

云原生智能网关

云原生网关



腾讯云

【 版权声明 】

©2013–2026 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 】



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

云原生网关

云原生网关概述

快速入门

使用云原生网关访问 HTTP 服务

使用云原生网关访问 TKE 服务

使用云原生网关访问 Polaris

使用云原生网关实现 WebSocket 服务转发

操作指南

网关管理

新建网关

查看网关详情

升级网关版本

升级网关产品版本

升级网关规格

集群分组管理

网关节点管理

删除网关

服务路由

服务

路由

服务来源

灰度发布

限流策略

健康检查

全局配置

证书管理

域名管理

Ingress

Kong Ingress介绍

使用 Kong Ingress Controller

安全防护

使用 Web 安全防火墙

Web 安全防护概述

域名防护

对象防护

使用 DDos 防护

结合 WAF 和 DDos实现安全防护

监控告警

查看默认监控

日志

查看默认日志

日志投递至 CLS

上报日志到 Kafka

插件上报日志到 CLS

查看日志大盘

Terraform 管理

参数配置

插件管理

弹性伸缩

事件中心

故障演练

迁移指南

与自建网关对比

从自建 Kong 网关迁移到腾讯云

从 Spring Cloud Gateway 迁移到云原生网关

实践教程

认证鉴权

开启 admin-api 并配置安全认证

使用 IP 访问控制

使用 Key Auth 密钥认证

使用 HMAC Auth 认证访问

使用 ACL 访问控制

使用 Bot Detection 插件拒绝爬虫请求

配置 JWT 认证鉴权

配置自定义鉴权

流量控制

使用缓存加速访问

使用熔断保护后端

使用流量镜像插件

请求路由

进行 URL 重写

使用重写插件改写请求或响应

云原生网关

云原生网关概述

最近更新时间：2026-07-23 17:07:31

云原生网关是腾讯云基于开源微服务网关推出的一款**高性能高可用**的网关产品，减少用户自建网关的开发及运维成本。云原生网关作为云上微服务架构的流量入口，集成请求分发、API 管理、流量监控、访问限制等功能，是微服务架构中的重要组件。

微服务平台 TSF 提供基于开源 Kong 的云原生网关服务，结合灰度分流、流量限制、安全控制等场景，提供基于微服务体系和云原生体系的可插拔的网关功能。

产品特点

- **兼容开源网关**：开源网关内核，兼容多个版本，开箱即用，降低自建复杂度。
- **高性能高可靠**：提供多种规格的实例，支持无缝扩缩容，轻松应对流量洪峰。
- **高性价比**：采用腾讯云底层资源，由腾讯云托管网关，无自建网关的人力成本及运维成本。
- **腾讯云特色功能**：提供与腾讯云环境、腾讯云产品的标准化对接方式，支持腾讯云开发的特色网关插件。

业务场景



功能特性

功能点	说明
数据可视化	在腾讯云控制台可查看云原生网关的集群信息，可对集群配置、系统参数进行修改。
批量修改系统参数	支持在腾讯云控制台批量调整云原生网关的系统参数，如请求体大小、TLS 版本等，不停服、安全、可靠。

日志监控	支持 CPU 利用率、内存使用率、入带宽流量、出带宽流量等底层资源的监控指标，支持通过插件将监控数据输出到 Prometheus 监控等开源平台或腾讯云监控平台并与腾讯云日志系统 CLS 高度集成，可直接在云上分析日志。
不停服扩缩容	集成腾讯云 CLB 的探活、负载均衡能力，进行自动节点故障流量切换，支持灵活升降配、支持按需扩展，过程对客户透明、支持资源规格扩缩容、支持副本数扩缩容。
IP 黑白名单	云原生网关中支持创建 IP 黑白名单，名单中可配置 IP 或网段，对访问进行限制。
容灾备份	提供高安全可靠性的网关托管，可备份网关数据和多可用区。
免运维	由腾讯云托管网关，无自建网关的人力成本及运维成本。
轻松迁移	使用迁移插件和官方工具，无门槛平滑迁移，开箱即用。

快速入门

使用云原生网关访问 HTTP 服务

最近更新时间：2026-07-23 17:07:32

本文将快速引导您如何在云原生网关中，访问后端服务为 HTTP 类型的 API。

概述

您需要依次完成以下步骤：

1. 创建云原生网关
2. 定义后端服务信息
3. 为服务配置路由规则
4. 调用 API

操作步骤

步骤1：创建云原生网关实例

进入 [云原生智能网关控制台](#)，参见 [新建网关](#)，创建一个云原生智能网关。

步骤2：定义后端服务信息

1. 单击创建好的网关实例的“ID”，进入实例详情页。
2. 在左侧导航栏选择**服务路由** > **服务**，在服务列表单击**新建**。
3. 在新建服务窗口中配置相关参数后，单击**确定**。本步骤是指定在云原生网关收到 Client 端的请求后，转发到哪个后端服务。本例中的后端服务可以为腾讯云上或在互联网上的任何可以访问到的地址。

