同通道可单独对**统一域名**进行配置。

TCP/UDP 监听器管理

新增 TCP/UDP 监听器

详情请见 接入管理相关配置页面 。

设置 TCP/UDP 监听器

详情请见 接入管理相关配置页面 。

HTTP/HTTPS 监听器管理

新增 HTTP/HTTPS 监听器

详情请见 接入管理相关配置页面 。

设置 HTTP/HTTPS 监听器

详情请见 接入管理相关配置页面 。

安全防护

详情请见 接入管理相关配置页面 。

访问加速通道

最近更新时间：2024-09-13 17:24:01

TCP/UDP 协议

您可以通过以下三种方法访问加速通道：

- 客户端直接访问加速通道 VIP+端口。
- 客户端访问加速通道域名+端口。
- 若客户端原来访问的是域名，可以将该域名 CNAME 到加速通道的域名，或者修改客户端本地 host，将原来访问的域名解析到加速通道 VIP。源站如果需要获取客户端真实 IP（仅 TCP 协议），需要安装 TOA 模块，具体可参见 服务端获取客户端真实 IP（仅针对 TCP 协议）。

HTTP/HTTPS 协议

将客户端访问的域名 CNAME 到加速通道的接入域名，或者修改客户端本地 host，将客户端要访问的域名解析到加速通道 VIP，然后客户端按照协议+URL访问即可实现加速。
源站可以直接从 HTTP 请求头中 X-Forwarded-For 字段中获取客户端真实 IP。

基础DDoS防护

最近更新时间：2024-08-30 17:55:51

DDoS 攻击，是一种通过向目标系统发送大量的网络流量，导致目标系统无法提供正常服务的攻击方式。腾讯云默认为全部 GAAP 加速通道实例提供免费基础DDoS 防护能力，有效阻止业务遭受恶意攻击，提高产品安全性。同时，您可通过绑定 DDoS 高防包的方式来获取更大防护能力。

DDoS 基础防护工作原理

防护能力

全部客户默认享受最高不超过2Gbps的 DDoS 防护能力。但当客户业务遭受到频繁的攻击时，腾讯云会根据客户的历史攻击情况调整其基础 DDoS 防护能力，以保障腾讯云平台的整体稳定。

> **说明：**
> 基础版 DDoS 防护无需配置，通道实例创建后默认享受免费基础防护能力。

防护原理

腾讯云 DDoS 防护网络会实时监控访问全球应用加速实例的流量。当监测到包括 DDoS 攻击在内的异常流量时，在不影响正常业务的前提下，DDoS 基础防护会将可疑流量从原始网络路径中重定向到清洗设备上，清洗掉攻击流量，这一过程，就是 DDoS 防护。更多关于 DDoS 基础防护相关信息，可参考 DDoS 防护解决方案对比 。

查看全球加速通道实例的防护能力

您可通过 DDoS 安全控制台查看相关云资源当前防护能力：

1. 登录 DDoS 防护控制台 。
2. 单击**云资产列表**，进入资产列表详情页面。
3. 在资产列表页单击**GAAP**页签。
4. 根据需求查看对应通道实例的防护能力数值。

提升全球加速通道实例的防护能力

您可通过接入基础防护加强版，或接入高防包服务来提升对应通道实例的安全防护能力，完成接入后将全力提升您的防护能力，且会拥有一定的自助解封机会，将攻击影响降到最低。

接入高防包

1. 登录 DDoS防护控制台 。
2. 单击**云资产列表**，进入资产列表详情页面。
3. 在资产列表页单击 **GAAP** 页签。
4. 单击对应通道实例右侧的**接入防护**。
5. 在 DDoS 防护接入界面选择对应高防包进行接入。

防护类型	基础防护加强版	DDoS 高防包
攻击防护	将基础防护能力拓展至不超过 10 Gbps，可有效应对部分小流量攻击场景	全力防护，抵御三/四层网

防护特性	依托腾讯云强大的云上自研防护集群第一时间发现攻击流量，秒级开启防护	全面覆盖全国各地大小供应商，平均防护延迟低于30ms
自助解封次数	服务开通即可解封当前被封堵的实例（仅1次机会）	每日3次
应用场景	可有效应对部分小流量攻击场景，适用于作为日常的 DDoS 攻击防护	客户服务器封堵紧急恢复场景

告警阈值

DDoS 攻击告警阈值可以根据需求自定义告警策略。

配置告警阈值

1. 登录 DDoS 防护控制台 。
2. 单击**云资产列表**，进入资产列表详情页面。
3. 在资产列表页单击**GAAP**页签。
4. 单击**设置告警阈值**，可以根据需求自定义告警策略，单击**确定**。

默认告警阈值

当攻击流量满足以下其中一条，即发起告警。

- DDoS 攻击：当攻击入流量 >= 2Gbps或者清洗流量 > 100MB。任一条件满足，即发起告警；
- CC 攻击：当攻击入流量 > 1000qps

查看攻击态势

1. 登录 DDoS防护控制台 。
2. 单击**云资产列表**，进入资产列表详情页面。
3. 在资产列表页单击 **GAAP** 页签。
4. 单击对应通道右侧的**攻击分析**，页面跳转至防护概览（总览）页面，查看攻击态势。

相关文档

- 更多 DDoS 相关解决方案，请参考 DDoS 防护解决方案对比 。
- 配置 DDoS 高防 IP 实现业务防护，请参考 网站业务接入 及 非网站业务接入 。

统计数据

最近更新时间：2024-10-25 10:56:22

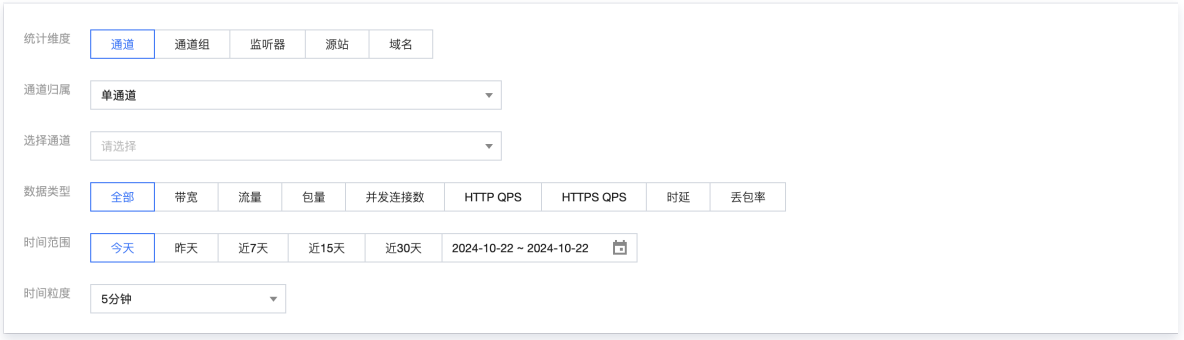
登录 全球应用加速控制台，进入“统计数据”页面。
该页面共提供了以下数据的筛选维度，分别为[通道]、[通道组]、[监听器]、[源站]、[域名]。

> **注意：**
> 域名的统计数据尚未全量，如有需要，可通过 工单联系 我们。

通道

查看通道维度的统计数据，如下图所示：

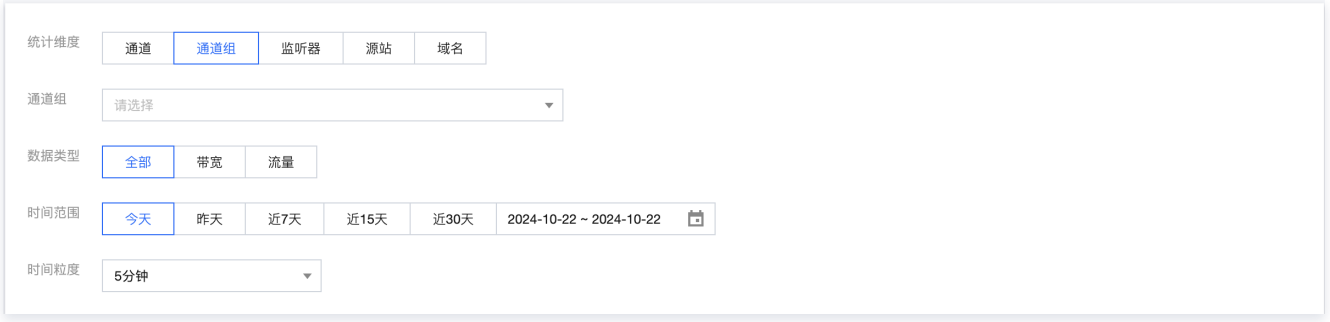
- 通道归属：默认选择为单通道，也可以选择已创建的通道组。
- 选择通道：选择**接入管理**中的通道，或者通道组内的通道。
- 数据类型：全部、带宽、流量、包量、并发连接数、HTTP QPS、HTTPS QPS、时延、丢包率。
- 时间范围：选择时间范围。
- 时间粒度：选择数据统计粒度，支持1分钟、5分钟、1小时和1天。
  [选择1分钟粒度，最长可查看1天的数据；选择5分钟粒度，最长可查看3天的数据；选择1小时粒度，最长可查看15天数据；选择1天粒度，最长可查看186天数据]



通道组

查看通道组维度的统计数据，如下图所示：

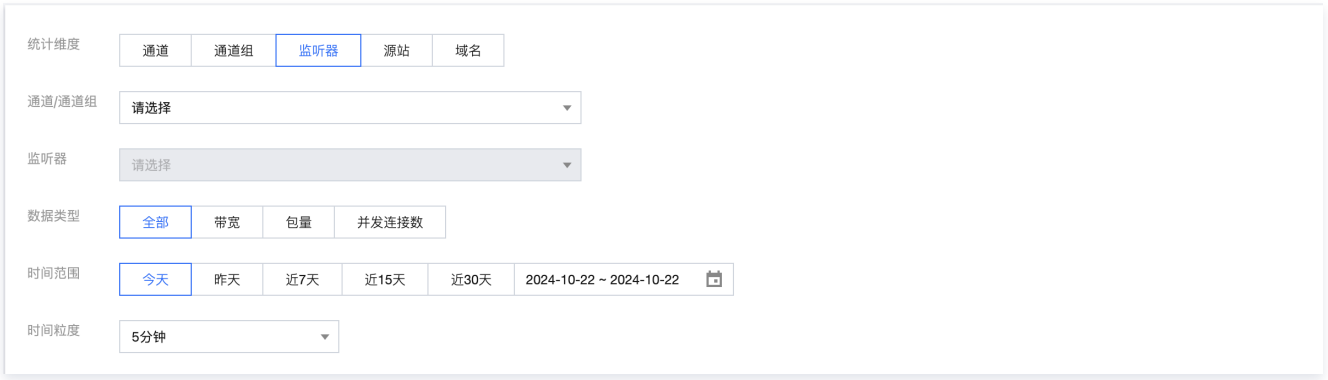
- 通道组：选择一条或多条通道组。
- 数据类型：全部、带宽、流量。
- 时间范围：选择时间范围。
- 时间粒度：选择数据统计粒度，支持1分钟、5分钟、1小时和1天。
  [选择1分钟粒度，最长可查看1天的数据；选择5分钟粒度，最长可查看3天的数据；选择1小时粒度，最长可查看15天数据；选择1天粒度，最长可查看186天数据]



监听器

查看监听器维度的统计数据，如下图所示：

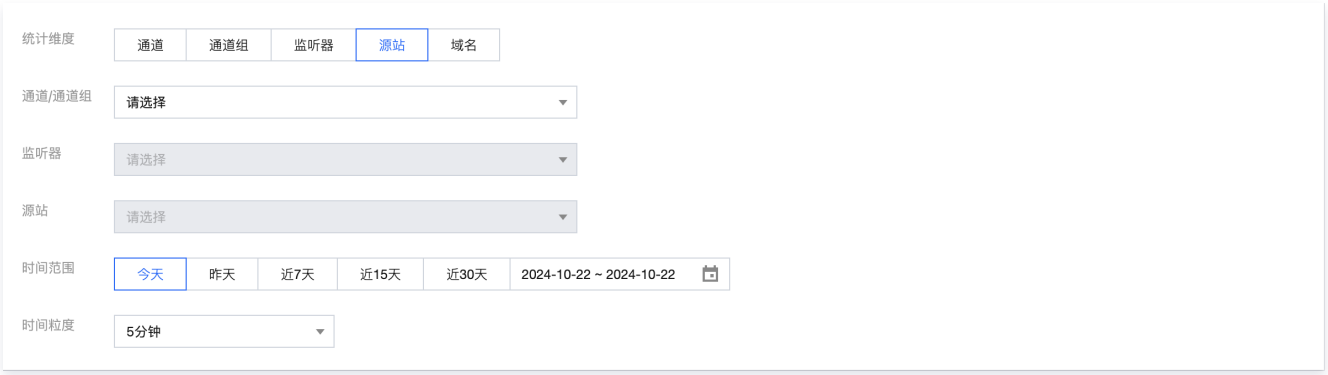
- 通道/通道组：选择监听器所归属的通道/通道组。
- 监听器： 选择监听器。
- 数据类型：全部、带宽、包量、并发连接数。
- 时间范围：选择时间范围。
- 时间粒度：选择数据统计粒度，支持1分钟、5分钟、1小时和1天。
  [选择1分钟粒度，最长可查看1天的数据；选择5分钟粒度，最长可查看3天的数据；选择1小时粒度，最长可查看15天数据；选择1天粒度，最长可查看186天数据]



源站

查看通道已绑定的源站中，健康状态的统计数据，如下图所示：

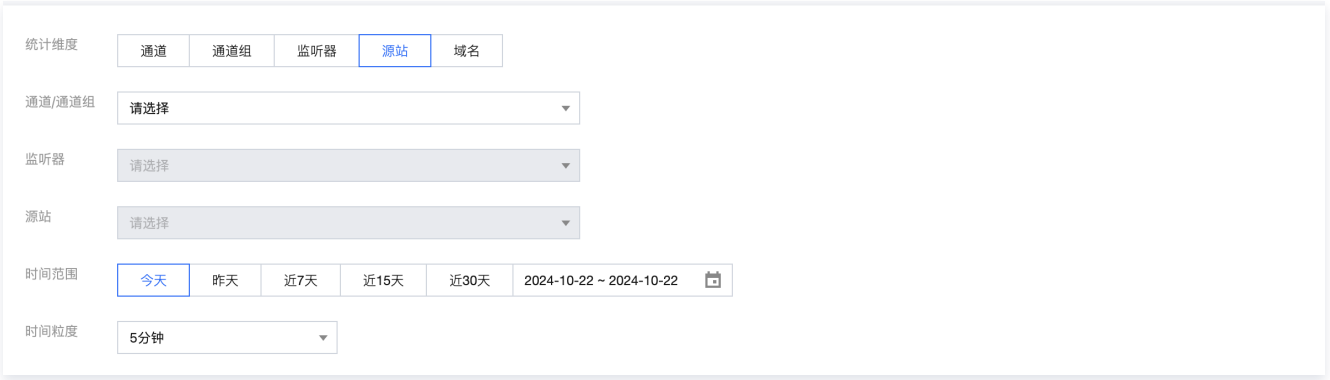
- 通道/通道组：选择源站所归属的通道/通道组。
- 监听器： 选择源站所归属的监听器。
- 源站： 选择源站。
- 时间范围：选择时间范围。
- 时间粒度：选择数据统计粒度，支持1分钟和5分钟。
  [选择1分钟粒度，最长可查看1天的数据；选择5分钟粒度，最长可查看31天的数据]



域名

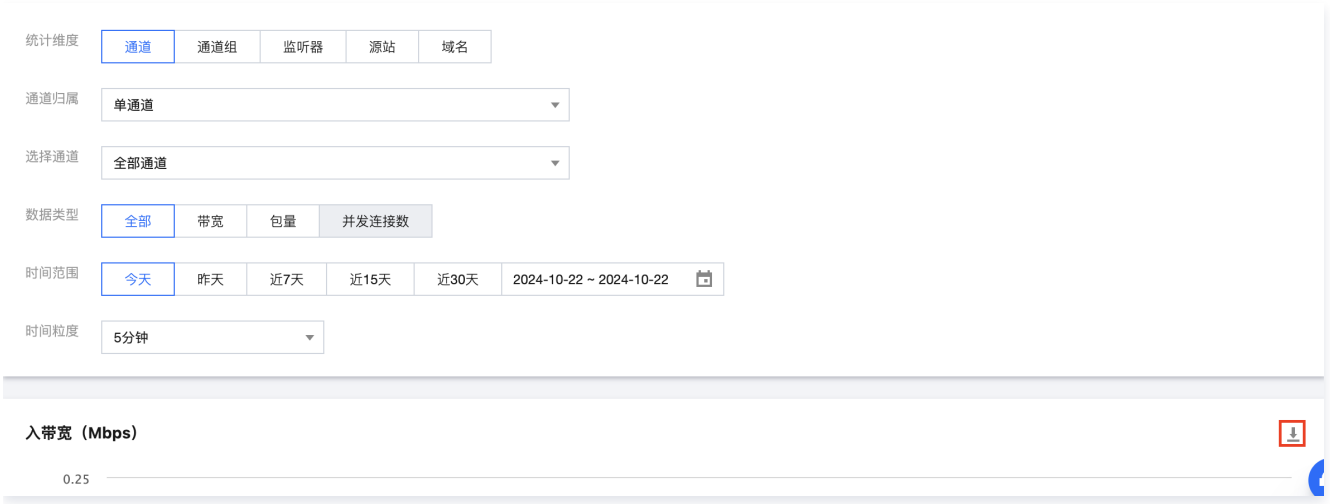
查看HTTP/HTTPS监听器配置中，监听域名的统计数据，如下图所示：

- 加速区域：选择中国境内、中国境外。
- 统计域名：支持单选、多选域名。
- HTTP协议：支持单选、全选。
- 数据类型：全部、请求量、状态码、Top10 URL。
- 时间范围：选择时间范围。
- 时间粒度：选择数据统计粒度，支持1分钟、5分钟、1小时和1天。
  [选择1分钟粒度，最长可查看1天的数据；选择5分钟粒度，最长可查看3天的数据；选择1小时粒度，最长可查看15天数据；选择1天粒度，最长可查看31天数据]



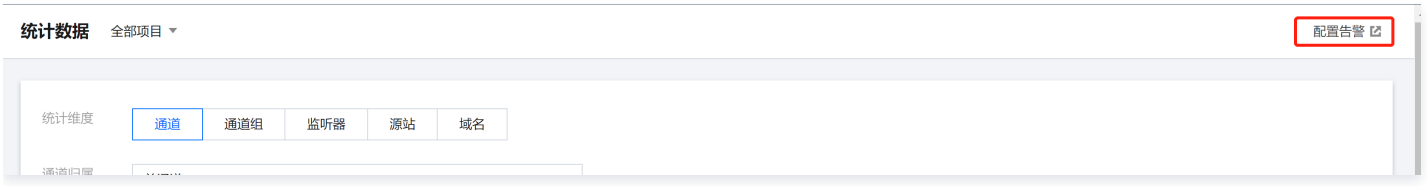
数据导出

进入“统计数据”页面，点击数据展示页右上角的下载图标，即可导出数据。



配置告警

进入“统计数据”页面，点击右上角“配置告警”，即可进行数据报警配置，具体操作流程详见 接入腾讯云可观测平台 。



接入腾讯云可观测平台

最近更新时间：2024-10-28 16:46:21

应用场景

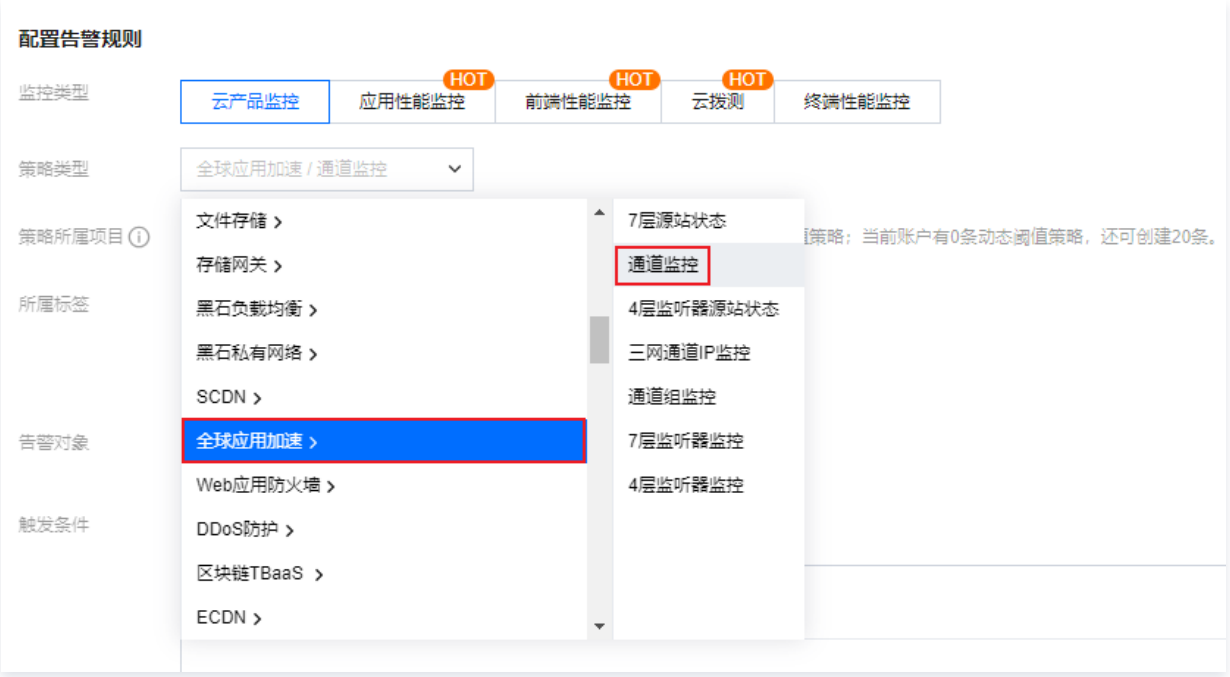
为提升使用体验，您可以在腾讯云可观测平台产品中配置对应的告警规则，当加速通道达到告警条件时，第一时间触发告警。

操作步骤

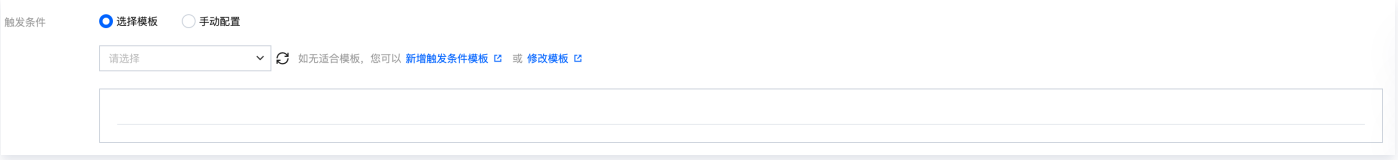
登录 腾讯云可观测平台控制台，进行如下操作。

通道监控

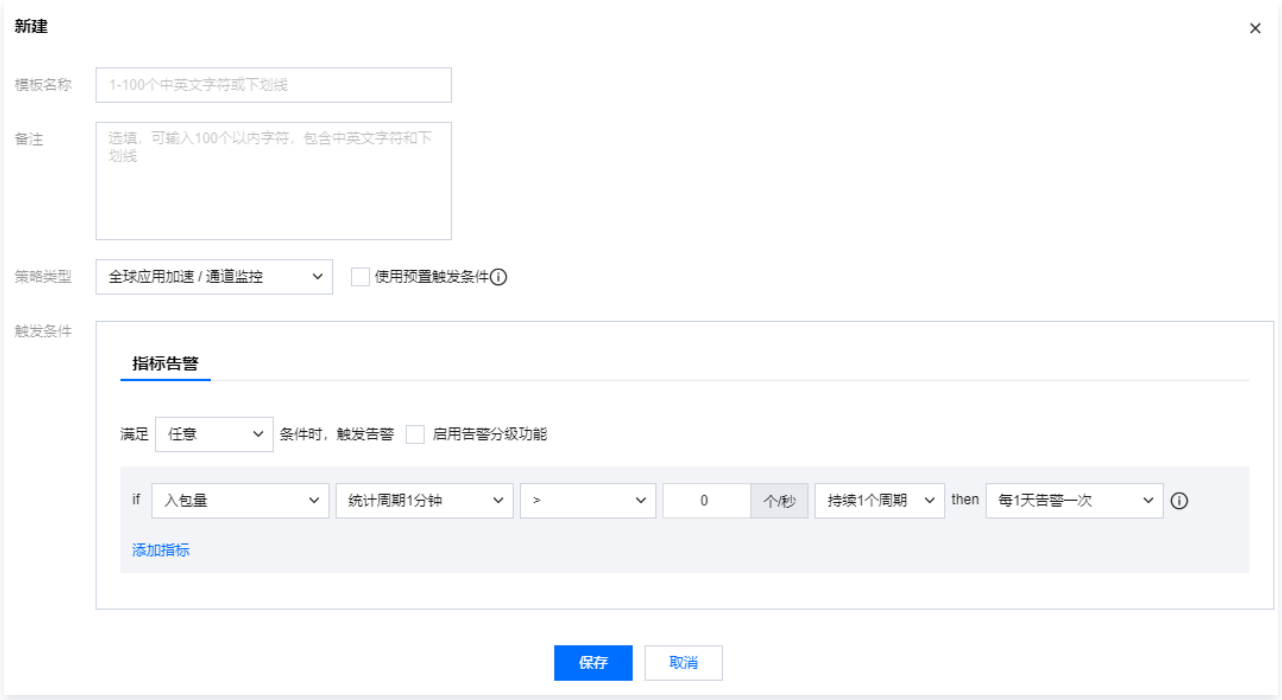
1. 在左侧目录中，选择“告警管理 > 告警配置 > 告警策略”，单击**新建策略**，进入新建策略页面。
2. 在“策略类型”中，选择**全球应用加速 > 通道监控**。



3. 在**配置告警规则**模块中，选择**告警对象**，您可根据需要添加对象进行监控。
   在“触发条件”中，您可选择“使用模板”或进行“手动配置”。
   如选择“使用模板”，您可使用之前已经配置的告警策略。如无合适模板，您可通过新建模板的方式进行配置。该模板会存储在控制台中，方便您的后续使用。
4. 新建模板具体配置过程：**触发条件**选择**选择模板**。单击**新增触发条件模板**，进入模板配置页面。

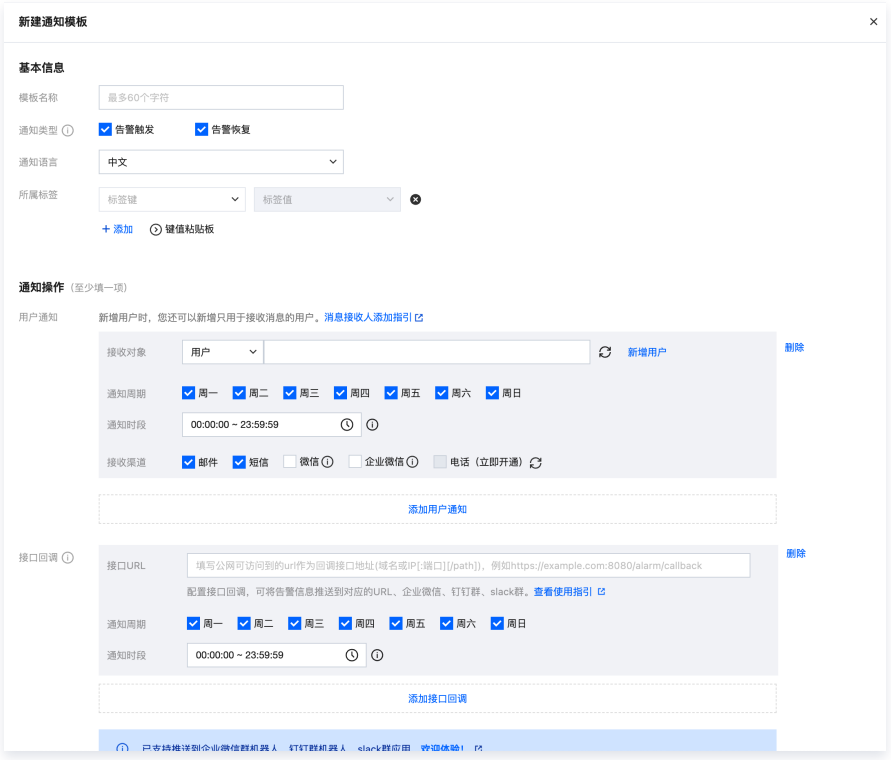


5. 单击**新建触发条件模板**，在弹出的窗口中配置触发条件，条件说明如下。
   - 模板名称：输入模板名称。
   - 备注：输入模板备注。
   - 策略类型：选择监控的服务，如**全球应用加速 > 通道监控**。
   - 使用预置触发条件：腾讯云可观测平台内置对应监控项的触发条件，勾选则开启。
   - 触发条件：分为指标告警和事件告警。在其下方单击添加，可以设置多个告警项。
     如选择使用“手动配置”，您可根据需要添加多个告警触发条件。



6. 在下一步**配置告警通知**中，单击**新建模板**，添加通知模板名称并选择接收对象及渠道。

> **注意：**
> 接收对象需绑定对应渠道，否则将无法收到告警通知。



根据业务场景需要，完成通知模板即可。

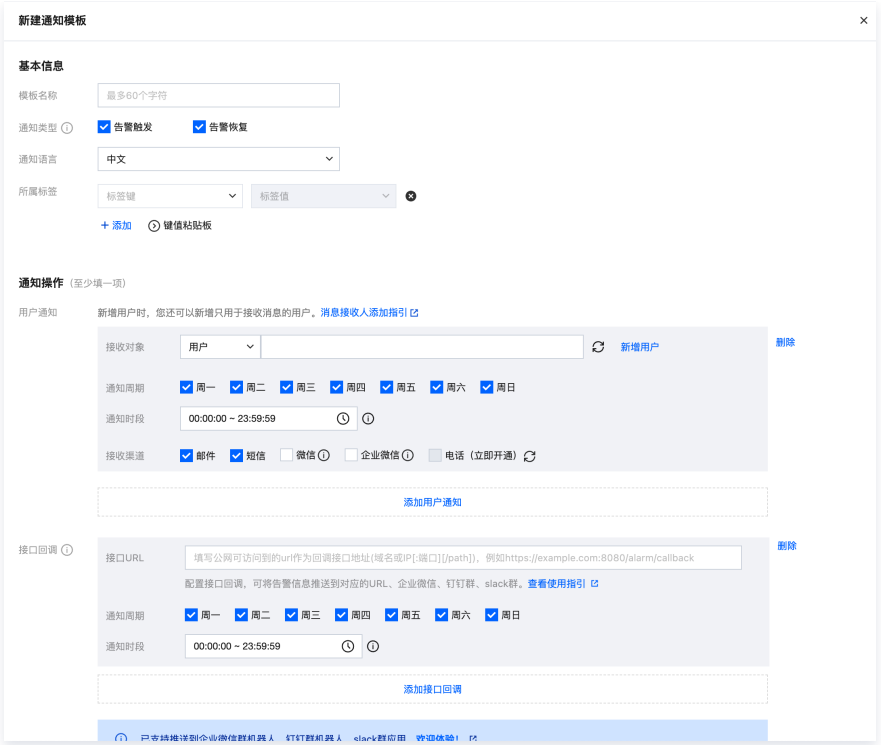
监听器监控

1. 在左侧目录中，选择**告警管理 > 告警配置 > 告警策略**，单击**新建策略**，进入新建策略页面。
2. 在“策略类型”中，选择**全球应用加速 > 4层监听器监控/7层监听器监控**。

3. 在**配置告警规则**模块中，选择**告警对象**，您可根据需要添加对象进行监控。
   在**触发条件**中，您可选择**使用模板**或进行**手动配置**。
   如选择**使用模板**，您可使用之前已经配置的告警策略。如无合适模板，您可通过新建模板的方式进行配置。该模板会存储在控制台中，方便您的后续使用。

4. 在下一步**配置告警通知**中，单击**新建模板**，添加通知模板名称并选择接收对象及渠道。

> **注意：**
> 接收对象需绑定对应渠道，否则将无法收到告警通知。

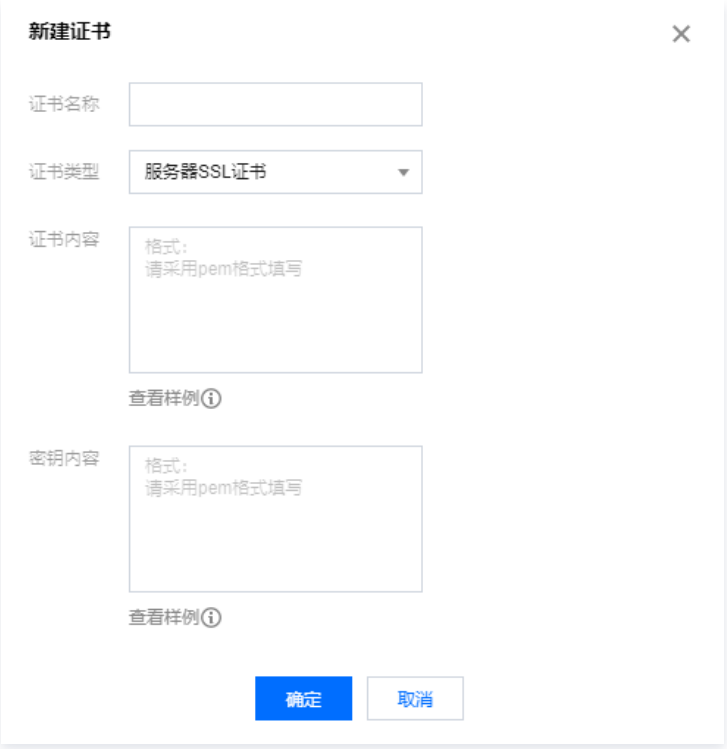

新建通知模板
基本信息
模板名称
最多60个字符
通知类型
告警触发
告警恢复
通知语言
中文
所属标签
标签键
标签值
添加
键值粘贴板
通知操作（至少填一项）
用户通知
新增用户时，您还可以新增只用于接收消息的用户。消息接收人添加指引
接收对象
用户
新增用户
删除
通知周期
周一
周二
周三
周四
周五
周六
周日
通知时段
00:00:00 ~ 23:59:59
接收渠道
邮件
短信
微信
企业微信
电话（立即开通）
添加用户通知
接口回调
接口URL
填写公网可访问到的url作为回调接口地址(域名或IP[:端口][/path])，例如https://example.com:8080/alarm/callback
配置接口回调，可将告警信息推送到对应的URL、企业微信、钉钉群、slack群。查看使用指引
删除
通知周期
周一
周二
周三
周四
周五
周六
周日
通知时段
00:00:00 ~ 23:59:59
添加接口回调

证书管理

最近更新时间：2025-05-16 17:48:12

证书新增

1. 登录 全球应用加速控制台，进入**证书管理**页面，单击**新增**。
2. 填写证书信息。



- 证书名称：用户自定义配置该证书的别名。
- 证书类型：分为[基础认证配置]、[客户端 CA 证书]、[源站 CA 证书]、[服务器 SSL 证书]、[通道 SSL 证书]；证书介绍、配置步骤见下表：
  其中[服务器 SSL 证书]、[通道 SSL 证书]需要配置密钥,密钥需要在腾讯云的 **SSL 证书管理**产品中进行购买。
- 证书内容：采用 pem 格式填写证书内容。
- 密钥内容：采用 pem 格式填写密钥内容。

证书类型	证书介绍	配置步骤
基础认证配置	用于客户端访问时进行账号密码验证	进入 HTTPS 监听器，点击**高级认证配置**，开启“基础认证”
客户端CA证书	用于客户端也提供证书时，进行双向认证	新建 HTTPS 监听器时，“SSL 解析方式”为双向认证，配置在“客户端证书”
服务器SSL证书	用于客户端到VIP建联时的安全加密	新建 HTTPS 监听器时，“协议”选择**监听器与源站之间使用 HTTP 协议**；配置在“服务器证书”； 也可以在“修改配置”、“修改证书”处修改证书

证书详情

进入**证书管理**页面，单击证书 ID/名称或单击该证书**详情**按钮即可查看该证书的详细信息。

证书删除

进入**证书管理**页面，单击证书**删除**按钮，然后单击弹窗**确定**按钮即可删除该证书。

获取访问用户真实 IP
通过 TOA 获取客户端真实 IP（仅针对 TCP 协议）
基本原理

最近更新时间：2024-09-13 17:24:02

加速通道转发数据包时，数据包同时会做 SNAT 和 DNAT，即数据包的源地址和目标地址均修改。源站看到的数据包的源地址是加速通道转发 IP 地址，而并非是客户端的真实 IP。为了将客户端 IP 传给服务器，加速通道将客户端的 IP 和 Port 在转发时放入了自定义的 tcp option 字段中。如下：

```
#define TCPOPT_ADDR 200
#define TCPOLEN_ADDR 8   /* |opcode|size|ip+port| = 1 + 1 + 6 */

/*
    * insert client ip in tcp option.
    * must be 4 bytes alignment.
    */

    struct ip_vs_tcpo_addr{
        __u8 opcode;
            __u8 opsize;
            __u16 port;
            __u32 addr;
    };
```

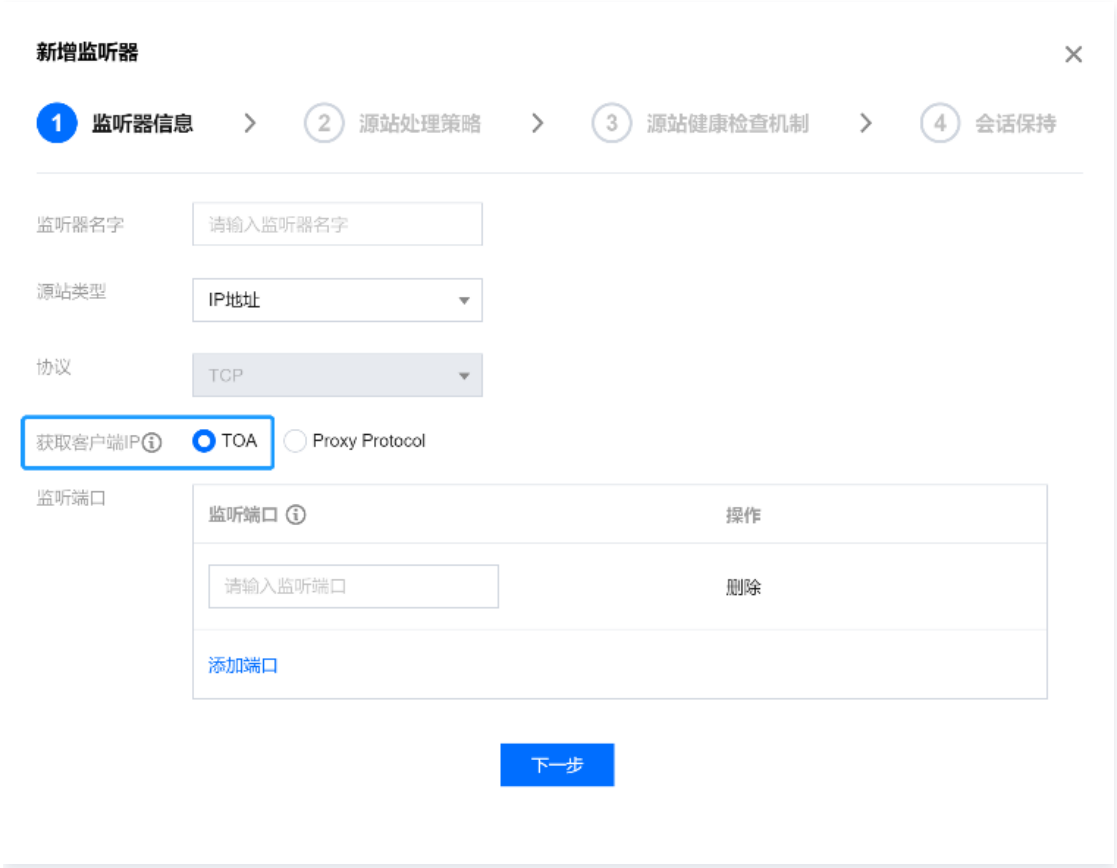
Linux 内核在监听套接字收到三次握手的 ACK 包之后，会从 SYN_REVC 状态进入到 TCP_ESTABLISHED 状态。这时内核会调用 tcp_v4_syn_recv_sock 函数。Hook 函数 tcp_v4_syn_recv_sock_toa首先调用原有的tcp_v4_syn_recv_sock函数，然后调用 get_toa_data 函数从 TCP OPTION 中提取出 TOA OPTION，并存储在 sk_user_data 字段中。再用 inet_getname_toa hook inet_getname，在获取源 IP 地址和端口时，首先调用原来的inet_getname，然后判断 sk_user_data 是否为空，如果有数据从其中提取真实的 IP 和 port，替换 inet_getname 的返回。
服务端程序在用户态调用 getpeername，返回的 IP 和 port 即为客户端的原始 IP。