新建服务

×

服务名称

test-service

服务类型

K8S 服务

注册中心

IP 列表

域名/IP

云函数

服务地址

service- encent.com

支持填写 IP 或域名，例如：10.0.0.1或www.tencent.com

服务端口

8000

后端服务的端口，支持1~65535的任意端口

请求协议

☒ HTTP
 ☐ HTTPS

请求路径

/test

超时时间

—

5000

+

毫秒

重试次数

—

5

+

提交

关闭

步骤3：为后端服务配置路由

1. 后端服务添加完成后，点击后端服务名称进入服务路由页面。
2. 单击**新建**，设置好路由信息，单击下一步。请求方法、请求路径、Host 需至少配置一种。

新建路由

×

1 基本信息配置

>

2 路由后端配置

路由名称

测试路由

路由名称允许为空，支持中文，英文大小写，数字，-_~

请求协议

HTTP&HTTPS

HTTPS

HTTP

匹配规则 ⓘ

请求方法

GET POST PUT

请求路径

/test

×

添加路径

Host

请输入 Host

×

添加 Host

▲ 高级配置

Strip Path ☐ 开启

保留 Host ☐ 开启

下一步

关闭

- 配置路由后端信息，确认后端服务信息无误后单击**确定**，完成路由创建。路由优先级检测（可选）：当同一服务下存在多条路由时，可通过路由优先级检测确认请求的匹配顺序，避免路由冲突。