Linux 后端版本调用
步骤一：创建 TCP 监听器并开启 TOA

最近更新时间：2024-09-13 14:55:51

> **注意：**
> 若您在后端适配过程中遇到无法解决的问题，可通过 工单联系 我们。

仅四层 TCP 支持 TOA 获取客户端真实 IP，请根据以下指引，在加速通道中选择开启 TOA。
控制台步骤：登录腾讯云 GAAP 控制台 > 进入加速通道（监听器配置）> 新增 TCP 监听器管理 > 勾选 TOA > 按照指引完成监听器、通道创建。



步骤二：后端服务加载 TOA 模块

最近更新时间：2024-09-13 17:24:02

您可以通过以下两种方式加载 TOA 模块：

- 方法一（推荐）：根据源站 Linux 版本，下载对应版本已编译好的 toa.ko 文件直接进行加载。
- 方法二：如果方法一中没有找到您当前的源站 Linux 版本，您可以通过下载 TOA 源码文件自行编译并加载。

> ⚠ 注意：
> 因不同安装环境的差异，如果您使用方法一加载过程中遇到问题，请尝试使用方法二，自行安装编译环境后加载。

方法一：直接下载源码并加载模块

1. 根据腾讯云上 Linux 的版本，下载对应的 TOA 包并解压。

   arm64
   - kernel-4.18.0.rar

   centos
   - CentOS-7.2-x86_64.tar.gz
   - CentOS-7.3-x86_64.tar.gz
   - CentOS-7.4-x86_64.tar.gz
   - CentOS-7.5-x86_64.tar.gz
   - CentOS-7.6-x86_64.tar.gz
   - CentOS-7.7-x86_64.tar.gz
   - CentOS-7.8-x86_64.tar.gz
   - CentOS-7.9-x86_64.tar.gz
   - CentOS-8.0-x86_64.tar.gz
   - CentOS-8.2-x86_64.tar.gz

   debian
   - Debian-11.1-x86_64.tar.gz
   - Debian-10.2-x86_64.tar.gz
   - Debian-9.0-x86_64.tar.gz

   suse linux
   - openSUSE-Leap-15.3-x86_64.tar.gz

   ubuntu
   - Ubuntu-14.04.1-LTS-x86_64.tar.gz
   - Ubuntu-16.04.1-LTS-x86_64.tar.gz
   - Ubuntu-18.04.1-LTS-x86_64.tar.gz
   - Ubuntu-20.04.1-LTS-x86_64.tar.gz

2. 解压完成后，执行 cd 命令进入刚解压的文件夹里，执行加载模块的指令：

```
insmod toa.ko
```

3. 执行下面指令确认是否已加载成功：

```
lsmod | grep toa
```

```
[root@VM-16-42-centos ~]# lsmod | grep toa
toa                    278528  0
```

4. 加载成功，在启动脚本里面加载 toa.ko 文件（重启机器 ko 文件需要重新加载）。

```
echo "insmod   xxxxx /toa.ko" >> /etc/rc.local
```

5. （可选）临时关闭 TOA：rmmod 路径/模块名。

```
rmmod toa.ko
```

6. （可选）若不再需要使用 TOA 模块，执行以下命令进行卸载。

```
rmmod toa
```

7. （可选）执行以下命令确认 TOA 模块是否卸载成功。若提示“TOA unloaded”，则说明卸载成功。

```
dmesg -T
```

方法二：自行编码并加载模块

1. 安装编译环境。

1.1 查看当前内核版本号，确认 kernel-devel，kernel-headers 已安装，并保证版本号与内核版本保持一致。

1.2 确认已安装 gcc 和 make。

1.3 如果以上环境依赖没有安装，可参考如下命令进行安装：

Centos

```
yum install -y gcc
yum install -y make
yum install -y kernel-headers kernel-devel
```

Ubuntu/Debian

```
apt-get install -y gcc
apt-get install -y make
apt-get install -y linux-headers-$(uname -r)
```

2. 安装完编译环境后，执行以下命令完成源码下载，编译和加载。

脚本一键编译并加载

```
/bin/bash -c "$(curl -fsSL https://thunder-pro-mainland-1258348367.cos.ap-
guangzhou.myqcloud.com/TOA/compile_install_toa.sh)"
```

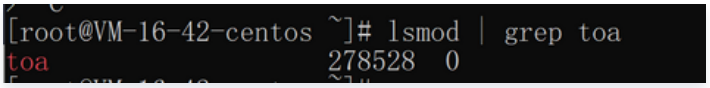
手工编译并加载

```
# 创建并进入编译目录
mkdir toa_compile && cd toa_compile
# 下载源代码tar包
curl -o toa.tar.gz https://thunder-pro-mainland-1258348367.cos.ap-
guangzhou.myqcloud.com/TOA/toa.tar.gz
# 解压tar包
tar -zxvf toa.tar.gz
# 编译toa.ko文件，编译成功后会在当前目录下生成toa.ko文件
make
# 加载toa模块
insmod toa.ko
# 拷贝到内核模块目录下
cp toa.ko /lib/modules/`uname -r`/kernel/net/netfilter/ipvs/toa.ko
# 设置系统启动时自动加载toa模块
echo "insmod /lib/modules/`uname -r`/kernel/net/netfilter/ipvs/toa.ko" >> /etc/rc.local
```

3. 执行下面指令确认是否已加载成功：

```
lsmod | grep toa
```

出现 toa 则表示已加载成功，如下图所示：



步骤三：验证获取客户端 IP 信息

最近更新时间：2024-09-13 14:55:51

您可以通过搭建 TCP 服务，并通过另外一台服务器模拟客户端请求进行验证，示例如下：

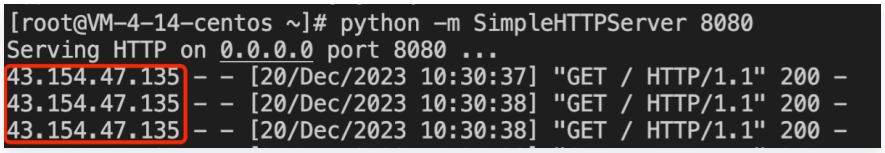
1. 在当前服务器上，可以通过 Python 创建一个 HTTP 服务来模拟 TCP 服务，如下所示：

```
# 基于python2
python2 -m SimpleHTTPServer 8080
# 基于python3
python3 -m http.server 8080
```

2. 用另一台服务器充当客户端，构造客户端请求，以 Curl 请求来模拟 TCP 请求：

```
# 利用curl发起http请求，其中域名为GAAP通道的域名，8080为GAAP源站端口
curl -i "http://link-xxxxxx.gaapqcloud.com.cn:8080/"
```

3. 如果TOA正常加载，在当前服务器上会看到客户端的真实地址信息，如下图红框所示：



4. 如果 TOA 未加载或 TOA 无法正常工作，在当前服务器上会看到客户端地址为 GAAP 回源的公网地址，建议排查是否安装正确版本的 TOA 包，如果没有对应版本 TOA 包建议用 步骤二中方法二：自行编码并加载模块 进行加载。

步骤四：（可选）修改源站业务代码，同时获取 IPv4/IPv6 客户端真实 IP

最近更新时间：2024-09-13 14:55:51

说明：
本章节操作仅在源站需同时获取 IPv4 和 IPv6 客户端地址信息时参考，该操作将指引您如何修改源站业务代码。

源站在建立服务监听时，可参考采用如下两种方式：

1. 采用 IPv4 的地址结构（`struct sockaddr_in`）搭建服务，其监听的是 IPv4 格式的地址。
2. 采用 IPv6 的地址结构（`struct sockaddr_in6`）搭建服务，其监听的是 IPv6 格式的地址。

示例代码

监听 IPv4 地址

C

```c
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netinet/in.h>
#include <memory.h>
#include <arpa/inet.h>

int main(int argc,  char **argv){
    int l_sockfd;
    // 服务器地址采用v4结构
    struct sockaddr_in serveraddr;
    // 业务修改点：客户端地址必须采用v6结构
    struct sockaddr_in6 clientAddr;
    int server_port = 10000;

    memset(&serveraddr, 0, sizeof(serveraddr));
    // 创建socket
    l_sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (l_sockfd == -1){
        printf("Failed to create socket.\n");
        return -1;
    }

    // 初始化服务器地址信息
    memset(&serveraddr, 0, sizeof(struct sockaddr_in));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(server_port);
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);

    int isReuse = 1;
    setsockopt(l_sockfd, SOL_SOCKET,SO_REUSEADDR,(const char*)&isReuse,sizeof(isReuse));

    // 关联socket和服务器地址信息
    int nRet = bind(l_sockfd,(struct sockaddr*)&serveraddr, sizeof(serveraddr));
    if(-1 == nRet)
    {
        printf("bind error\n");
```

```
        return -1;
    }
    // 监听socket
    listen(l_sockfd, 5);

    int clientAddrLen = sizeof(clientAddr);
    memset(&clientAddr, 0, sizeof(clientAddr));
    // 接受来自客户端的连接
    int linkFd = accept(l_sockfd, (struct sockaddr*)&clientAddr, &clientAddrLen);
    if(-1 == linkFd)
    {
        printf("accept error\n");
        return -1;
    }
    // 业务修改点：根据客户端sin6_family的类型，判断客户端是v4地址还是v6地址
    //    当为AF_INET时，表示客户端是IPv4，将客户端地址指针转换为struct sockaddr_in*进行获取
    //    当为AF_INET6时，表示客户端是IPv6，使用struct sockaddr_in6*进行获取
    if (clientAddr.sin6_family == AF_INET) {
        printf("AF_INET accept getpeername %s : %d successful\n",
                inet_ntoa(((struct sockaddr_in*)&clientAddr)->sin_addr),
                ntohs(((struct sockaddr_in*)&clientAddr)->sin_port));
    }else if (clientAddr.sin6_family == AF_INET6){
        char addr_p[128] = {0};
        inet_ntop(AF_INET6, (void *)&((struct sockaddr_in6*)&clientAddr)->sin6_addr, addr_p,
(socklen_t )sizeof(addr_p));
        printf("AF_INET6 accept getpeername %s : %d successful\n",
                addr_p,
                ntohs(((struct sockaddr_in6*)&clientAddr)->sin6_port));
    }else{
        printf("unknow sin_family:%d \n", clientAddr.sin6_family);
    }
    close(l_sockfd);
    return 0;
}
```

Java

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketAddress;


public class ServerDemo {


    /** 若采用 IPv4 的地址结构搭建服务，使用 IPV4_HOST */
    public static final String IPV4_HOST = "0.0.0.0";


    /** 若采用 IPv6 的地址结构搭建服务，使用 IPV6_HOST */
    public static final String IPV6_HOST = "::";
```

```
    public static void main(String[] args) {
        int serverPort = 10000;
        try (ServerSocket serverSocket = new ServerSocket()) {
            // 设置地址复用
            serverSocket.setReuseAddress(true);
            // 绑定服务器地址和端口，这里使用 IPv4
            serverSocket.bind(new InetSocketAddress(InetAddress.getByName(IPV4_HOST), serverPort));
            System.out.println("Server is listening on port " + serverPort);


            while (true) {
                // 接受客户端连接
                Socket clientSocket = serverSocket.accept();
                System.out.println("New client connected: " +
clientSocket.getRemoteSocketAddress());


                // 处理客户端请求
                handleClientRequest(clientSocket);
            }
        } catch (IOException e) {
            System.err.println("Failed to create server socket: " + e.getMessage());
        }
    }


    /**
     * 处理函数，具体业务具体实现，这里只做为示例
     * 此函数的作用是将 client 的输入原封不动的返回给 client
     */
    private static void handleClientRequest(Socket clientSocket) {
        try (InputStream inputStream = clientSocket.getInputStream();
             OutputStream outputStream = clientSocket.getOutputStream()) {


            // 读取客户端发来的数据
            byte[] buffer = new byte[1024];
            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                // 将接收到的数据原样回复给客户端
                outputStream.write(buffer, 0, bytesRead);
            }


        } catch (IOException e) {
            // 当客户端断开连接后
            System.err.println("Failed to handle client request: " + e.getMessage());
        } finally {
            try {
                clientSocket.close();
            } catch (IOException e) {
                System.err.println("Failed to close client socket: " + e.getMessage());
            }
        }
    }
}
```

监听 IPv6 地址

C

```c
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netinet/in.h>
#include <memory.h>
#include <arpa/inet.h>

int main(int argc,  char **argv){
    int l_sockfd;
    // 服务器地址采用v6结构
    struct sockaddr_in6 serveraddr;
    // 客户端地址采用v6结构
    struct sockaddr_in6 clientAddr;
    int server_port = 10000;

    memset(&serveraddr, 0, sizeof(serveraddr));

    // 创建socket
    l_sockfd = socket(AF_INET6, SOCK_STREAM, 0);
    if (l_sockfd == -1){
        printf("Failed to create socket.\n");
        return -1;
    }
    // 设置服务器地址信息
    memset(&serveraddr, 0, sizeof(struct sockaddr_in6));
    serveraddr.sin6_family = AF_INET6;
    serveraddr.sin6_port = htons(server_port);
    serveraddr.sin6_addr = in6addr_any;

    int isReuse = 1;
    setsockopt(l_sockfd, SOL_SOCKET,SO_REUSEADDR,(const char*)&isReuse,sizeof(isReuse));
    // 关联socket和服务器地址信息
    int nRet = bind(l_sockfd,(struct sockaddr*)&serveraddr, sizeof(serveraddr));
    if(-1 == nRet)
    {
        printf("bind error\n");
        return -1;
    }
    // 监听socket
    listen(l_sockfd, 5);

    int clientAddrLen = sizeof(clientAddr);
    memset(&clientAddr, 0, sizeof(clientAddr));

    // 接受来自客户端的连接请求
    int linkFd = accept(l_sockfd, (struct sockaddr*)&clientAddr, &clientAddrLen);
    if(-1 == linkFd)
    {
        printf("accept error\n");
        return -1;
    }

    // 这里收到的客户端地址信息全部都采用v6的结构进行存储
```

```
    // 其中，客户端的IPv4地址也被映射成了一个IPv6的地址，例如：::ffff:119.29.1.1
    char addr_p[128] = {0};
    inet_ntop(AF_INET6, (void *)&clientAddr.sin6_addr, addr_p, (socklen_t )sizeof(addr_p));
    printf("accept %s : %d successful\n", addr_p, ntohs(clientAddr.sin6_port));

    // 业务修改点：通过系统宏定义IN6_IS_ADDR_V4MAPPED来判断一个IPv6地址是否是IPv4的映射地址（代表客户端是IPv4）
    if(IN6_IS_ADDR_V4MAPPED(&clientAddr.sin6_addr)) {
        struct sockaddr_in real_v4_sin;
        memset (&real_v4_sin, 0, sizeof (struct sockaddr_in));
        real_v4_sin.sin_family = AF_INET;
        real_v4_sin.sin_port = clientAddr.sin6_port;
        // 读取最后四个字节即为客户端真实IPv4地址
        memcpy (&real_v4_sin.sin_addr, ((char *)&clientAddr.sin6_addr) + 12, 4);
        printf("connect %s successful\n", inet_ntoa(real_v4_sin.sin_addr));
    }

    close(l_sockfd);
    return 0;
}
```

Java

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketAddress;

public class ServerDemo {

    /** 若采用 IPv4 的地址结构搭建服务，使用 IPV4_HOST */
    public static final String IPV4_HOST = "0.0.0.0";

    /** 若采用 IPv6 的地址结构搭建服务，使用 IPV6_HOST */
    public static final String IPV6_HOST = "::";

    public static void main(String[] args) {
        int serverPort = 10000;
        try (ServerSocket serverSocket = new ServerSocket()) {
            // 设置地址复用
            serverSocket.setReuseAddress(true);
            // 绑定服务器地址和端口，这里使用 IPv4
            serverSocket.bind(new InetSocketAddress(InetAddress.getByName(IPV6_HOST), serverPort));
            System.out.println("Server is listening on port " + serverPort);

            while (true) {
                // 接受客户端连接
                Socket clientSocket = serverSocket.accept();
                System.out.println("New client connected: " +
clientSocket.getRemoteSocketAddress());

                // 处理客户端请求
                handleClientRequest(clientSocket);
            }
        } catch (IOException e) {
```

```
            System.err.println("Failed to create server socket: " + e.getMessage());
        }
    }

    /**
     * 处理函数，具体业务具体实现，这里只做为示例
     * 此函数的作用是将 client 的输入原封不动的返回给 client
     */
    private static void handleClientRequest(Socket clientSocket) {
        try (InputStream inputStream = clientSocket.getInputStream();
             OutputStream outputStream = clientSocket.getOutputStream()) {

            // 读取客户端发来的数据
            byte[] buffer = new byte[1024];
            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                // 将接收到的数据原样回复给客户端
                outputStream.write(buffer, 0, bytesRead);
            }

        } catch (IOException e) {
            // 当客户端断开连接后
            System.err.println("Failed to handle client request: " + e.getMessage());
        } finally {
            try {
                clientSocket.close();
            } catch (IOException e) {
                System.err.println("Failed to close client socket: " + e.getMessage());
            }
        }
    }
}
```

控制台输出结果

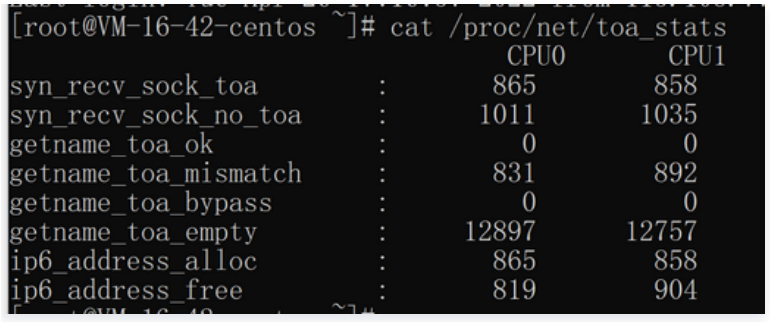
```
Server is listening on port 10000
New client connected: /127.0.0.1:50680
New client connected: /0:0:0:0:0:0:0:1:51124
New client connected: /127.0.0.1:51136
```

步骤五：（可选）查看 TOA 相关的计数状态

最近更新时间：2024-09-13 17:24:02

为保障 TOA 内核模块运行的稳定性，TOA 内核模块还提供了监控功能。在插入 toa.ko 内核模块后，可以通过以下两种方式监控 TOA 模块的工作状态。执行以下命令查看 TOA 相关的计数状态。

```
cat /proc/net/toa_stats
```



其中主要的监控指标对应的含义如下所示：

指标名称	说明
syn_recv_sock_toa	接收带有 TOA 信息的连接个数。
syn_recv_sock_no_toa	接收并不带有 TOA 信息的连接个数。
getname_toa_ok	调用 getsockopt 获取源 IP 成功即会增加此计数，另外调用 accept 函数接收客户端请求时也会增加此计数。
getname_toa_mismatch	调用 getsockopt 获取源 IP 时，当类型不匹配时，此计数增加。例如某条客户端连接内存放的是 IPv4 源 IP，并非为 IPv6 地址时，此计数便会增加。
getname_toa_empty	对某一个不含有 TOA 的客户端文件描述符调用 getsockopt 函数时，此计数便会增加。
ip6_address_alloc	当 TOA 内核模块获取 TCP 数据包中保存的源 IP、源 Port 时，会申请空间保存信息。
ip6_address_free	当连接释放时，toa 内核模块会释放先前用于保存源 IP、源 port 的内存，在所有连接都关闭的情况下，所有 CPU 的此计数相加应等于 ip6_address_alloc 的计数。

查看 Client IP

最近更新时间：2024-09-13 14:55:51

参考方法一：直接在 nginx 日志中查看（日志路径：/var/log/nginx/access.log）
参考方法二：使用 wireshark 查看 tcpdump 抓包获取为。

1. 在后端服务器执行以下命令进行抓包：

```
sudo tcpdump -i eth0 -w dump.pcap
```

-i 指定要抓取的网卡
-w 指定结果保存位置

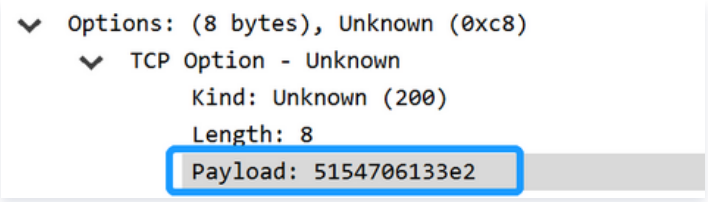
2. 客户端访问测试地址后，按下 ctrl + c 停止抓包：

```
[root@VM-16-42-centos ~]# sudo tcpdump -i eth0 -w dump.pcap
dropped privs to tcpdump
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C361 packets captured
362 packets received by filter
0 packets dropped by kernel
```

3. 用 sz 命令或其他方式把 dump.pcap 文件下载到本地：

```
sz dump.pcap
```

4. wireshark 打开下载的 dump.pcap 文件，从 TCP Option 中查看客户端真实 IP。
此字段后4个字节（十六进制）即为客户端真实 IP



常见问题

最近更新时间：2024-09-13 14:55:51

签名报错，例如 module verification failed: signature and/or required key missing - tainting kernel

- 解决：linux 内核会有对模块有签名校验，取决于编译内核时候是否开启此特性
- 解决方法一：编译内核时，去掉签名支持 CONFIG_MODULE_SIG=n
- 解决方法二：有证书的情况下，可进行签名，举个例子：

```
/usr/src/linux-4.9.61/scripts/sign-file sha512/usr/src/linux-4.9.61/certs/signing_key.pem /usr/src/linux-4.9.61/certs/signing_key.x509 toa.ko
```

编译时报错没有 /lib/modules 目录

- 解决：常见以下三种情况
- 没安装有内核包
- 路径被修改过，需要自行纠正下
- 安装的内核没有 build 目录，需手动软连接到对应版本内核的 header，例如

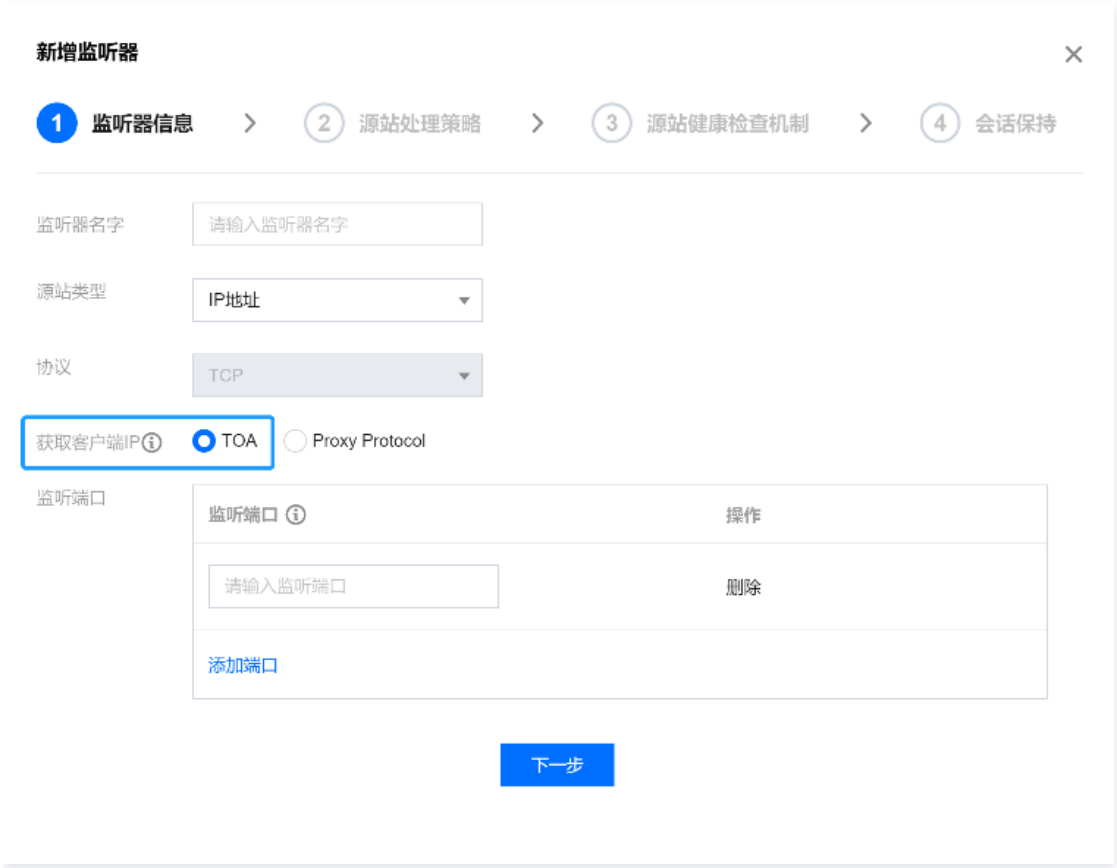
```
cd /lib/modules/4.9.0-13-amd64 && ln -s /usr/src/linux-headers-4.9.0-13-amd64 build
```

Windows 后端版本调用
步骤一：创建 TCP 监听器并开启 TOA

最近更新时间：2024-09-13 14:55:51

> **注意：**
> 若您在后端适配过程中遇到无法解决的问题，可通过 工单联系 我们。

仅四层 TCP 支持 TOA 获取客户端真实 IP，请根据以下指引，在加速通道中选择开启 TOA。
控制台操作步骤：登录 腾讯云 GAAP 控制台 > 加速通道（监听器配置） > 新增 TCP 监听器管理 > 勾选 TOA > 按照指引完成监听器、通道创建。



步骤二：后端服务加载 TOA 模块

最近更新时间：2024-09-13 17:24:02

下载文件

获取文件。

通用版本

文件说明

文件	说明
WinPcap_4_1_3.exe	winpcap 驱动，详情见 WinPcap 文档。
lib_toa.lib	TOA 静态库。
toa_fetcher.h	静态库依赖的头文件。
pcap.h	静态库依赖的头文件。

环境准备

1. 安装 winpcap 驱动：双击 WinPcap_4_1_3.exe（不需重启）。
2. 添加 lib_toa.lib 到工程的 lib 库路径下。
3. 添加 toa_fetcher.h，pcap.h 到工程的头文件中。

Go 版本

文件说明

文件	说明
WinPcap_4_1_3.exe	winpcap 驱动，详情请参见 WinPcap 官网。
toa_win.exe	Windows 服务器端 TOA 服务程序。
toa_win.conf	Windows 服务器端 TOA 服务程序配置文件。
program_auto_up.bat	Windows 服务器端服务监控 bat 脚本。
demo.go	Go 语言编写的示例程序，用于访问 TOA 服务程序。

部署步骤

1. 修改配置文件 toa_win.conf，参数说明如下：

参数	是否必选	说明
ToaWinPort	是	toa_win.exe 的服务端口，用于与 TOA 获取客户端通信，默认为9999。
NetworkCardIP	是	用于识别网络接口的 IP 地址字符串，例如：10.75.132.39，该网卡为与客户端通信的网卡。
ServerListenIP	是	服务器的 IP 地址字符串，例如：10.75.132.39，用于过滤 TCP 流。
ServerListenPortList	否	服务器的端口列表，用于过滤 TCP 流，最多可以填三个端口。ServerListenPortList 和 PortRange 必须至少设置一个。
PortRange	否	服务器端口范围列表，用于过滤 TCP 流，最多可以填三个端口。ServerListenPortList 和 PortRange 必须至少设置一个。
CacheSeconds	否	缓存的时长，单位：秒，默认为15秒。

> ⚠ **注意：**
> 配置文件必须和 toa_win.exe 放在同一个目录下。

```
#ToaWinPort
9999
#NetworkCardIP
172.19.0.9
#ServerListenIP
172.19.0.9
#ServerListenPortList
9102;5555;6666
#PortRange
6666-7777;7777-8888
#CacheSeconds
15
```

2. 修改 program_auto_up.bat。
   修改路径为程序所在的目录，将脚本添加到定时任务中，周期性执行该脚本用于监控 toa_win.exe 程序，当程序退出时，自动拉起。

```
@echo off
set Program="toa_win.exe"
tasklist -v | findstr %Program% > NUL
if ErrorLevel 1 (
    echo "process not exists" >> auto_up_log.txt
    echo %date%+ %time% >> auto_up_log.txt
    C:
    cd C:\xxxx\
    toa_win.exe
)else (
    echo "process exists"
)
```

3. 启动 toa_win.exe 程序，log 日志将存在同一目录下的 toa_win.log。此时，可以通过 udp 通信的方式向 TOA 服务获取真实的 IP 地址，详情请参见 使用方法 。

步骤三：使用方法

最近更新时间：2024-09-13 17:24:02

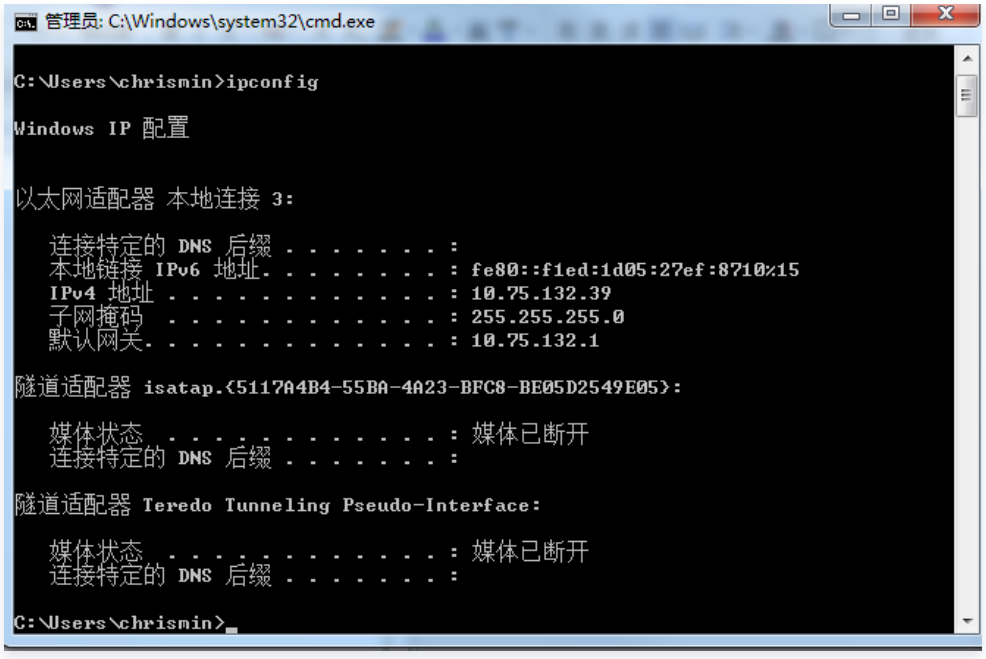
通用版本

数据结构和函数说明

- **class ToaFetcher**
  主体类，用于管理 TOA 的获取和释放。
- **InitUpToaFetcher**
  1.1 函数说明：
  该函数用于初始化 TOA fetcher。

```
bool InitUpToaFetcher(char *ncard_ip_str, char *svr_ip_str, u_short svr_port[], u_short
svr_port_num, u_short cache_secs=TIMER_CACHE_SECS)
```

  1.2 入参说明：
  - ncard_ip_str： 用于识别网络接口的 IP 地址字符串，例如：10.75.132.39，该网卡为与客户端通信的网卡，如下图所示：



  - svr_ip_str：服务器的 IP 地址字符串，例如：10.75.132.39，用于过滤 TCP 流。
  - svr_port：服务器的端口列表，用于过滤 TCP 流，最多可以填三个端口，svr_port 和 port_range_ptr 至少设置其中一个。
  - svr_port_num：服务器的端口个数。
  - port_range_ptr：服务器的端口范围数组指针，其中元素为指向一个字符串的指针，端口范围字符串格式：10001-10005，用于过滤 TCP 流，最多填三个范围，svr_port 和 port_range_ptr 至少设置其中一个。
  - port_range_num：服务器的端口范围个数。
  - cache_secs：缓存的时长，单位：秒，默认15秒，详见 toa_fetcher.h：TIMER_CACHE_SECS，缓存时间到期后，将不再保存该 TOA。

  1.3 返回值
  - TRUE：表示创建 TOA 获取旁路线程成功。
  - FALSE：表示创建 TOA 获取旁路线程失败。
- **FetchToaValue**
  1.1 函数说明
  该函数用于获取 TOA 值，tcp-syn 包交互后，最长需要等待 1ms 后可以获取到 TOA，正常情况下三次握手需要消耗1ms以上。

```
bool FetchToaValue(u_long fake_client_ip_addr, u_short fake_client_port, u_long
&real_client_ip_addr, u_short &real_client_port)
```

1.2 入参说明

- fake_client_ip_addr：客户端伪 IP 地址，采用网络序存储，从服务器 accept 函数返回的对端地址中获取。
- fake_client_port：客户端伪端口号，采用网络序存储，从服务器 accept 函数返回的对端地址中获取。
- real_client_ip_addr：客户端真实 IP 地址，采用网络序存储，从 TOA 中获取。
- real_client_port：客户端真实端口号，采用网络序存储，从 TOA 中获取。

1.3 返回值

- TRUE：获取 TOA 成功。
- FALSE：未获取到 TOA，一般是超过缓存时间导致 TOA 被清掉。

- **StopToaFetcher**

1.1 函数说明

该函数用于停止 TOA fetcher。

```
void StopToaFetcher()
```

1.2 入参说明

无。

1.3 返回值

无。

- **GetFetcherStatus**

1.1 函数说明

该函数用于获取 Fetcher 状态。

```
int GetFetcherStatus()
```

1.2 入参说明

无。

1.3 返回值

- 0：表示初始状态。创建实例后，初始状态处于该状态，Fetcher 初始化中，该状态保持不变，当中间出现错误时，返回 -1，当成功运行时，返回 1。
- -1：表示异常状态。
- 1：表示正常运行中。

- **FetchThreadHandler**

1.1 函数说明

该函数用于获取 TOA 旁路线程句柄。

```
HANDLE FetchThreadHandler()
```

1.2 入参说明

无。

1.3 返回值

TOA 旁路线程句柄，当 ToaFetcher 实例被销毁时，将主动关闭该句柄。

- **FetchErrorInfo**

1.1 函数说明

该函数用于获取错误码。

```
bool FetchErrorInfo(int* err_code_ptr, char* err_msg_ptr)
```

1.2 入参说明

- err_code_ptr：一个整型指针指向错误码，用于返回错误码。
- err_msg_ptr ：一个字符指针指向字符串缓冲区，至少50字节，用于返回错误信息。

1.3 返回值

- TRUE：获取正常。
- FALSE：获取异常。

错误码

错误码	错误信息	说明
0	Ok	正常
-1001	Exceed max server port number	超过最大的端口数，请检查 InitUpToaFetcher：svr_port_num。
-1002	Invalid IP address	非法的 IPv4 地址。
-1003	No suitable network interface	未找到合适的网络接口。
-1004	System Error: find dev error	系统错误：未找到 dev，请联系 lib 开发者。
-1005	System Error: start timer error	系统错误：定时器启动错误，请联系 lib 开发者。
-1006	System Error: compile filter error	系统错误：过滤规则编译错误，请联系lib开发者。
-1007	System Error: set filter error	系统错误：过滤规则设置错误，请联系 lib 开发者。
-1008	System Error: open pcap error	系统错误：打开 dev 错误，请联系 lib 开发者。
-1009	System Error: start pcap error	系统错误：启动监听错误，请联系 lib 开发者。
-1010	System Error: begin thread error	系统错误：启动线程错误，请联系 lib 开发者。
-1999	Unknown error	未知错误，请联系 lib 开发者。

示例

- **初始化 ToaFetcher：**

```
char ncard_ip_str[] = "1.1.1.1";
char svr_ip_str[] = "1.1.1.1";
char port_range[3][100] = {"10001-10005", "20001-20005", "30001-30005"};
char* port_range_ptr[3] = {port_range[0], port_range[1], port_range[2]};
      u_short svr_port_list[3] = {1111, 2222, 3333};
      ToaFetcher inst = ToaFetcher();
      inst.InitUpToaFetcher((char*)ncard_ip_str, (char*)svr_ip_str, svr_port_list, 3);
```

- **获取 TOA：**

```
void GetToa(SOCKADDR_IN client_addr, ToaFetcher * toa_fetcher_ptr)
{
  u_long fake_client_ip_addr = 0;
  u_short fake_client_port = 0;
  u_long real_client_ip_addr = 0;
  u_short real_client_port = 0;
  memcpy(&fake_client_ip_addr, &client_addr.sin_addr, 4);
  memcpy(&fake_client_port, &client_addr.sin_port, 2);
  bool ret = toa_fetcher_ptr->FetchToaValue(fake_client_ip_addr, fake_client_port,
real_client_ip_addr, real_client_port);
  if(ret == FALSE){
      printf("No toa found\n");
```

```
    }else{
        //fpp: 自定义的打印函数
        fpp("real_client_ip_addr", &real_client_ip_addr, 4);
     fpp("real_client_port", &real_client_port, 2);
    }
}
```

Go版本

TOA 获取端通过本机 UDP 通信的方式向 toa_win.exe 获取真实 IP 地址。

协议格式

- **请求：** `| ID (4Bytes) |  FakeIPAddress (4Bytes) | FakePort (2Bytes) |`

  字段说明如下：
  - ID：4字节，用于唯一标识一个请求，响应中将原始返回。
  - FakeIPAddrss：4字节，客户端伪 IP 地址，采用网络序存储，从服务器 accept 函数返回的对端地址中获取。
  - FakePort：2字节，客户端伪端口号，采用网络序存储，从服务器 accept 函数返回的对端地址中获取。
- **响应：** `| ID (4Bytes) | Code (1Byte) | RealIPAddress (4Bytes) | RealPort (2Bytes) |`