新建路由

×

✓ 基本信息配置

>

2 路由后端配置

目标后端

☒ 单服务 ☐ 多服务

后端服务

服务名称

test-service

服务类型

域名/IP

请求路径

/test

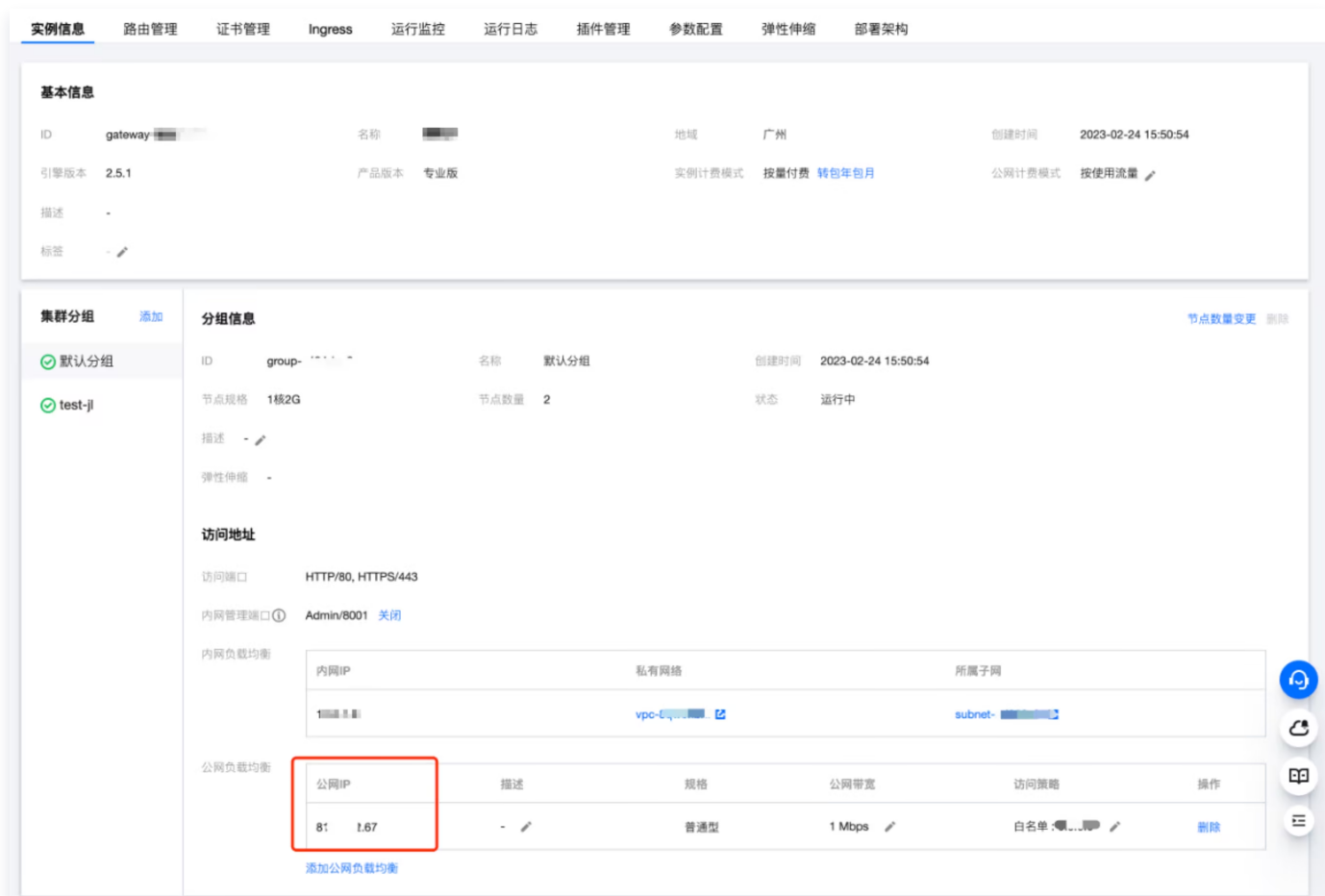
上一步

提交

关闭

步骤4：调用 API

1. 进入云原生网关的基本信息页面，获取负载均衡地址。



实例信息 路由管理 证书管理 Ingress 运行监控 运行日志 插件管理 参数配置 弹性伸缩 部署架构

基本信息

ID	gateway-...	名称	...	地域	广州	创建时间	2023-02-24 15:50:54
引擎版本	2.5.1	产品版本	专业版	实例计费模式	按量付费 转包年包月	公网计费模式	按使用流量
描述	-						
标签	-						

集群分组 添加

分组信息 节点数量变更 删除

ID	group-...	名称	默认分组	创建时间	2023-02-24 15:50:54
节点规格	1核2G	节点数量	2	状态	运行中
描述	-				
弹性伸缩	-				

访问地址

访问端口 HTTP/80, HTTPS/443

内网管理端口 ① Admin/8001 关闭

内网负载均衡

内网IP	私有网络	所属子网
10.10.10.1	vpc-...	subnet-...

公网负载均衡

公网IP	描述	规格	公网带宽	访问策略	操作
81.1.67	-	普通型	1 Mbps	白名单...	删除

添加公网负载均衡

2. 使用负载均衡地址和配置的路由路径访问 API，执行 curl 命令测试，即可通过云原生智能网关访问到后端服务。

```
curl http://<网关负载均衡 IP>/<路由路径>
```



```
-MB0 ~ % curl -i http://1.1.1.1:233/test
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Date: Thu, 17 Nov 2022 10:21:06 GMT
Server: apigw/1.0.15
Vary: Accept-Encoding
Via: kong/2.5.1
X-API-Id: api-1nrn0kuc
X-API-Requestid: 7d2e7cca76b1665143af4...
X-Kong-Proxy-Latency: 147
X-Kong-Upstream-Latency: 8
Content-Length: 11

hello Kong
```

使用云原生网关访问 TKE 服务

最近更新时间：2026-07-23 17:07:32

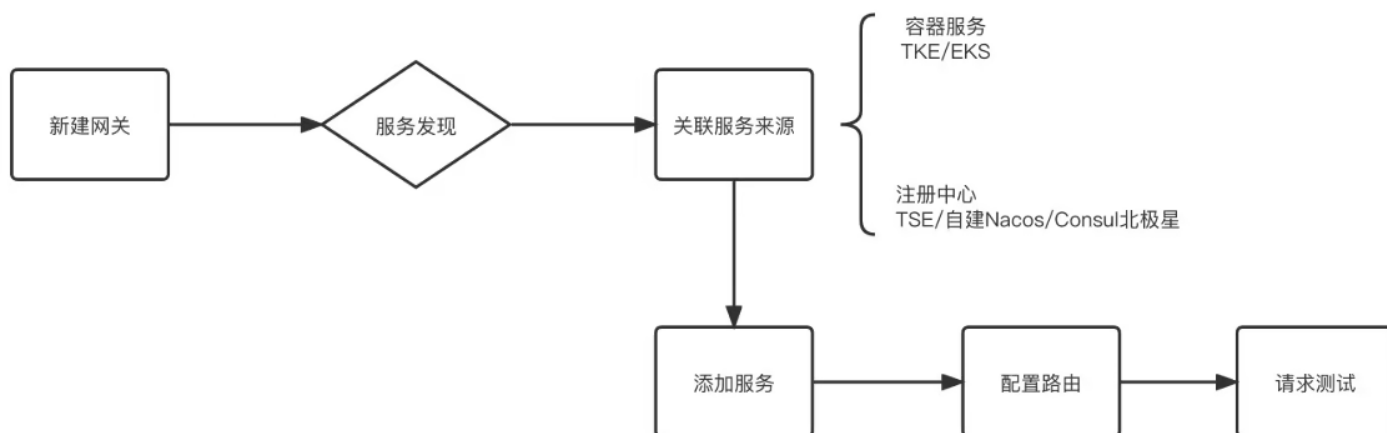
概述

您可以为容器服务创建一个云原生网关，从注册中心或容器服务 TKE 关联服务，然后在网关中为服务创建路由策略，以便该服务通过网关对外提供服务。本指导以容器服务为例，您需要完成以下步骤：