字段说明如下：
  - ID：4字节，用于唯一标识一个请求，和请求携带上来的一致。
  - Code：1字节，0：成功获取到真实 IP 和 Port，1：获取失败。
  - RealIPAddress：4字节，网络序，当 Code=0 时存在，表示真实的客户端 IP 地址。
  - RealPort：2字节，网络序，当 Code=0 时存在，表示真实的客户端 Port。

示例

详情请参见 demo.go，可以自行开发 TOA 获取客户端程序，也可以使用 demo.go 中的 queryToa 函数进行获取。

1. **函数说明**

```
func queryToa(serverAddr string, fakeIp string, fakePort uint16)(int32, string, uint16)
```

2. **入参说明**
   - serverAddr：字符串类型，toa_win.exe 的服务通信地址，格式：127.0.0.1:9999。
   - fakeIp：字符串类型，伪 IP 地址，格式：1.2.3.4。
   - fakePort：uint16类型，伪 Port，格式：8899。
3. **返回值**
   - 第一个返回值：int32类型，用于表示 error code。
     - 0：成功获取。
     - -1：toa 获取失败，可能因为 fakeIP 和 fakePort 不对或者 cache 到期。
     - -2：网络通信导致的失败。
   - 第二个返回值：字符串类型，当 toa 获取成功时，返回真实的 IP，否则为空字符串。
   - 第三个返回值：uint16 类型，当 toa 获取成功时，返回真实的 Port，否则为0。

通过 Proxy Protocol 获取客户端真实 IP（仅针对 TCP 协议）

基本原理

最近更新时间：2024-12-02 14:54:03

Proxy Protocol 是通过为 TCP 添加一个头部信息，来方便的传递客户端信息（协议栈、源 IP、目的 IP、源端口、目的端口等），在网络情况复杂又需要获取用户真实 IP 时非常有用。其本质是在三次握手结束后由代理在连接中插入一个携带了原始连接四元组信息的数据包。
Proxy Protocol 方式获取客户端 IP 需要先在控制台配置开启使用（当前仅支持协议为 TCP 的监听器使用），加速服务在和源站建立连接后，会在传输的第一个 payload 的报文前插入Proxy Protocol 协议文本。

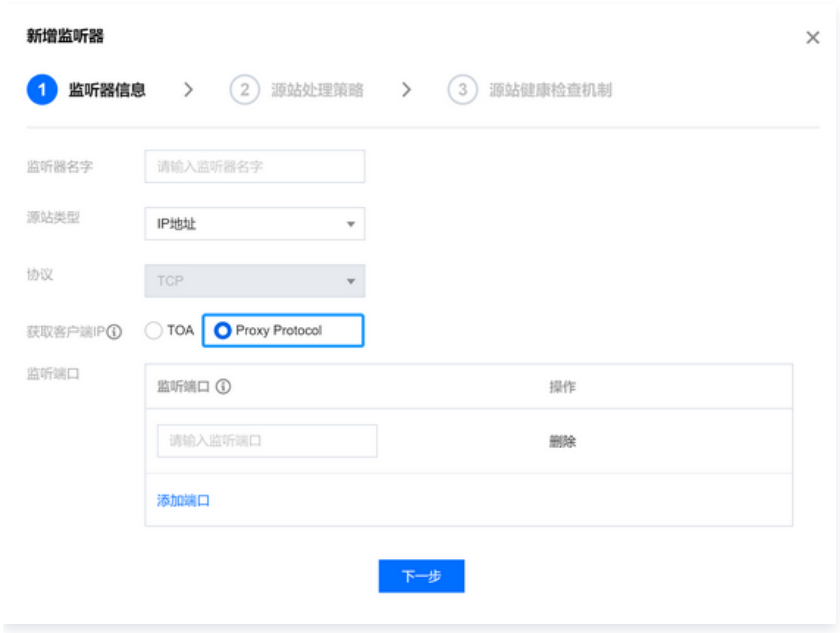
操作步骤

最近更新时间：2024-09-13 14:55:51

> **注意：**
> 若您在后端适配过程中遇到无法解决的问题，可通过 工单联系 我们。

步骤一：创建TCP监听器并开启 Proxy Protocol

仅四层 TCP 支持 Proxy Protocol 获取客户端真实 IP，请根据以下指引，在加速通道中选择开启 Proxy Protocol。
控制台操作步骤：登录 腾讯云 GAAP 控制台 > 加速通道（监听器配置）> 新增 TCP 监听器管理 > 勾选 Proxy Protocol > 按照指引完成监听器、通道创建。



步骤二：后端服务适配 Proxy Protocol 协议

当前 Nginx 和 HaProxy 都已经支持 Proxy Protocol 协议。
以 Nginx 为例，配置支持 Proxy Protocol 协议只需要将参数 proxy_protocol 添加在 server 块中的 listen 指令后：

```
http {
    #...
    server {
        listen 80   proxy_protocol;
        listen 443  ssl proxy_protocol;
        #...
    }
}
```

不支持 Proxy Protocol 的应用程序，需要在 TCP 连接建立后，读取 Proxy Protocol 的文本行并进行字符串解析来获取客户端 IP，字符示例如下所示：

```
PROXY TCP4 1.1.1.2 2.2.2.2 12345 80\r\n
```

步骤三：查看 Client IP

- 参考方法一：直接在 nginx 日志中查看（日志路径：/var/log/nginx/access.log）。
- 参考办法二： 执行命令 nc -l port 查看。

```
[root@VM-16-42-centos ~]# nc -l 80
PROXY TCP4 112.97.[redacted].1[redacted] 172.16.9.142 41131 80
GET / HTTP/1.1
Host: link-cfs4lo35.gaapqcloud.com.cn
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 14_8 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.2 Mobile/15E148 Safari/604.1
Accept-Language: zh-tw
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

通过 HTTP 请求头获取客户端真实 IP（支持 HTTP/HTTPS 协议）
基本原理

最近更新时间：2024-09-13 14:55:51

使用 HTTP/HTTPS 监听器后，源站可直接从 HTTP 请求头中 X-Real-IP 或 X-Forwarded-For 字段中获取客户端真实 IP，此为默认生效功能。同时支持从 “回源 HTTP 请求头配置” 自定义，若从源站到程序还有中间链路 (如 CLB，自建 nginx)，则需要自行配置，以防字段被中间链路覆盖。

操作步骤

最近更新时间：2024-09-13 14:55:51

> **注意：**
> 若您在后端适配过程中遇到无法解决的问题，可通过 工单联系 我们。

步骤一：创建 HTTP/HTTPS 监听器

控制台操作步骤：登录 腾讯云 GAAP 控制台 > 加速通道（监听器配置） > 新增 HTTP/HTTPS 监听器管理 > 按照指引完成监听器、通道创建。

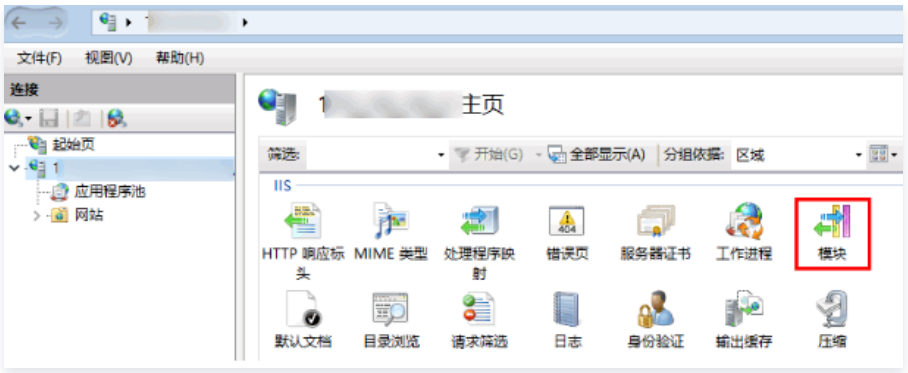
步骤二：后端服务适配

以下对常见的应用服务器 X-Forwarded-For 配置方案进行举例介绍：

- IIS 7 配置方案
- Apache 配置方案
- Nginx 配置方案

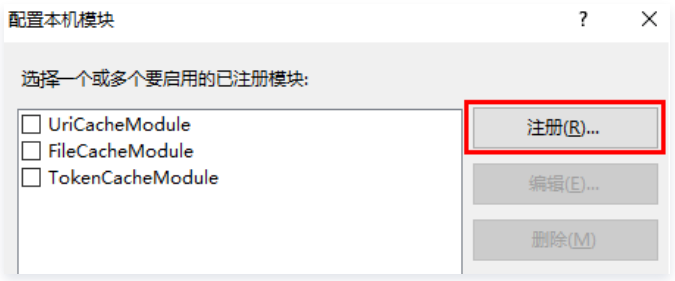
IIS 7 配置方案

1. 下载与安装插件 F5XForwardedFor 模块，根据自己的服务器操作系统版本将 x86\Release 或者 x64\Release 目录下的 F5XFFHttpModule.dll 和F5XFFHttpModule.ini 拷贝到某个目录，这里假设为 C:\F5XForwardedFor，确保 IIS 进程对该目录有读取权限。
2. 选择 **IIS 服务器**，双击**模块**功能。

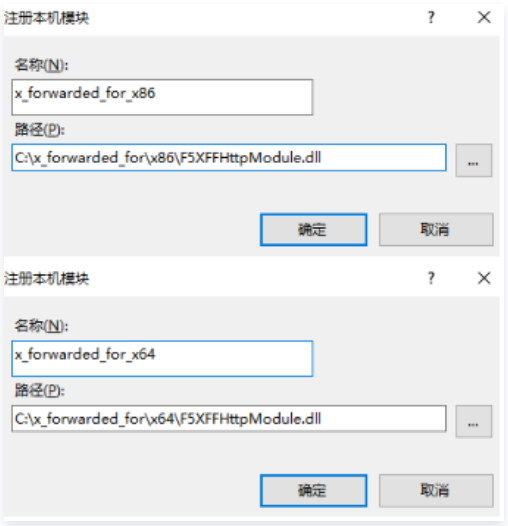


3. 单击**配置本机模块**。

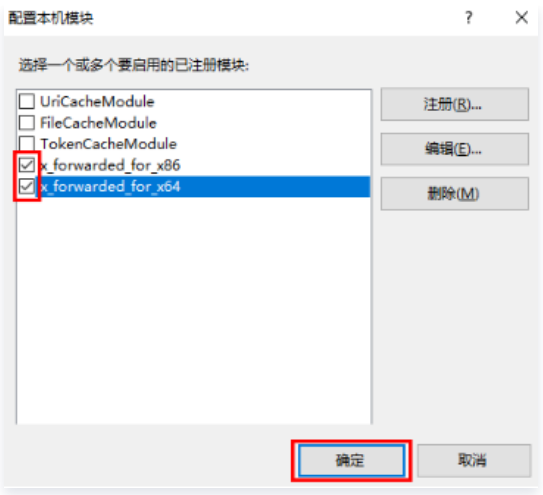
4. 在弹出框中单击**注册**。



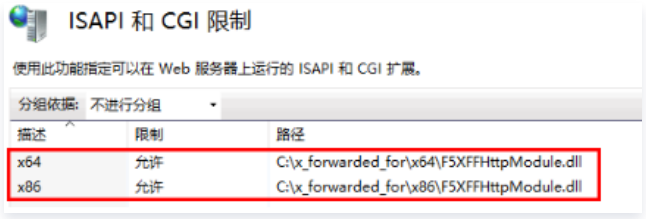
5. 添加下载的 DLL 文件，如下图所示：



6. 添加完成后，勾选并单击**确定**。



7. 在 IIS 服务器的 “ISAPI 和 CGI 限制”中，添加如上两个 DLL ，并将限制设置为允许。



8. 重启 IIS 服务器，等待配置生效。

Apache 配置方案

1. 安装 Apache 第三方模块“mod_rpaf”，需执行如下命令：

```
wget http://stderr.net/apache/rpaf/download/mod_rpaf-0.6.tar.gz
tar zxvf mod_rpaf-0.6.tar.gz
cd mod_rpaf-0.6
/usr/bin/apxs -i -c -n mod_rpaf-2.0.so mod_rpaf-2.0.c
```

2. 修改 Apache 配置 /etc/httpd/conf/httpd.conf，需在最末尾添加：

```
LoadModule rpaf_module modules/mod_rpaf-2.0.so
RPAFenable On
RPAFsethostname On

RPAFproxy_ips IP地址    //IP 地址为通道的转发IP
RPAFheader X-Forwarded-For
```

3. 添加完成后，重启 Apache。

```
/usr/sbin/apachectl restart
```

Nginx 配置方案

1. 当 Nginx 作为服务器时，获取客户端真实 IP，需使用 http_realip_module 模块，默认安装的 Nginx 是没有编译 http_realip_module 模块的，需要重新编译 Nginx，在configure 增加 --with-http_realip_module 选项， 确保 http_realip_module 模块编译进 nginx 中 。编译代码如下：

```
wget  http://nginx.org/download/nginx-1.14.0.tar.gz
tar  zxvf nginx-1.14.0.tar.gz
```

```
cd nginx-1.14.0
./configure --user=www --group=www --with-http_stub_status_module --without-http-cache --with-
http_ssl_module --with-http_realip_module
makemake install
```

2. 修改 nginx.conf。

```
vi /etc/nginx/nginx.conf
修改如下红色部分：
fastcgi connect_timeout 300;
fastcgi send_timeout 300;
fastcgi read_timeout 300;
fastcgi buffer_size 64k;
fastcgi buffers 4 64k;
fastcgi busy_buffers_size 128k;
fastcgi temp_file_write_size 128k;
set_real_ip_from IP地址;   //IP 地址为通道转发IP
real_ip_header X-Forwarded-For;
```

3. 重启 Nginx。