1. 创建云原生网关
2. 关联服务来源
3. 添加服务
4. 为服务配置路由规则
5. 调用 API。目前云原生网关仅对微服务场景，即作为 Polaris（北极星）服务接入层支持开通白名单新购，其余场景不支持新接入。非上述场景访问控制台时将重定向至 Polaris（北极星）控制台。

体验流程

微服务如果部署到容器服务 TKE，云原生网关可直接从 TKE 关联服务。



操作步骤

步骤1：创建云原生网关

参见 [新建网关](#)，创建一个云原生智能网关。如您已创建云原生智能网关，可直接前往 [腾讯微服务平台控制台](#) 进行后续操作。

步骤2：关联服务来源

1. 单击创建好的云原生网关的 ID，进入基本信息页面。
2. 在左侧菜单栏选择 **服务路由 > 服务来源**，在服务来源单击 **新建**，服务来源类型选择 **容器服务**，服务实例选择与云原生网关 在同一 VPC 下的可用实例，单击 **确定**。

新建服务来源

来源类型 *

注册中心

容器服务

私有 DNS

来源产品

TKE 标准集群

直接对接部署在 TKE 标准集群 上的服务

TKE Serverless集群

直接对接注册在 TKE Serverless集群 上的服务

选择实例 *

请选择

选购实例

请选择和云原生API网关有相同VPC的可用实例

确定

取消

步骤3：添加服务

1. 选择服务路由 > 服务，在服务列表单击**新建**，服务类型选择 K8S 服务，服务来源选择上一步关联的服务来源。填写容器服务的命名空间名称和服务名称。

新建服务

服务名称

请输入服务名称

服务类型

K8S 服务

注册中心

IP 列表

域名/IP

私有域名

云函数

服务来源

请选择服务来源

新建服务来源

命名空间

请选择命名空间

服务列表

请选择服务

请求协议

☒ HTTP

☐ HTTPS

☐ TCP

☐ UDP

请求路径

请输入请求路径

超时时间

-

60000

+

毫秒

重试次数

-

5

+

高级配置

负载均衡算法

轮询

慢启动

☐ 开启

开启后，服务对应的节点将在慢启动时间内，将权重从1逐步增加到目标值

确定

取消

2. 单击**确认**，创建完成后可以看到服务列表中服务的信息。

路由

服务

服务来源

全局配置

新建

请选择条件进行过滤

服务名称	服务类型	服务来源	创建时间	操作
	域名/IP	-	2023-09-21 17:29:39	编辑 删除
	域名/IP	-	2023-09-20 19:46:53	编辑 删除
	私有域名	-	2023-09-25 15:30:09	编辑 删除

步骤4：为服务配置路由规则

1. 后端服务添加完成后，单击服务名称进入服务路由页面。
2. 单击**新建**，设置好路由信息，单击下一步。请求方法、请求路径、Host 需至少配置一种。

新建路由

×

1 基本信息配置

>

2 路由后端配置

路由名称

测试nginx路由

路由名称允许为空，支持中文，英文大小写，数字，-，~

请求协议

HTTP&HTTPS

HTTPS

HTTP

匹配规则

请求方法

GET POST

请求路径

/myservice

×

添加路径

Host

请输入 Host

×

添加 Host

高级配置

Strip Path

☒ 开启

保留 Host

☐ 开启

下一步

关闭

3. 配置路由后端信息，确认后端服务信息无误后单击**确定**，完成路由创建。路由优先级检测（可选）：当同一服务下存在多条路由时，可通过路由优先级检测确认请求的匹配顺序，避免路由冲突。

新建路由

✓ 基本信息配置

>

2 路由后端配置

目标后端 ☒ 单服务

后端服务

服务名称

httpbin

服务类型

域名/IP

请求路径

/get

上一步

确定

取消

步骤5：调用 API

1. 进入云原生网关的基本信息页面，获取负载均衡地址。

实例信息 部署架构 参数配置 弹性伸缩 Konga 控制台

基本信息

[转包年包月](#)

ID		名称		地域	广州	创建时间	2023-09-15 11:53:33
引擎版本	2.5.1	产品版本	专业版	实例计费模式	按量付费	公网计费模式	按使用流量 ✎
描述	勿删						
标签	✎						

集群分组

[添加](#)

☒ 默认分组

分组信息

[删除](#)

ID		名称	默认分组	创建时间	2023-09-15 11:53:33
私有网络	chlc	子网	chlo	状态	运行中
节点规格	1核2G 变更	节点数量	2 变更	弹性伸缩	-
描述	✎				

访问地址

访问端口 [HTTP: 80](#) [HTTPS: 443](#) [TCP: 5181,8000-8009](#) [UDP: 7000-7009](#) [查看当前使用端口](#) [修改](#)

内网管理端口 [①](#) - [开启](#)

内网私有连接 [①](#)

1	4						编辑 删除
ID	ne	规格	普通型	带宽	共享	描述	-
	3r						

[添加](#)

公网负载均衡

1	6						编辑 删除
ID	ne	规格	普通型	带宽	1 Mbps	描述	clareclb
	2						
可用区类型	单可用区	可用区	广州六区	访问策略	白名单: 0.0.0.0		✎

[添加](#)

2. 使用负载均衡地址和配置的路由路径访问 API，执行 curl 命令测试，即可通过云原生智能网关访问到容器服务。

```
curl http://<网关负载均衡 IP>/<路由路径>
```

```
~ % curl -i http://10.10.10.10/myservice
HTTP/1.1 200 OK
Accept-Ranges: bytes
Connection: keep-alive
Content-Length: 615
Content-Type: text/html; charset=UTF-8
Date: Thu, 17 Nov 2022 11:24:46 GMT
Etag: "634fada5-267"
Last-Modified: Wed, 19 Oct 2022 07:56:21 GMT
Server: nginx/1.23.2
Via: kong/2.5.1
X-Kong-Proxy-Latency: 6
X-Kong-Upstream-Latency: 5

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

3/3

使用云原生网关访问 Polaris

最近更新时间：2026-07-23 17:07:32

概述

您可以为北极星实例创建一个云原生网关，从注册中心关联服务，然后在网关中为服务创建路由策略，以便该服务通过网关对外提供服务。本指导以 Polaris 为例，您需要完成以下步骤：

1. 创建云原生网关
2. 关联服务来源
3. 添加服务
4. 为服务配置路由规则
5. 调用 API

操作步骤

步骤1：创建云原生网关

参见 [新建网关](#)，创建一个云原生智能网关。如您已创建云原生智能网关，可直接前往 [腾讯微服务平台控制台](#) 进行后续操作。

步骤2：关联服务来源

1. 单击创建好的云原生网关的 ID，进入基本信息页面。
2. 在左侧菜单栏选择 **服务路由** > **服务来源**，在服务来源单击**新建**，服务来源类型选择**注册中心**，来源产品选择 **Polaris**，服务实例选择与云原生网关在同一 VPC 下的可用实例，单击**确定**。

新建服务来源

来源类型

注册中心容器服务私有 DNS

来源产品

TSE Nacos 推荐

直接对接注册在TSE Nacos 2.x上的服务

TSE Consul 推荐

直接对接注册在TSE Consul上的服务

TSE PolarisMesh 推荐

直接对接注册在TSE PolarisMesh上的服务

自建 Nacos

对接注册在Nacos上的服务

自建 Consul

对接注册在Consul上的服务

自建 PolarisMesh

对接注册在PolarisMesh上的服务

TSF 注册中心 推荐

对接注册在腾讯云微服务平台 TSF 上的服务

选择实例

选购实例

请选择和云原生API网关有相同VPC的可用实例

访问Token

请输入访问Token

如您开启了鉴权功能，请填写访问Token

确定

取消

步骤3：添加服务

1. 选择服务路由 > 服务，在服务列表单击新建，服务类型选择注册中心，服务来源选择上一步关联的服务来源。填写 Polaris 服务的命名空间 ID 和服务名称。服务列表为必填项，请确保从关联的服务来源中选择正确的服务。

新建服务

×

服务名称

pc

服务类型

K8S 服务

注册中心

IP 列表

域名/IP

私有域名

云函数

服务来源

新建服务来源

命名空间

Polaris

Nacos默认命名空间请填写empty_ns

服务列表

polaris.checker

服务协议

HTTP

HTTPS

TCP

UDP

服务路径

请输入服务路径

超时时间

—

60000

+

毫秒

重试次数

—

5

+

高级配置

负载均衡算法

轮询

慢启动

开启

开启后，服务对应的节点将在慢启动时间内，将权重从1逐步增加到目标值

确定

取消

2. 单击**确认**，创建完成后可以看到服务列表中服务的信息。

路由管理

中国大陆

扫码关注公众号

操作指南

路由

服务

服务来源

全局配置

新建

请选择条件进行过滤

Q

刷新

下载

ID/名称	健康检查	服务类型	服务来源	服务路径	创建时间	操作
69	不支持健康检查	私有域名	PrivateDNS 腾讯云私有解析	/get	2023-09-26 16:29:04	编辑 删除

步骤4：为服务配置路由规则

1. 后端服务添加完成后，单击服务名称进入服务路由页面。
2. 单击**新建**，设置好路由信息，单击**下一步**。请求方法、请求路径、Host 需至少配置一种。

新建路由

1 基本信息配置 > 2 路由后端配置

路由名称

路由名称允许为空，支持中文，英文大小写，数字，-_~

请求协议

HTTP&HTTPS

HTTPS

HTTP

匹配规则 ⓘ

请求方法

请选择



请求路径

/get



添加路径

Host

请输入 Host



添加 Host

Header

Header 键

Header 值



支持数字，大小写字母，_-

新增 Header

高级配置

Strip Path ☒ 开启

保留 Host ☐ 开启

- 配置路由后端信息，确认后端服务信息无误后单击**确定**，完成路由创建。路由优先级检测（可选）：当同一服务下存在多条路由时，可通过路由优先级检测确认请求的匹配顺序，避免路由冲突。

新建路由

1 基本信息配置

>

2 路由后端配置

目标后端

☒ 单服务

后端服务

服务名称

test-jl

服务类型

私有域名

服务来源

PrivateDNS

服务路径

/get

上一步

确定

取消

步骤5：调用 API

1. 进入云原生网关的基本信息页面，获取负载均衡地址。

基本信息

ID

gatew

名称

地域

广州

创建时间

2023-09-26 16:14:12

引擎版本

2.5.1

产品版本

专业版

实例计费模式

包年包月 2024-04-26 16:14:12到期

公网计费模式

按使用流量

描述

-

标签

-

集群分组

添加

默认分组

运行中

ID/名称

grou

组)

节点规格

1核2G 变更

节点可用区类型

随机可用区 查看节点部署

弹性伸缩

-

创建时间

2023-09-26 16:14:12

节点数量

2 变更

网络配置

私有网络

pc-pncg3lip

子网

onet-399lu1we

访问端口

HTTP: 80 HTTPS: 443 TCP: 8000-8009 UDP: 7000-7009 查看当前使用端口 修改

内网管理端口

- 开启

内网私有连接

ID

network

规格

普通型

带宽

共享

描述

-

公网负载均衡

添加

2. 使用负载均衡地址和配置的路由路径访问 API，执行 curl 命令测试，即可通过云原生智能网关访问到后端服务。