```
service nginx restart
```

全球统一域名接入

最近更新时间：2024-06-21 17:22:11

新建统一域名

1. 登录 全球应用加速控制台，进入“统一域名”页面，单击**新建**。
2. 配置“新建统一域名”信息。

- 所属项目：该统一域名所属项目（后续可以更换项目）。
- 域名名称：最多30个字符，支持中文。
- 默认入口：未加速地区的访问地址，通常为源站的 IP 地址；支持配置多个域名，以分号隔开。

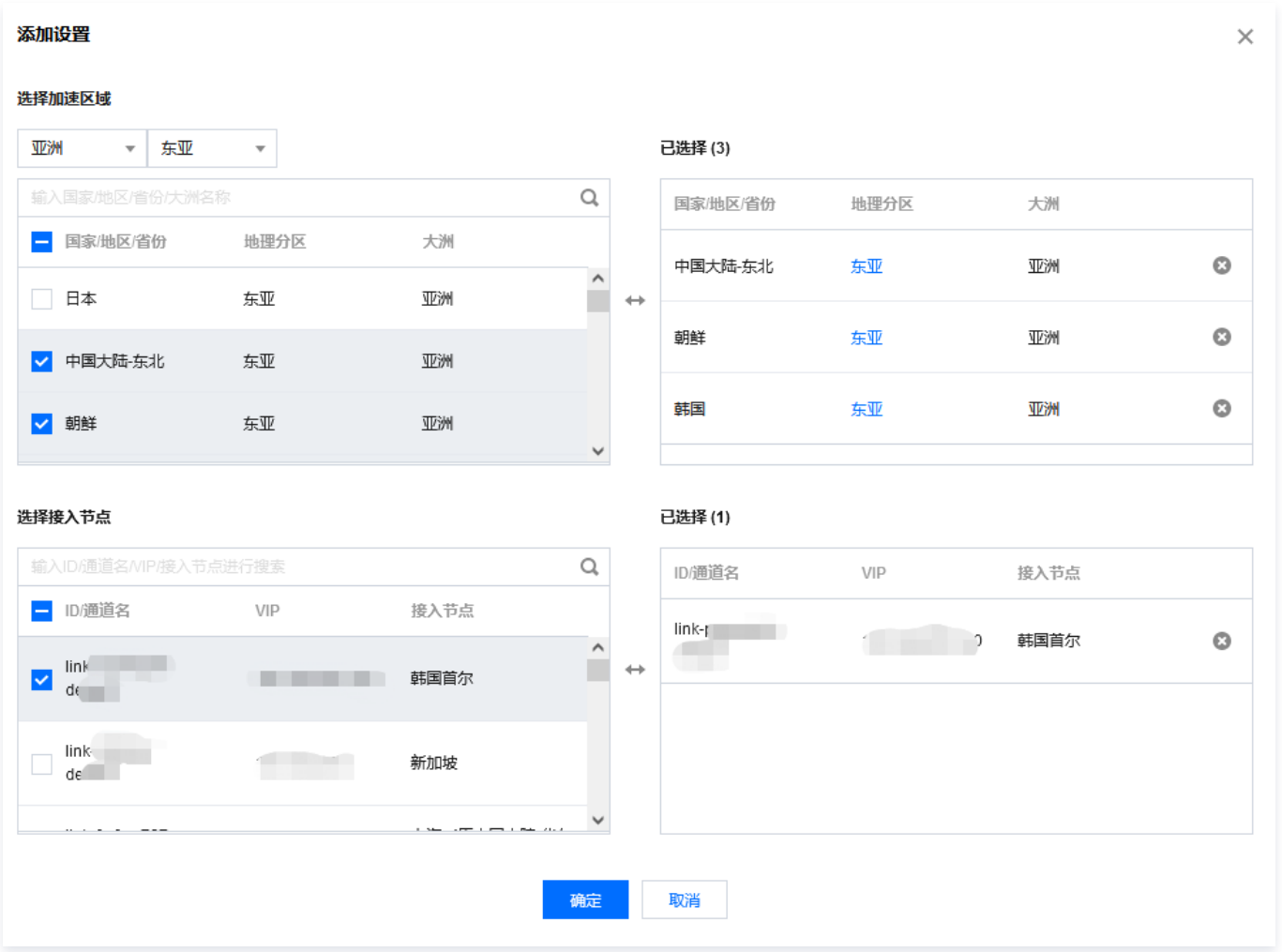
> ⚠ **注意**
> 统一域名默认数量为5个，如需添加请联系专员。

配置接入区域

1. 进入“统一域名”页面，单击指定域名或单击该域名对应的**设置**，进入下一级页面。

- 域名：用于客户端访问的域名。
- 默认入口：未加速地区的访问IP地址，通常为源站的IP地址。
- 通道数量：该统一域名下的通道数量。
- 状态：仅“开启”状态下，统一域名才可以正常使用。

2. 单击**添加设置**弹出“添加设置”弹窗，在列表中选择加速区域，以及选择接入节点，即为指定区域选择对应的加速通道，确认后即可生效。如下图所示：



配置默认入口

启用全球统一域名接入后，为避免出现因绕行导致的不必要网络延时及额外流量费用，您需对以下几种用户的访问路径进行额外配置：

1. 位于源站及源站附近用户：源站及附近区域用户一般只需通过公网进行直连，无需使用加速通道访问源站。
2. 远离加速通道入口区域用户：此类用户一般通过公网直连源站，接入加速通道效果不明显。

在“统一域名”页面中单击指定域名或单击该域名对应**设置**选项，进入下级页面。之后单击上方“默认入口”右侧**编辑**图标，即可对“默认入口”进行配置。当您对该入口 IP 地址进行配置后，除选定加速通道覆盖区域用户外，其他区域用户将通过公网直连该 IP 地址至您的源站。

国家与地区映射关系

最近更新时间：2024-10-28 16:46:21

由于世界各国疆域面积各异，为了方便数据展示及加速点广泛覆盖，全球应用加速对一些距离相近的国家进行加速区域合并。同时，对一些疆域面积较大的国家，全球应用加速进行区域拆分，以提高用户使用体验。当您使用“全球统一域名接入”功能，选择加速点覆盖区域时，可参照下表的国家与地区映射关系，配置全球加速点的覆盖区域。

大洲	地理分区	国家/地区	省/州
亚洲	东亚	中国大陆-华东	山东、江苏、安徽、浙江、江西、福建、上海
		中国大陆-华南	广东、广西、海南
		中国大陆-华中	湖北、湖南、河南
		中国大陆-华北	北京、天津、河北、山西、内蒙古
		中国大陆-西北	宁夏、新疆、青海、陕西、甘肃
		中国大陆-西南	四川、云南、贵州、西藏、重庆
		中国大陆-东北	辽宁、吉林、黑龙江
		蒙古国	
		朝鲜	
		韩国	
		日本	
	东南亚	文莱	
		中国澳门	
		柬埔寨	
		东帝汶	
		印度尼西亚	
		老挝	
		马来西亚	
		缅甸	
		菲律宾	
		中国香港	
		新加坡	
		中国台湾	
		泰国	
		越南	
	南亚	孟加拉	
		不丹	
		印度	

		马尔代夫	
		尼泊尔	
		巴基斯坦	
		斯里兰卡	
	中亚	哈萨克斯坦	
		吉尔吉斯斯坦	
		塔吉克斯坦	
		土库曼斯坦	
		乌兹别克斯坦	
	西亚	阿富汗	
		伊拉克	
		伊朗	
		叙利亚	
		约旦	
		黎巴嫩	
		以色列	
		巴勒斯坦	
		沙特阿拉伯	
		巴林	
		卡塔尔	
		科威特	
		阿联酋	
		阿曼	
		也门	
		格鲁吉亚	
		亚美尼亚	
		阿塞拜疆	
		土耳其	
		塞浦路斯	
欧洲	北欧	芬兰	
		瑞典	
		挪威	
		冰岛	
		丹麦	

	法罗群岛	
东欧	爱沙尼亚	
	拉脱维亚	
	白俄罗斯	
	立陶宛	
	乌克兰	
	摩尔多瓦	
中欧	波兰	
	捷克	
	斯洛伐克	
	匈牙利	
	德国	
	奥地利	
	瑞士	
	列支敦士登	
西欧	英国	
	爱尔兰	
	荷兰	
	比利时	
	卢森堡	
	法国	
	摩纳哥	
南欧	罗马尼亚	
	保加利亚	
	塞尔维亚	
	马其顿	
	阿尔巴尼亚	
	希腊	
	斯洛文尼亚	
	克罗地亚	
	波斯尼亚和墨塞哥维那	
	意大利	
	梵蒂冈	
	圣马力诺	

		马耳他	
		西班牙	
		葡萄牙	
		安道尔	
非洲	北非	埃及	
		利比亚	
		苏丹	
		突尼斯	
		阿尔及利亚	
		摩洛哥	
		马德拉群岛	
	东非	埃塞俄比亚	
		厄立特里亚	
		索马里	
		吉布提	
		肯尼亚	
		坦桑尼亚	
		乌干达	
		卢旺达	
		布隆迪	
		塞舌尔	
	中非	乍得	
		中非	
		喀麦隆	
		赤道几内亚	
		加蓬	
		刚果（布）	
		刚果（金）	
		圣多美及普林西比	
	西非	毛里塔尼亚	
		塞内加尔	
		冈比亚	
		马里	
		布基纳法索	

		几内亚	
		几内亚比绍	
		佛得角	
		塞拉利昂	
		利比里亚	
		科特迪瓦	
		加纳	
		多哥	
		尼日利亚	
		贝宁	
		尼日尔	
	南非	赞比亚	
		安哥拉	
		津巴布韦	
		马拉维	
		莫桑比克	
		博茨瓦纳	
		纳米比亚	
		南非	
		斯威士兰	
		莱索托	
		马达加斯加	
		科摩罗	
		毛里求斯	
		留尼旺	
大洋洲	大洋洲	澳大利亚	
		新西兰	
		巴布亚新几内亚	
		所罗门群岛	
		瓦努阿图	
		密克罗尼西亚	
		马绍尔群岛	
		帕劳	
		瑙鲁	

		基里巴斯	
		图瓦卢	
		萨摩亚	
		斐济群岛	
		汤加	
		库克群岛	
		关岛	
		新克里多尼亚	
		瓦利斯与富图纳	
		纽埃	
		托克劳	
		美属萨摩亚	
		北马里亚纳	
北美洲	北美	美国东部	缅因州、新罕布什尔州、佛蒙特州、马萨诸塞州、罗得岛州、康涅狄格州、纽约州、宾夕法尼亚州、新泽西州、特拉华州、马里兰州、华盛顿哥伦比亚特区、弗吉尼亚州、西弗吉尼亚州、北卡罗来纳州、南卡罗来纳州、佐治亚州、佛罗里达州、肯塔基州、田纳西州、密西西比州、亚拉巴马州
		美国西部	爱达荷州、蒙大拿州、怀俄明州、内华达州、犹他州、科罗拉多州、亚利桑那州、新墨西哥州、阿拉斯加州、华盛顿州、俄勒冈州、加利福尼亚州、夏威夷州
		美国中部	威斯康星州、密歇根州、伊利诺伊州、印第安纳州、俄亥俄州、密苏里州、北达科他州、南达科他州、内布拉斯加州、堪萨斯州、明尼苏达州、艾奥瓦州、俄克拉何马州、得克萨斯州、阿肯色州、路易斯安那州
		墨西哥	
		格陵兰	
	中美洲	危地马拉	
		伯利兹	
		萨尔瓦多	
		洪都拉斯	
		尼加拉瓜	
		哥斯达黎加	
		巴拿马	
	加勒比海区	巴哈马	
		古巴	
		牙买加	
		海地	
		多米尼加	
		安提瓜和巴布达	

		圣基茨和尼维斯	
		多米尼克	
		圣卢西亚	
		圣文森特和格林纳丁斯	
		格林纳达	
		巴巴多斯	
		特立尼达和多巴哥	
		波多黎各	
		英属维尔京群岛	
		美属维尔京群岛	
		安圭拉	
		蒙特塞拉特	
		瓜德罗普	
		马提尼克	
		荷兰加勒比区	
		阿鲁巴	
		特克斯和凯科斯群岛	
		开曼群岛	
		百慕大	
南美洲	南美洲北部	哥伦比亚	
		委内瑞拉	
		圭亚那	
		法属圭亚那	
		苏里南	
	南美洲中西部	厄瓜多尔	
		秘鲁	
		玻利维亚	
	南美洲东部	巴西	
	南美洲南部	智利	
		阿根廷	
		乌拉圭	
		巴拉圭	

配置权限

最近更新时间：2024-09-13 17:24:02

拥有 GAAP 权限的主账号或其他账号（AdministratorAccess 权限账号），可通过配置访问管理权限，使协作者账号拥有 GAAP 全读写或只读访问权限。
用户可通过策略关联用户、用户关联策略两种方式为协作者账号进行授权。更多信息，请参见 访问管理 CAM 。

准备步骤

1. 使用拥有 GAAP 权限的主账号或其他账号（AdministratorAccess 权限账号），登录 腾讯云控制台 。
2. 在顶部导航中，选择**云产品** > **管理与审计** > 访问管理，进入访问管理控制台。

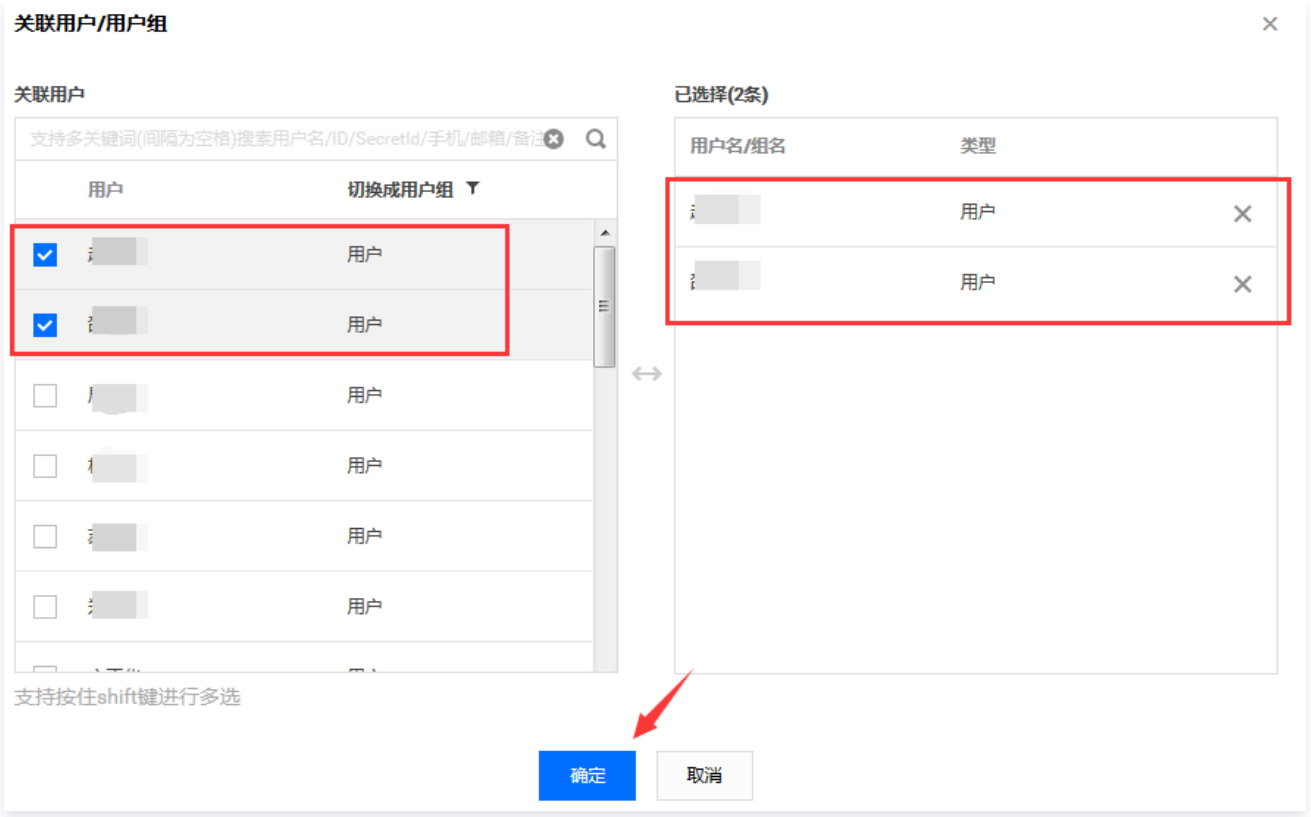
> **说明：**
> 您也可以在腾讯云控制台右上角，选择**您的账户名称** > **访问管理**，进入访问管理控制台。

操作步骤

策略关联用户

1. 在左侧菜单中，单击**策略**，进入管理页面。
2. 在搜索栏中，检索“GAAP”，找到2条结果。选择策略权限，单击**关联用户/组**。

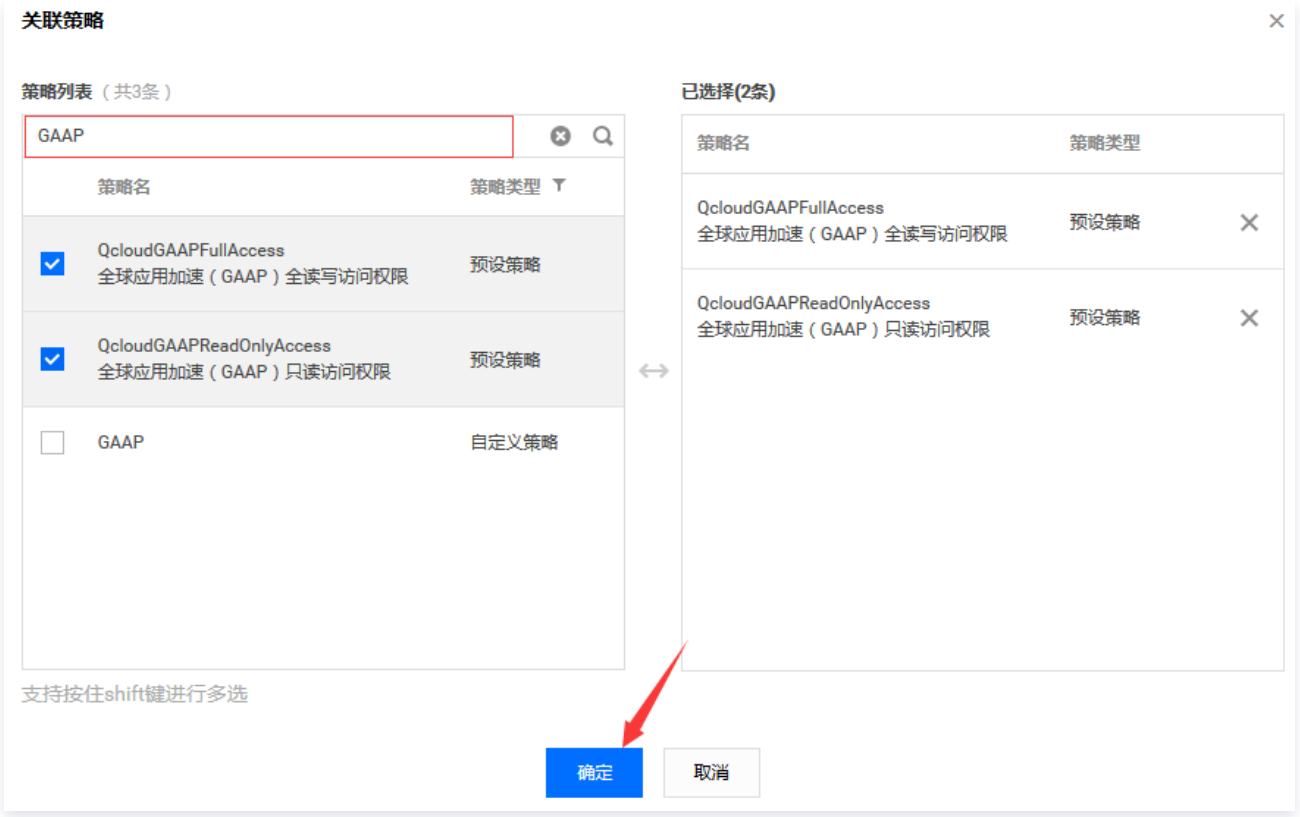
3. 勾选需要授权的用户，单击**确定**，即授权成功。



用户关联策略

1. 在左侧菜单中，选择**用户** > **用户列表**，进入管理页面。
2. 在列表中，找到需要授权的用户所在行，单击操作栏中的**授权**。

3. 在关联列表中，检索“GAAP”，勾选策略权限，单击**确定**，即授权成功。



查看和解除权限

授权成功的用户，可在 用户列表 中，单击用户名称，查看权限、解除权限。

全球加速2.0
加速区域

最近更新时间：2025-06-30 17:53:02

加速区域概述

加速区域是最靠近终端客户的区域，终端用户通过 GA 加速区域的接入 IP 就近接入腾讯云全球加速网络。

说明：
- 如终端用户区域不在 GA 支持区域列表内，可选择就近区域实现覆盖。
- 全球加速跨境段由中国联通代运营，如加速区域和终端节点组存在跨境，您的账号需要先通过 跨境资质审核，详情可参见 跨境云专线服务协议 。
- 单次创建，最多添加10个加速地域。

支持的加速区域

地区	包含的区域
中国大陆	北京
	上海
	广州
	成都
	重庆
	南京
港澳台地区	中国香港
亚太	新加坡
	印尼雅加达
	韩国首尔
	日本东京
	泰国曼谷
欧洲	德国法兰克福
北美	美国西部硅谷
	美国东部弗吉尼亚
南美	巴西圣保罗

加速 IP 类型

加速 IP 类型分为常规 BGP IP 及精品 BGP IP，您可按需选择加速 IP 类型。
- 常规 BGP IP：国内多线 BGP 网络覆盖超过二十家网络运营商（三大运营商、教育网、广电等），BGP 公网出口支持秒级跨域切换，保证您的用户无论使用哪种网络，均能享受高速、安全的网络质量。
- 精品 BGP IP：专属线路，避免绕行国际运营商出口网络；延时更低，可有效提升境外业务对中国大陆用户覆盖质量。

说明：
- 常规 BGP IP：对于中国港澳台地区和其他国家地域的常规 BGP IP，主要面向从中国大陆境外发起的访问；如果您在中国大陆境内访问境外的 IP，可能会有较多的延迟和丢包，

- 精品 BGP IP：延时更低，适用于中国大陆境内访问境外 IP，且对网络质量敏感的业务场景。目前仅中国香港区域支持选择精品 BGP IP 作为接入 IP。

IP 协议栈

当前仅支持 IPv4 协议接入。

加速区域带宽

- 加速地域带宽是创建加速区域时配置的带宽上限，实际业务带宽无法突破该带宽上限，允许配置的范围是2-1000Mbps，如需突破请 提交工单 咨询。
- 加速地域带宽上限与计费无直接关联，最终会统计实际业务流量作为计费依据。
- 带宽峰值仅作为带宽最高上限峰值，不作为承诺指标。当出现带宽资源争抢时，带宽峰值可能会受到限制。

添加加速区域

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击**加速区域**页签下的**添加加速区域**。（注：单次添加最多添加10个加速地域）
4. 根据提示配置对应信息。

配置项	说明
区域	所创建的加速区域，全球加速将会在所选区域创建加速IP供终端用户接入。
带宽峰值	用户从该区域接入全球加速可达到的最大业务带宽峰值，带宽峰值仅决定业务带宽上限，与最终费用没有直接关系，创建后将根据实际业务流量统计费用。单位：Mbps。 • 每个加速区域支持分配的带宽范围为2-1000Mbps。 • 带宽峰值仅作为带宽最高上限峰值，不作为承诺指标。当出现带宽资源争抢时，带宽峰值可能会受到限制。
IP 地址协议	终端用户接入全球加速的 IP 地址协议，当前仅支持 IPv4 网络接入。
公网质量类型	终端用户接入全球加速的公网质量类型。 • 普通 BGP：使用腾讯云普通 BGP IP 作为接入 IP，同时接入多家运营商线路，保障最优访问质量。 • 精品 BGP：使用运营商精品公网线路接入，相比普通 BGP，中国大陆用户接入质量更优。选择中国香港作为加速区域时，可选择加速 IP 公网质量类型为精品 BGP。

编辑加速区域

前提条件

已完成实例及加速区域的创建。您可对已有加速区域进行带宽编辑操作。

> ⚠ **注意：**
> 如修改后的带宽上限低于修改前，可能造成业务受损，请谨慎操作。

操作指南

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页，并单击**加速区域**页签。
3. 单击已有加速区域右侧操作栏的**编辑带宽**。
4. 在弹窗中输入目标带宽上限，单击**确定**。

删除加速区域

前提条件

已完成实例及加速区域创建。加速区域删除后，对应加速IP及带宽配置将被释放，全球加速将无法再为该区域提供加速服务。

操作指南

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页，并单击**加速区域**页签。
3. 单击已有加速区域右侧操作栏的**删除**。
4. 在弹窗中单击**确定**。

> ⚠ **注意：**
> 加速区域删除后加速 IP 将被释放，相关配置无法恢复，请充分确认影响后操作。

监听器
配置 TCP 和 UDP 监听器

最近更新时间：2025-06-26 10:02:01

监听器概述

创建全球加速实例后，您需要为实例配置监听器。监听器负责监听客户端请求，并将流量分发至后端终端节点上。
全球加速监听器需配置：
1. 监听协议和监听端口。监听器的监听端口，亦被称为前端端口，用来接收请求并向后端服务器转发请求的端口。
2. 监听策略，如均衡策略、会话保持 等。
3. 添加终端节点组。需创建终端节点组并添加终端节点。

支持的协议类型

全球加速支持监听来自客户端的四层和七层请求，并将这些请求分发到后端终端节点上，而后由后端终端节点处理请求。四层和七层监听器的区别主要体现在：当用户请求到来时，是依据四层协议还是七层协议来进行转发流量，例如：对 TCP、UDP 等四层协议请求进行四层转发，对 HTTP、HTTPS 等七层协议请求进行七层转发。
四层协议：传输层协议，主要通过 VIP + Port 接收请求并分配流量到后端服务器。
七层协议：应用层协议，基于 URL、HTTP 头部等应用层信息进行流量分发。
腾讯云全球加速支持以下协议的请求转发：

- TCP（传输层）
- UDP（传输层）
- HTTP（应用层）
- HTTPS（应用层）

协议分类	协议	说明	应用场景
四层协议	TCP	面向连接的、可靠的传输层协议。 传输的源端和终端需先三次握手建立连接，再传输数据。 支持基于客户端 IP（源 IP）的会话保持。 支持获取客户端源IP。	适用于对可靠性和数据准确性要求高、对传输速度要求较低的场景，如文件传输、收发邮件、远程登录等。
	UDP	无连接的传输层协议。 传输的源端和终端不建立连接，不需维护连接状态。 每一条 UDP 连接都只能是点到点的。 支持一对一，一对多，多对一和多对多的交互通信。 支持基于客户端 IP（源 IP）的会话保持。	适用于对传输效率要求高、对准确性要求相对较低的场景，如即时通讯、在线视频等。
七层协议	HTTP	应用层协议。 支持基于请求域名和 URL 的转发。	需要对请求的内容进行识别的应用，例如 Web 应用、App 服务等。详情请参见 配置 HTTP 和 HTTPS 监听器 。
	HTTPS	加密的应用层协议。 支持基于请求域名和 URL 的转发。 统一的证书管理服务，可在全球加速控制台完成证书上传及替换。 支持单向认证和双向认证。	需加密传输的 HTTP 应用。详情请参见 配置 HTTP 和 HTTPS 监听器 。

支持的端口范围

端口类型	说明	限制
监听端口（前端端口）	监听端口是全球加速接收请求并向终端节点转发请求的端口。您可以配置的端口范围为1 - 64999。	在同一个全球加速实例内： UDP 类协议可以和 TCP 类协议的监听端口重复。例如，可以同时创建监听器 TCP:80 和监听器 UDP:80。 同一类协议下监听端口不可重复，TCP/TCP SSL/HTTP/HTTPS 同属于 TCP 类。例如，不可以同时创建监听器 TCP:80 和监听器 HTTP:80。

终端节点端口（后端端口）	七层监听器支持配置终端节点端口，终端节点端口是后端服务器提供服务的端口，接收并处理来自全球加速的流量。您可以配置的终端节点端口范围为1 - 64999。	在同一个全球加速实例内： 不同监听协议的服务端口可以重复。例如，监听器 HTTP:80 和监听器 HTTPS:443 可以同时绑定同一台后端服务器的同一个端口。
健康检查端口	健康检查端口用于全球加速向后端服务器发送探测请求，以确认服务器是否正常运行。若端口响应正常，则认为服务器健康。您可以配置的终端节点端口范围为1 - 64999。	-