```
curl http://<网关负载均衡 IP>/<路由路径>
```

使用云原生网关实现 WebSocket 服务转发

最近更新时间：2026-07-23 17:07:32

WebSocket 的服务可以按照正常的 HTTP（HTTPS）服务配置，只要访问网关的时候使用 WebSocket 的协议，网关会自动升级协议为 WebSocket，用户不需要明确指定协议为 WebSocket。本文将快速引导您如何在云原生网关中，访问后端服务为 WebSocket 类型的 API。

操作步骤

步骤1：创建云原生网关实例

参见 [新建网关](#)，创建一个云原生智能网关。如您已创建云原生智能网关，可直接前往 [腾讯微服务平台控制台](#) 进行后续操作。

步骤2：定义后端服务信息

1. 单击创建好的网关实例的“ID”，进入实例详情页。
2. 在左侧导航栏选择 **服务路由 > 服务**，在服务列表单击 **新建**。
3. 在新建服务窗口中配置相关参数后，单击 **确定**。

❗ 说明：

新建网关服务，如果后端服务 WebSocket 提供的是 WS 协议，选择服务协议为 HTTP，如果后端服务提供的是 WSS 协议，选择服务协议为 HTTPS。服务的类型也可以为 K8S 服务/ IP 列表等。

新建服务

服务名称

websocket

服务类型

K8S 服务

注册中心

IP 列表

域名/IP

私有域名

云函数

服务地址

支持填写 IP 或域名，例如：10.0.0.1或www.tencent.com

服务端口

8765

后端服务的端口，支持1~65535的任意端口

服务协议 ⓘ



HTTP



HTTPS



TCP



UDP

服务路径

/

超时时间

-

60000

+

毫秒

重试次数

-

5

+

确定

取消

步骤3：为后端服务配置路由

1. 后端服务添加完成后，单击后端服务名称进入服务路由页面。
2. 单击**新建**，设置路由信息。
 - 请求方法：配置为 ANY。
 - 请求路径：配置为 /websocket。
3. 单击**下一步**。
4. 配置路由后端信息，确认后端服务信息无误后单击**确定**，完成路由创建。路由优先级检测（可选）：当同一服务下存在多条路由时，可通过路由优先级检测确认请求的匹配顺序，避免路由冲突。

新建路由



✓ 基本信息配置 > 2 路由后端配置

目标后端 ☒ 单服务

后端服务

服务名称

服务类型 IP 列表

服务路径

上一步

确定

取消

步骤4：访问 WebSocket 服务

1. 进入云原生网关的基本信息页面，获取负载均衡地址。

实例信息

路由管理

证书管理

Ingress

运行监控

运行日志

插件管理

参数配置

弹性伸缩

部署架构

基本信息

IDgateway-123456789名称123456789地域广州创建时间2023-02-24 15:50:54

引擎版本2.5.1产品版本专业版实例计费模式按量付费 转包年包月公网计费模式按使用流量

描述-

标签-

集群分组

添加

默认分组

test-jl

分组信息

节点数量变更

删除

IDgroup-123456789名称默认分组创建时间2023-02-24 15:50:54

节点规格1核2G节点数量2状态运行中

描述-

弹性伸缩-

访问地址

访问端口HTTP/80, HTTPS/443

内网管理端口Admin/8001 关闭

内网负载均衡

公网负载均衡

添加公网负载均衡

内网IP	私有网络	所属子网
10.0.1.8	vpc-123456789	subnet-123456789

公网IP	描述	规格	公网带宽	访问策略	操作
81.1.67	-	普通型	1 Mbps	白名单	删除

2. 使用负载均衡地址和配置的路由路径访问 API。

- 使用 WS 协议访问网关：`websocat ws://<网关负载均衡 IP 或域名>/websocket`。
- 使用 WSS 协议访问网关：`websocat wss://<网关负载均衡 IP 或域名>/websocket -k`。