您需要为全球加速实例创建监听器，用于监听用户请求及将流量转发到后端终端节点，全球加速 GA 支持 TCP、UDP、HTTP 及 HTTPS 协议，本章节为您介绍 TCP、UDP 监听器配置及操作指南。

操作指南

前提条件

已完成全球加速实例创建。

创建监听器

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**添加监听器**。
4. 配置监听器。

配置类型	配置项	说明
基础配置	监听器名称	• 以大小写字母或中文开头 • 长度 2-128 字符 • 支持数字、英文句号 “.” 或短划线 “-”、下划线 “_”。
	路由类型	• 智能路由：根据延时选择最近终端节点组进行转发
	协议	支持选择 TCP、UDP、HTTP、HTTPS • TCP（传输控制协议）：面向连接、可靠传输（确认/重传机制）、保证数据顺序，适用于网页浏览（HTTP/HTTPS）、文件传输（FTP）、电子邮件（SMTP）等对可靠性要求高的场景。 • UDP（用户数据报协议）：无连接、不可靠传输、低延迟，适用于实时音视频（如 VoIP）、在线游戏、DNS 查询等对速度敏感且允许丢包的场景。
	端口	支持端口范围为 1-64999
高级配置	获取客户端源 IP	• TCP 协议：开启后，可通过 ProxyProtocol 代理协议获取客户端真实 IP。 • UDP 协议：不支持获取客户端源 IP。
	会话保持	• 开启：来自同一个 IP 的用户请求保持访问相同源站。 • 关闭：无法保障来自同一个 IP 的用户请求保持访问相同源站。
	连接空闲超时时间	指定连接空闲超时时间。在超时时间内一直没有数据交互，全球加速会中断当前连接，直到下一次请求来临时重新建立新的连接。监听协议不同，取值范围不同。 • TCP监听：取值范围为10-900秒，默认值为900秒。 • UDP监听：取值范围是10-20秒，默认是20秒。

5. 配置终端节点组

监听器创建时，您可以为监听器创建默认终端节点组，来承接监听器转发到后端的流量。配置终端节点组时，您需要为节点组添加终端节点并按需开启健康检查。

> **说明：**
> 监听器首次创建时配置的节点组为默认终端节点组，TCP 和 UDP 监听器仅支持创建一个默认终端节点组，不支持创建自定义终端节点组。

配置类型	配置项	说明
终端节点组	节点组名称	• 以大小写字母或中文开头 • 长度 2-128 字符 • 支持数字、英文句号 “.” 或短划线 “-”、下划线 “_”。
	地域	终端节点组所在地域，全球加速会将来自加速区域的流量转发到终端节点组地域。 **注意：** 如加速区域与终端节点组属于同一地域，可能导致加速效果不佳。
	后端服务类型	终端节点是最终提供服务的后端源站，终端节点类型支持自定义域名及自定义 IP。
	后端服务	最终提供服务的后端源站，您可为一个终端节点组最多添加4个终端节点，支持输入自定义 IP 或自定义域名。例如： • `10.1.1.1` • `192.168.0.0` • `1.1.1.1` • `example.com`
	权重	终端节点权重，全球加速将按照您配置的终端节点权重来分发业务流量到后端服务器。 • 默认值：100 • 配置范围：1-100
	健康检查	• 开启：全球加速将按配置的健康检查参数来检查后端源站的可用性。 • 关闭：全球加速不对源站进行健康检查探测。

	检查协议	全球加速用于检测后端服务器是否可用的网络协议
		• TCP：支持 TCP 与自定义探测
		  ○ TCP：全球加速通过 TCP 协议检测后端服务器是否可用。
		  ○ 自定义探测：通过手动配置健康检查的检查请求、检查返回结果来对后端服务器进行检测。
		• UDP：支持 PING 与自定义探测
		  ○ PING：全球加速通过 PING 协议检测后端服务器是否可用。
		  ○ 自定义探测：通过手动配置健康检查的检查请求、检查返回结果来对后端服务器进行检测。

	响应超时时间	全球加速向后端服务器发送健康检查请求后，等待服务器响应的最长时间。若超时未收到响应，则判定本次检查失败。 • 默认值：2s • 配置范围：2s-60s
	健康检查间隔	两次健康检查之间的时间间隔。 • 默认值：30s • 配置范围：5s-300s。
	不健康阈值	连续健康检查失败的次数达到该阈值后，后端服务器被标记为不健康，并从流量分发池中移除。 • 默认值：3次 • 配置范围：1次-10次。
	健康阈值	连续健康检查成功的次数达到该阈值后，不健康的服务器被重新标记为健康并恢复流量分发。 • 默认值：3次

		• 配置范围：1次-10次。

编辑监听器

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击**监听器**，进入监听器列表页。
4. 单击**实例 ID**或操作下的**实例详情**，可进入监听器实例详情页。
5. 在已有监听器操作下的**配置管理**下拉列表，单击管理终端节点组，进入终端节点组列表页，对终端节点组进行管理。

删除监听器

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击**监听器**，进入监听器列表页。
4. 单击监听器右侧的**删除**。
5. 在弹出提示框中单击**确认**，完成删除。

> **注意：**
> 监听器删除后，将释放组内所有监听和终端节点组的绑定关系，业务将不再进行加速，且删除后无法恢复和访问，请充分确认影响后再进行删除。

配置 HTTP 和 HTTPS 监听器

最近更新时间：2025-06-26 10:02:01

监听器概述

创建全球加速实例后，您需要为实例配置监听器。监听器负责监听客户端请求，并将流量分发至后端终端节点上。
全球加速监听器需配置：
1. 监听协议和监听端口。监听器的监听端口，亦被称为前端端口，用来接收请求并向后端服务器转发请求的端口。
2. 监听策略。如均衡策略、会话保持 等。
3. 添加终端节点组。需创建终端节点组并添加终端节点。

支持的协议类型

全球加速支持监听来自客户端的四层和七层请求，并将这些请求分发到后端终端节点上，而后由后端终端节点处理请求。四层和七层监听器的区别主要体现在：当用户请求到来时，是依据四层协议还是七层协议来进行转发流量，例如：对 TCP、UDP 等四层协议请求进行四层转发，对 HTTP、HTTPS 等七层协议请求进行七层转发。
四层协议：传输层协议，主要通过 VIP + Port 接收请求并分配流量到后端服务器。
七层协议：应用层协议，基于 URL、HTTP 头部等应用层信息进行流量分发。
腾讯云全球加速支持以下协议的请求转发：

- TCP（传输层）
- UDP（传输层）
- HTTP（应用层）
- HTTPS（应用层）

协议分类	协议	说明	应用场景
四层协议	TCP	面向连接的、可靠的传输层协议。 传输的源端和终端需先三次握手建立连接，再传输数据。 支持基于客户端 IP（源 IP）的会话保持。 支持获取客户端源IP。	适用于对可靠性和数据准确性要求高、对传输速度要求较低的场景，如文件传输、收发邮件、远程登录等。详情请参见 配置 TCP 和 UDP 监听器 。
	UDP	无连接的传输层协议。 传输的源端和终端不建立连接，不需维护连接状态。 每一条 UDP 连接都只能是点到点的。 支持一对一，一对多，多对一和多对多的交互通信。 支持基于客户端 IP（源 IP）的会话保持。	适用于对传输效率要求高、对准确性要求相对较低的场景，如即时通讯、在线视频等。详情请参见 配置 TCP 和 UDP 监听器 。
七层协议	HTTP	应用层协议。 支持基于请求域名和 URL 的转发。	需要对请求的内容进行识别的应用，例如 Web 应用、App 服务等。
	HTTPS	加密的应用层协议。 支持基于请求域名和 URL 的转发。 统一的证书管理服务，可在全球加速控制台完成证书上传及替换。 支持单向认证和双向认证。	需加密传输的 HTTP 应用。

支持的端口范围

端口类型	说明	限制
监听端口（前端端口）	监听端口是全球加速接收请求并向终端节点转发请求的端口。您可以配置的端口范围为1 - 64999。	在同一个全球加速实例内： UDP 类协议可以和 TCP 类协议的监听端口重复。例如，可以同时创建监听器 TCP:80 和监听器 UDP:80。 同一类协议下监听端口不可重复，TCP/TCP SSL/HTTP/HTTPS 同属于 TCP 类。例如，不可以同时创建监听器 TCP:80 和监听器 HTTP:80。

终端节点端口（后端端口）	七层监听器支持配置终端节点端口，终端节点端口是后端服务器提供服务的端口，接收并处理来自全球加速的流量。您可以配置的终端节点端口范围为1 - 64999。	在同一个全球加速实例内： 不同监听协议的服务端口可以重复。例如，监听器 HTTP:80 和监听器 HTTPS:443 可以同时绑定同一台后端服务器的同一个端口。
健康检查端口	健康检查端口用于全球加速向后端服务器发送探测请求，以确认服务器是否正常运行。若端口响应正常，则认为服务器健康。您可以配置的终端节点端口范围为1 - 64999。	-

您需要为全球加速实例创建监听器，用于监听用户请求及将流量转发到后端终端节点，全球加速 GA 支持 TCP、UDP、HTTP 及 HTTPS 协议，本章节为您介绍 HTTP 及 HTTPS 监听器配置及操作指南。

操作指南

前提条件

已完成全球加速实例创建。

创建监听器

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**添加监听器**。
4. 配置监听器。

配置类型	配置项	说明
基础配置	监听器名称	• 以大小写字母或中文开头 • 长度 2-128 字符 • 支持数字、英文句号 “.” 或短划线 “-”、下划线 “_”。
	协议	支持选择 TCP、UDP、HTTP、HTTPS • HTTP（超文本传输协议）：应用层协议，明文传输、无加密，适用于普通网页浏览、数据抓取等非敏感信息传输。 • HTTPS（安全超文本传输协议）：HTTP+SSL/TLS 加密，提供数据加密和身份认证，适用于在线支付、登录认证等需要安全传输的场景。
	端口	支持端口范围为 1-64999
	SSL 解析方式	HTTPS 监听器与客户端的认证方式。 • 单向认证：仅客户端验证服务端身份，服务端不验证客户端身份，选择该认证方式，仅需上传服务器证书到全球加速。 • 双向认证：客户端和服务端互相验证身份，客户端需提供证书供服务端验证。选择该认证方式时，需同时上传服务器证书及 CA 证书到全球加速。
	服务器证书	由 CA 机构颁发给网站的数字证书，用于验证服务器身份并建立加密连接。选择单向认证并完成上传后，全球加速会将该证书返回给客户端用于建立加密连接。
	客户端 CA 证书	由根 CA 或中间 CA 持有的证书，用于签发和验证服务器证书的合法性，上传后，全球加速将用该证书验证客户端的合法性。 **说明：** 仅在认证模式选择双向认证时需上传客户端 CA 证书。
	TLS 安全策略组	创建 HTTPS 监听器时支持按需选择不同 TLS 安全策略组（tls_policy_1.0-2、tls_policy_1.1-2、tls_policy_1.2、tls_policy_1.2_strict），不同策略组包含不同

		TLS 版本及加密算法套件，详情可参见 TLS 安全策略组 。
高级配置	获取客户端源 IP	开启后，默认携带 X-Forwarded-For、X-Forwarded-Ip、X-Forwarded-Proto、X-Real-IP 字段。
	连接空闲超时时间	指定连接空闲超时时间。在超时时间内一直没有数据交互，全球加速会中断当前连接，直到下一次请求来临时重新建立新的连接。 • 默认值：15s。 • 配置范围：1-60s。
	连接请求超时时间	指定连接请求超时时间，客户端与服务器建立连接所需的最大等待时间，如果超过这个时间仍未建立连接，则认为连接请求超时。 • 默认值：60s。 • 配置范围：1s-180s。

5. 配置终端节点组

监听器创建时，您可以为监听器创建默认终端节点组，来承接监听器转发到后端的流量。配置终端节点组时，您需要为节点组添加终端节点并按需开启健康检查。

> **说明：**
> 监听器首次创建时配置的节点组为默认终端节点组，HTTP/HTTPS 监听器支持创建自定义终端节点组。

配置类型	配置项	说明
终端节点组	节点组名称	• 以大小写字母或中文开头 • 长度 2-128 字符 • 支持数字、英文句号 “.” 或短划线 “-”、下划线 “_”。
	地域	终端节点组所在地域，全球加速会将来自加速区域的流量转发到终端节点组地域。 **注意：** 如加速区域与终端节点组属于同一地域，可能导致加速效果不佳。
	后端服务类型	终端节点是最终提供服务的后端源站，终端节点类型支持自定义域名及自定义 IP。
	后端服务	最终提供服务的后端源站，您可为一个终端节点组最多添加4个终端节点，支持输入自定义 IP 或自定义域名。例如： • `10.1.1.1` • `192.168.0.0` • `1.1.1.1` • `example.com`
	权重	终端节点权重，权重取值范围为：1-100。全球加速将按照您配置的终端节点权重来分发业务流量到后端服务器。
	回源协议	全球加速回源到终端节点时所使用的协议。 • 监听协议为 HTTP：回源协议仅支持 HTTP。 • 监听协议为 HTTPS：回源协议支持选择 HTTP 或 HTTPS。
	端口映射	支持配置监听端口与后端服务端口的映射关系。全球加速将根据配置转发数据包到终端节点对应端口。 • 监听端口：不支持修改，与监听器端口保持一致。 • 终端节点端口：支持修改配置范围为1-64999。
	健康检查	• 开启：全球加速将按配置的健康检查参数来检查后端源站的可用性。 • 关闭：全球加速不对源站进行健康检查探测。

检查协议	全球加速用于检测后端服务器是否可用的网络协议，对于 HTTP 和 HTTPS 监听器，均只支持通过 HTTP 协议进行健康检查。
响应超时时间	全球加速向后端服务器发送健康检查请求后，等待服务器响应的最长时间。若超时未收到响应，则判定本次检查失败。 • 默认值：2s • 配置范围：2s-60s
健康检查间隔	两次健康检查之间的时间间隔。 • 默认值：30s • 配置范围：5s-300s
不健康阈值	连续健康检查失败的次数达到该阈值后，后端服务器被标记为不健康，并从流量分发池中移除。 • 默认值：3次 • 配置范围：1次-10次
健康阈值	连续健康检查成功的次数达到该阈值后，不健康的服务器被重新标记为健康并恢复流量分发。 • 默认值：3次 • 配置范围：1次-10次
检查域名	指健康检查时请求的域名。
检查路径	指定健康检查的 URL 路径（如/checkHealth），全球加速会向该路径发送 HTTP 请求，根据返回状态码判断服务是否健康。
请求方式	支持 HEAD 或 GET 方法： • HEAD：仅请求响应头，轻量高效。 • GET：获取完整响应，适用于需检查内容完整性的场景。
状态监测码	健康检查通过 HEAD 或 GET 请求访问指定路径（如/health），若返回状态码在预设范围内且未超时，则标记服务为健康，否则触发隔离机制，支持配置以下状态监测码：http_2xx、http_3xx、http_4xx、http_5xx。

相关文档

- 证书管理
- 转发策略
- TLS安全策略组

终端节点组

最近更新时间：2025-05-16 17:48:13

概述

终端节点组（Endpoint Group）是指一组位于特定地域的终端节点（Endpoint）的集合，用于接收并处理通过腾讯云全球加速网络转发的客户端请求。其核心作用是将流量从加速入口（加速 IP）高效分发到后端服务（如 ECS、CLB 等），实现跨地域的低延迟访问。终端节点组的主要功能如下：

- 流量分发：全球加速监听器（Listener）根据路由规则（当前仅支持智能路由）将客户端请求转发到关联的终端节点组，再由终端节点组内的终端节点将请求送达后端服务。
- 地域关联：每个终端节点组绑定一个特定地域（如北京、上海），确保流量就近接入后端服务，减少网络延迟。
- 多节点容灾：一个终端节点组通常可添加至多4个终端节点，通过健康检查实现高可用，故障时自动切换。

> **说明：**
> 全球加速跨境段由中国联通代运营，如加速区域和终端节点组存在跨境，您的账号需要先通过 跨境资质审核 ，详情可参见 跨境云专线服务协议 。

终端节点组类型

终端节点组分为默认终端节点组及自定义终端节点组两种类型，首次创建监听器时创建的节点组即为默认终端节点组。

- 默认终端节点组：
  - TCP/UDP 监听：仅支持创建1个默认终端节点组，创建后将根据配置的健康检查策略将业务流量转发到健康的终端节点。
  - HTTP/HTTPS 监听：仅支持创建1个默认组。
- 自定义终端节点组：
  - 适用于HTTP/HTTPS监听，支持创建最多10个自定义节点组，可通过转发策略（如基于 URL）将部分流量定向到特定自定义节点组。

> **说明：**
> - 默认终端节点组创建后即绑定默认转发策略。
> - 仅 HTTP 和 HTTPS 监听器支持配置转发策略并绑定自定义终端节点组，详情可参见 转发策略 。

终端节点类型

终端节点（Endpoint）是指客户端请求最终到达并处理的后端服务实例，它是全球加速网络中将加速流量转发到实际业务服务器的代理节点。配置完成后全球加速通过公网将业务流量转发到终端节点。终端节点支持以下类型：

- 自定义公网 IP 地址
- 自定义域名

回源协议

回源协议是指全球加速将客户端请求转发到后端服务器（终端节点）时所使用的应用层协议。例如：客户端通过 HTTPS 访问负载均衡器，但负载均衡器回源时可能使用 HTTP。仅在七层监听器创建终端节点组时支持选择回源协议。

端口映射

端口映射是指全球加速服务在应用层（HTTP/HTTPS）将客户端请求的前端监听端口与后端服务器的实际服务端口进行关联和转发。

- 前端端口：客户端访问全球加速的端口（如80或443）。
- 后端端口：全球加速将请求转发到后端终端节点的端口（如8080或8443）。
- 映射关系：创建终端节点组时，通过配置端口映射将前端端口的请求定向到后端指定端口，实现协议转换或端口解耦。

> **说明：**
> 终端节点组配置端口映射时，监听端口需要与监听器端口保持一致，不支持修改。

健康检查

健康检查是全球加速服务中用于检测后端终端节点是否可用的机制。通过定期发送探测请求，判断后端服务的运行状态，确保流量只被分发到健康的节点，从而保障业务的高可用性。支持对终端节点组配置健康检查，详情可参见 配置HTTP和HTTPS监听器 。

相关文档

- 配置HTTP和HTTPS监听器
- 转发策略
- 证书管理

转发策略

最近更新时间：2025-06-25 20:03:21

概述

转发策略是指全球加速服务基于应用层（HTTP/HTTPS）的请求内容（如域名、URL 路径、请求头等）将流量智能分发到不同终端节点组的规则集合。您可通过创建转发策略并绑定自定义终端节点组，实现精细化的流量控制和路由。

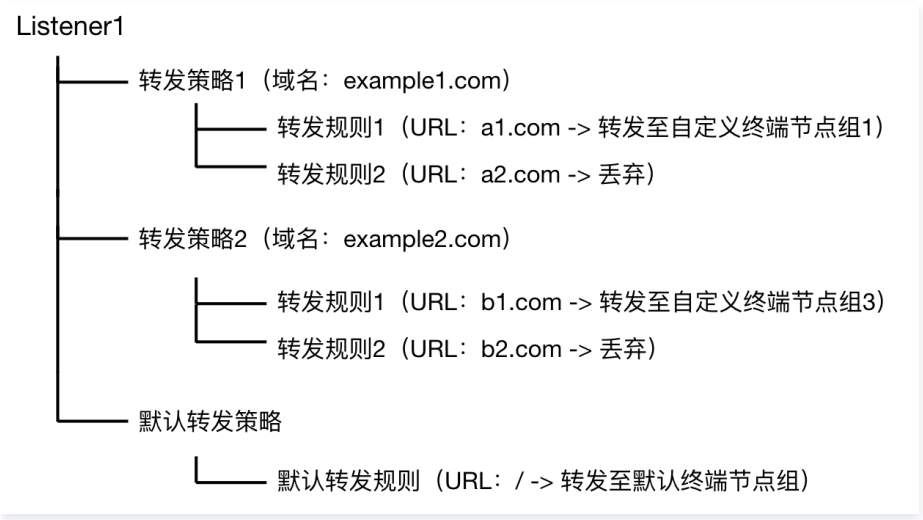
策略类型

转发策略类型包括默认策略与自定义策略：

- 默认策略：监听器创建时自动创建，与默认终端节点组自动关联，不支持编辑或者删除，且无法绑定其他自定义终端节点组。默认终端节点组删除后，默认策略将被联动删除。
- 自定义策略：用户可基于域名、路径定义精细化路由规则。自定义策略可绑定到自定义终端节点组实现流量分配，但无法绑定默认终端节点组。

策略组成

- 匹配域名：需要匹配的域名，一条转发策略对应一个域名匹配，基于域名添加匹配 URL。
- URL 路径：需要匹配的路径，一个域名下支持配置多条转发规则，一条转发规则对应一个 URL。
- 转发动作：命中转发策略后全球加速对应执行的策略动作，转发至策略绑定的自定义终端节点组或丢弃。
- 回源 HOST：全球加速向源站发起请求时，在 HTTP 头部 Host 字段中携带的域名标识，支持修改回源请求中的 HOST 字段，如不填写则使用默认 HOST。
- 回源 SNI：全球加速在 HTTPS 回源握手阶段告知源站请求的目标域名，仅 HTTPS 转发策略支持修改。



工作原理

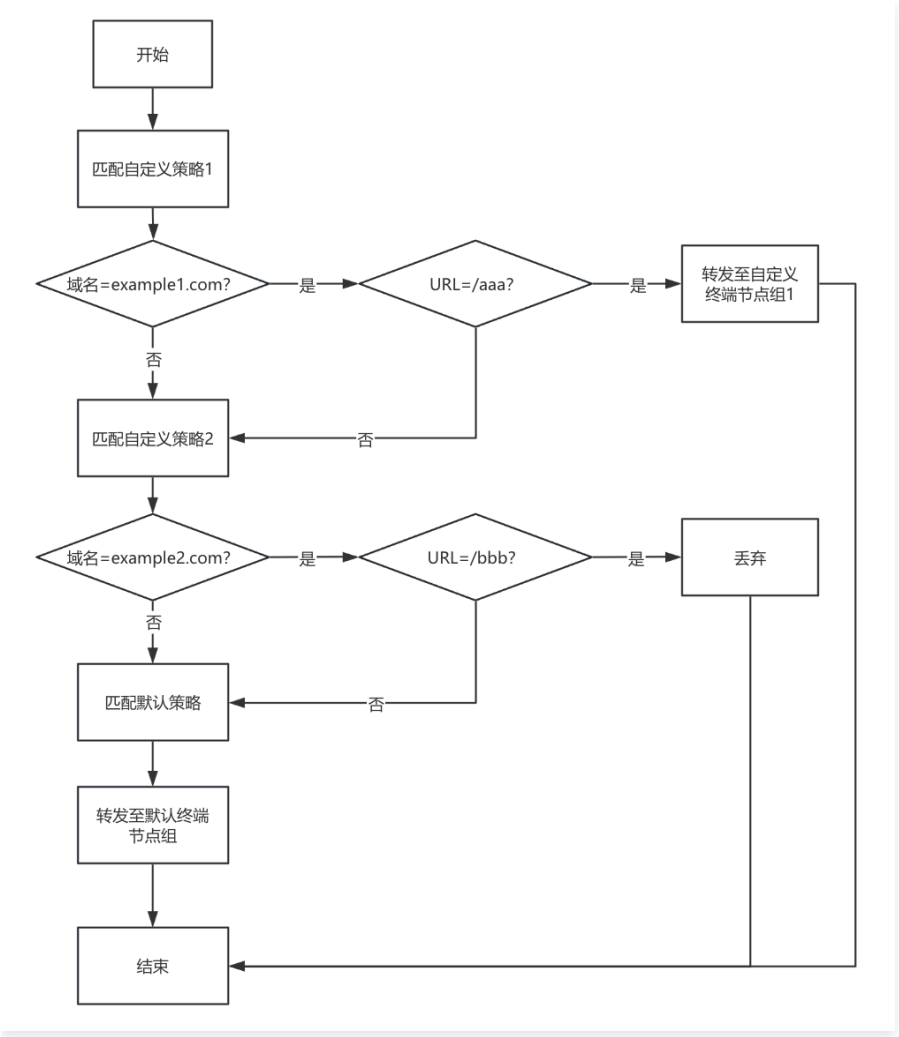
1. 请求解析

   全球加速节点接收用户请求后，解析 HTTP/HTTPS 头部、URL 路径、域名等信息。
2. 按优先级对自定义转发策略逐条匹配：

   条件组合：转发策略匹配条件包含域名+路径，需全部满足才会触发动作。
3. 动作执行

- 匹配成功：立即执行策略对应动作（转发至指定终端节点组或丢弃）。
- 匹配失败：继续匹配其余的自定义策略，如未匹配上任何自定义策略，最终按默认策略动作执行，将请求发送至默认终端节点组。

例如：
一个监听器下配置了两条自定义转发策略，自定义策略1匹配域名为 example1.com，匹配 URL 为/aaa，转发动作为 转发至自定义终端节点组1；自定义策略2匹配域名为 example2.com，匹配 URL 为/bbb，转发动作为 丢弃；则该监听器匹配流程如下：



配置转发策略

前提条件

已完成全球加速实例及 HTTP&HTTPS 监听器创建。

操作步骤

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**转发策略**进入转发策略页签，单击**添加 HTTP 转发策略**。
5. 在弹窗中输入希望匹配的域名。
6. 单击域名右侧的**添加转发规则**并按下表指引完成对应配置。

配置项	说明
域名	客户端访问加速服务时使用的域名（如www.example.com），全球加速会基于该域名匹配预设的转发规则。
URL	转发路径，必填，长度1-80，支持字符集如下：a-z A-Z 0-9 _ . - /。全球加速会基于该域名+URL 对业务流量进行精确匹配。
转发动作	流量命中转发规则配置的域名及 URL 后，全球加速会执行对应的转发动作。 • 转发至：将命中的流量转发至规则绑定的自定义终端节点组。 • 丢弃：全球加速将丢弃命中规则的流量，不进行转发。

回源 SNI	当全球加速以 HTTPS 协议回源时，通过 SNI 在 TLS 握手阶段明确告知源站请求的目标域名，源站据此返回对应的SSL 证书。对命中规则的流量，支持修改回源请求中的 SNI 字段。 **说明：** 仅 HTTPS 监听器支持通过 HTTPS 协议回源。
回源 HOST	全球加速向源站发起请求时，会在 HTTP 头部 Host 字段中携带域名标识。对命中规则的流量，支持修改回源请求中的 HOST 字段。如不填写则使用默认 HOST。
回源请求头	支持在 HTTPS 转发规则中配置回源请求头，用于自定义全球加速向源站发起请求时携带的 HTTP 头部信息。

编辑转发策略

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**转发策略**进入转发策略页签。
5. 单击已有规则右侧的**编辑转发规则**进行修改

删除转发策略

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**转发策略**进入转发策略页签。
5. 单击已有规则右侧的**编辑转发规则**进行修改。

说明：
默认转发策略不支持编辑或删除，默认终端节点组删除后，默认策略联动删除，可通过再次创建默认终端节点组来恢复默认策略。

相关文档

- 配置HTTP和HTTPS监听器
- 证书管理

证书管理

最近更新时间：2025-06-25 20:03:21

概述

HTTPS 证书（也称为 SSL/TLS 证书）是一种由受信任的证书颁发机构（CA）签发的数字证书，用于验证网站身份并启用加密连接。它通过 SSL/TLS 协议在客户端（如浏览器）和服务器之间建立安全通道，确保数据传输的机密性（防止窃听）、完整性（防止篡改）和真实性（防止冒充）。证书包含网站的公钥、域名、颁发机构信息及有效期。使用全球加速创建 HTTPS 监听器时，您需要进行证书上传与管理。

认证模式

您可按需为全球加速监听器实例选择 HTTPS 认证模式，支持单向认证及双向认证，单向认证和双向认证的核心区别在于身份验证的方向和严格程度。

- 单向认证：仅客户端验证服务器身份，服务器不验证客户端，该认证模式下，您仅需上传服务器证书到全球加速。适用普通网站浏览、电商平台等公开服务，用户无需预先配置证书。
- 双向认证：客户端和服务器互相验证身份，该认证模式下，您需同时上传服务器证书及客户端证书到全球加速。适用企业内网、金融系统、医疗数据交换等高安全需求场景，仅允许持有合法证书的客户端访问。

对比项	单向认证	双向认证
验证方	仅客户端验证服务器	双方互相验证
客户端证书	不需要	必须配置
安全性	中等（防窃听、篡改）	更高（防冒充、中间人攻击）
复杂度	配置简单，仅需上传服务器证书	需上传服务器证书与客户端证书
典型应用	普通网站	银行系统、内部 API

证书类型

证书类型分为默认服务器证书、自定义服务器以及 CA 证书，仅在认证模式选择双向认证时需要上传和管理 CA 证书。

证书类型	说明
默认服务器证书	监听器创建时上传的服务器证书即为默认证书，当客户端请求未匹配上其他任何自定义服务器证书时，全球加速将返回默认证书用于 HTTPS 认证。 • 默认证书仅支持替换，不支持删除或添加。 • 一个 HTTPS 监听器有且仅有一本默认证书。
自定义服务器证书	当需要通过一个全球加速实例加速多个 HTTPS 域名时，可为监听器添加多个自定义证书，每个证书对应不同域名。 **说明：** 自定义证书支持替换，所替换的新证书域名需与老证书域名保持一致，否则无法完成替换。
CA 证书	认证模式选择双向认证时，除服务器证书外，还需要上传 CA 证书，用于验证客户端身份的合法性。 • CA 证书支持替换，不支持删除或添加。 • 一个 HTTPS 监听器有且仅有一本 CA 证书。

关联域名

全球加速支持单个 HTTPS 监听器下添加多个域名证书，实现多域名加速时的灵活管理。在添加自定义服务器证书时，您需要创建证书与域名的关联关系，关联后，全球加速将根据客户端请求域名返回对应证书，如没有匹配到任何自定义证书包含的域名，将返回默认证书。

上传证书

前提条件

已完成全球加速实例及 HTTPS 监听器创建。

操作步骤

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**证书管理**，进入证书管理页签。
5. 单击添加证书，在弹窗中完成添加配置。

配置项	说明
证书类型	所添加的证书类型，仅支持添加自定义服务器证书。
证书	选择所需添加的证书，您可在 SSL 控制台 对证书进行统一管理。
关联域名	服务器证书包含的域名，全球加速根据客户端请求域名返回对应证书。

替换证书

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**证书管理**，进入证书管理页签。
5. 单击已有证书右侧的**替换**，进行证书替换。

> **说明：**
> 自定义服务器证书替换时，所替换的新证书域名需与老证书域名保持一致，否则无法完成替换。默认证书及 CA 证书替换时，无需域名一致。

删除证书

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**监听器 ID**，进入监听器详情页。
4. 单击**证书管理**，进入证书管理页签。
5. 单击已有自定义服务器证书右侧的**删除**。
6. 在弹窗中，单击**确定**，完成删除。

相关文档

- 配置HTTP和HTTPS监听器

访问控制

最近更新时间：2025-06-25 20:03:21

概述

全球加速支持通过设置安全访问策略，对加速实例进行外网访问权限控制，提高网络访问的安全性。可通过来源 IP、协议、端口对流量进行访问限制。

- 全球加速默认不对客户端流量进行防控，需在访问控制页签开启该功能。
- 功能开启需选择默认处理策略，准许或拒绝全部流量进入全球加速实例，并通过访问规则进一步控制。

创建访问控制

前提条件

已完成全球加速实例创建。

创建访问控制策略

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**访问控制**，进入访问控制配置页。
4. 单击**创建访问控制**，选择默认处理策略，完成访问控制策略的创建。
5. 单击**状态**开关，启用控制策略。

创建访问规则

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**访问控制**，进入访问控制配置页。
4. 单击**添加规则**，在弹窗中对访问规则进行配置。

配置项	说明
来源 IP	客户端流量源 IP，来源支持以下格式 • 单个 IP: 192.168.0.1 • CIDR: 192.168.1.0/24
协议	客户端来源协议，支持 TCP、UDP。
协议端口	来源协议端口，支持以下格式 • 单个端口: 80 • 多个端口: 80,443 • 连续端口: 3306-20000 • 所有端口: ALL
策略	允许：全球加速放通命中规则的流量。 拒绝：全球加速将拒绝命中规则的流量访问。
备注	规则备注，非必填。

5. 单击**确定**，完成规则配置。

编辑规则

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**访问控制**，进入访问控制配置页。
4. 在已有规则右侧，单击**编辑**。

5. 在弹窗中完成对应配置，单击**确定**，完成编辑。

删除规则

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**访问控制**，进入访问控制配置页。
4. 选中需要删除的规则，单击**删除规则**。
5. 在弹窗中单击**确定**，完成删除。

删除访问控制策略

1. 登录 全球加速控制台 。
2. 在实例列表页面，单击目标**实例 ID**，进入实例详情页。
3. 单击监听器页签下的**访问控制**，进入访问控制配置页。
4. 单击访问安全控制右侧的**删除**，并点击弹窗中的**确定**，完成删除。

> ⚠ **注意：**
> 策略删除后，全部访问规则也会对应删除，全球加速将不再对业务进行访问控制，请充分确认影响后再操作。

TLS安全策略组

最近更新时间：2025-05-16 17:48:13

概述

TLS（Transport Layer Security）是一种用于保障网络通信安全的加密协议，其前身是 SSL（Secure Sockets Layer）。TLS 通过加密、身份验证和数据完整性保护，确保客户端（如浏览器）与服务器之间的数据传输不被窃听或篡改。它广泛应用于 HTTPS、电子邮件、VPN 等场景，是互联网保密通信的工业标准。TLS 协议经历了多个版本的迭代，每个版本在安全性和性能上有所改进：

- TLS 1.0（1999年）：首个版本，基于 SSL 3.0，但存在安全漏洞（如易受 BEAST 攻击），已逐渐被弃用。
- TLS 1.1（2006年）：修复了 TLS 1.0的部分漏洞，但仍使用较弱的加密算法（如 SHA-1），目前也不推荐使用。
- TLS 1.2（2008年）：主流版本，支持更强的加密算法（如 AES-GCM、SHA-256），提供更好的安全性和效率。
- TLS 1.3（2018年）：最新版本，简化握手流程（减少延迟）、移除不安全算法（如 RC4），并强制使用前向保密（PFS），安全性最高。

密码套件是 TLS 握手时协商的一组算法组合，用于定义加密、身份验证和密钥交换方式。创建 HTTPS 监听器时支持按需选择 TLS 安全策略组，不同安全策略组对 TLS 版本、加密套件包的支持度不同。详情如下：

TLS 安全策略组	支持 TLS 版本	支持加密算法套件
tls_policy_1.0-2	TLSv1.0、TLSv1.1和TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES256-SHA384 AES128-GCM-SHA256 AES256-GCM-SHA384 AES128-SHA256 AES256-SHA256 ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA AES128-SHA AES256-SHA DES-CBC3-SHA
tls_policy_1.1-2	TLSv1.1和TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES256-SHA384 AES128-GCM-SHA256 AES256-GCM-SHA384 AES128-SHA256 AES256-SHA256

		ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA AES128-SHA AES256-SHA DES-CBC3-SHA
tls_policy_1.2	TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES256-SHA384 AES128-GCM-SHA256 AES256-GCM-SHA384 AES128-SHA256 AES256-SHA256 ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA AES128-SHA AES256-SHA DES-CBC3-SHA
tls_policy_1.2_strict	TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA
tls_policy_1.2_strict-1.3	TLSv1.2及 TLSv1.3	TLS_AES_128_GCM_SHA256 TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_CCM_SHA256 TLS_AES_128_CCM_8_SHA256 ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-ECDSA-AES128-SHA256

		ECDHE-ECDSA-AES256-SHA384 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES256-SHA384 ECDHE-ECDSA-AES128-SHA ECDHE-ECDSA-AES256-SHA ECDHE-RSA-AES128-SHA ECDHE-RSA-AES256-SHA

相关文档

配置 HTTP 和 HTTPS 监听器。