```
→ ~ websocat ws://111[REDACTED].35/websocket
Hello
Echo: Hello
World
Echo: World
█
```

操作指南

网关管理

新建网关

最近更新时间：2026-07-23 17:07:34

操作场景

本文将介绍如何通过微服务平台控制台创建一个云原生智能网关实例。

操作步骤

1. 登录 [微服务平台控制台](#)，在左侧导航栏单击云原生网关 > 实例列表。
2. 单击**新建**，进入云原生智能网关创建页。

参数	是否必选	说明
计费模式	是	支持 包年包月 和 按量计费 两种模式，如果您的网关使用时间在一个月以上，建议采用 预付费（包年包月） 模式。具体价格请参考 价格说明 文档。
网关配置	是	普通场景选择 默认配置 即可，如果对网关版本有要求，可以选择 自定义配置 。
地域	是	选择与您部署业务最靠近的地域。
产品版本	是	支持开发版、标准版、专业版，专业版提供弹性伸缩等高级特性，可用于测试、生产环境。具体区别请参考 规格对比 。
网关版本	是	选择 自定义配置 时可指定网关版本，当前支持 2.8.1、2.5.4、2.5.2 等版本，建议选择最新稳定版本。
节点规格	是	云原生智能网关集群由多个节点组成，此处选择每个节点的规格。
节点数量	是	云原生智能网关集群由多个节点组成，此处选择节点的数量。 ● 节点数量为 2 时，仅支持随机分配可用区。 ● 节点数量大于等于 3 时，支持选择随机可用区和指定可用区（最少选择 2 个可用区，最多选择 3 个可用区）。 ● 节点可用区类型选择仅专业版支持指定，开发版和标准版无该选项。 ● 若开启公网负载均衡，公网负载均衡也提供可用区类型选项，可单独配置。

部署架构	是	- 开发版：同城单可用区。 - 标准版、专业版：仅支持同城双可用区，提供高可用版网关。
节点网络	是	选择当前云原生智能网关所部署的腾讯云 VPC 网络。
公网负载均衡	否	开启公网后，将产生费用，具体价格参考 价格说明 文档中的公网流量费用部分。 - 计费模式：支持按使用流量和按带宽计费两种。 - 带宽上限：支持1-2048Mbps 范围。 - 可用区类型：支持单可用区和多可用区。选择多可用区时，您需要选择一个主可用区和一个备可用区，当主可用区故障时，负载均衡可在较短时间内自动切换到备可用区并恢复服务，以保障网关的可用性。
日志采集	否	- 开启实时日志服务：默认为您提供网关实时日志服务和简单搜索能力，免费使用。如需日志能力持久化存储，用于排障、审计等场景，建议开启 CLS 日志服务。 - 开启 CLS 日志服务：为您开启网关日志投递，提供日志分析。在勾选本选项前，请先确认您已开通 CLS 日志服务。日志服务由 CLS 日志 服务提供，会产生费用，具体计费项请查看 CLS 计费说明。
名称	是	输入当前云原生智能网关的名称，最长60个字符，支持中英文大小写，-，_。
描述	否	输入当前云原生智能网关的描述信息。
资源标签	否	用于分类管理资源，具体使用方法可参见 标签管理 。

3. 单击**立即购买**，耐心等待实例创建完成。

结果验证

返回网关列表页面，查看创建的网关信息和状态。网关信息和创建时一致，且状态为运行中，则说明网关新建成功。

注意事项

- 创建云原生智能网关要求账号必须拥有 `ApiGateway_QCSRole` 角色，并且角色下必须有 `QcloudAccessForApiGatewayRoleInCloudNativeAPIGateway` 策略。如无相关角色和策略，请在创建过程中弹出的授权弹窗中完成授权。
- 云原生智能网关日志功能依赖于 [腾讯云日志服务](#)（简称 CLS），请您确保已经开通 CLS 服务，否则在创建云原生智能网关前请前往 [CLS 控制台](#) 开通服务。

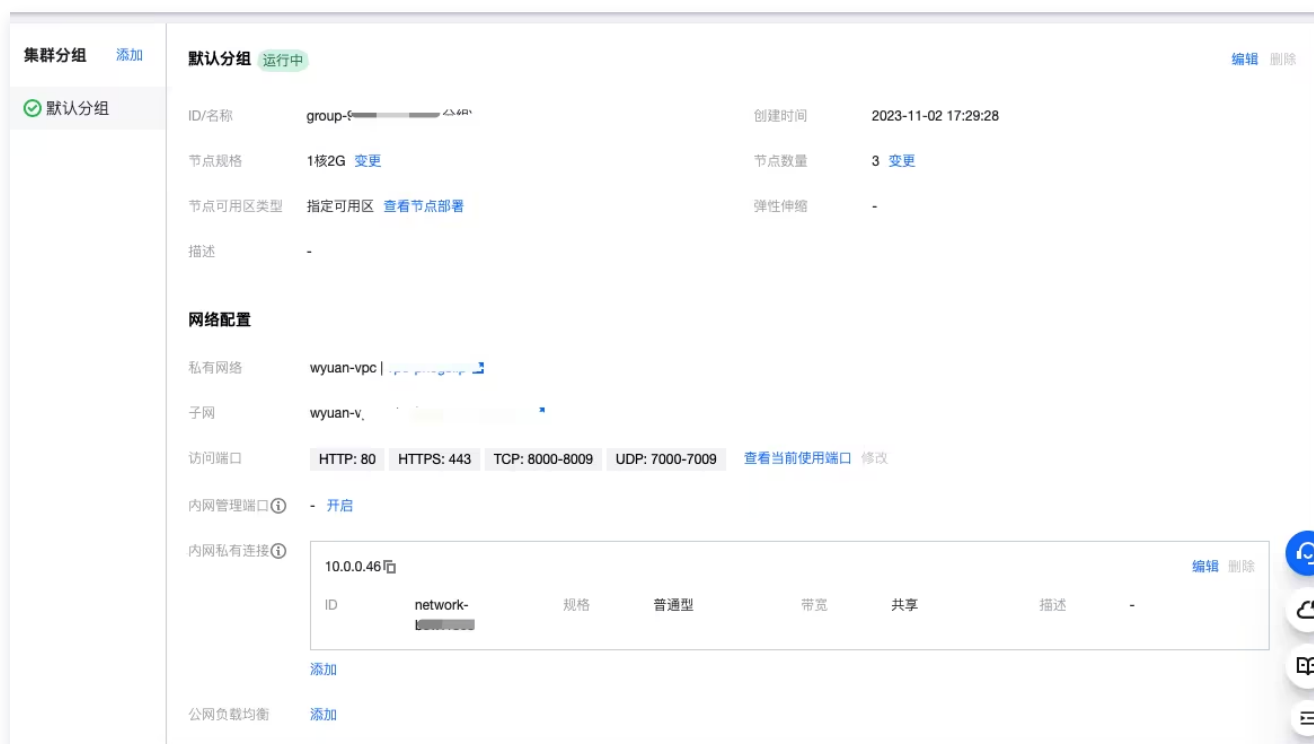
查看网关详情

最近更新时间：2026-07-23 17:07:34

本文介绍在创建一个网关实例后，如何查看网关的基本信息以及访问地址等。

操作步骤

1. 登录 [微服务平台控制台](#)，在左侧导航栏单击云原生网关 > 实例列表。
2. 在实例列表页面，单击刚刚创建好的实例的“ID”，进入网关实例基本信息页面，可查看实例的基本信息和集群分组信息。
 - **实例信息：**您可以在基本信息区域进行如下操作：
 - 单击基本信息模块右上角的编辑，可修改网关实例的名称和说明。
 - 单击公网计费模式的编辑，可以修改公网的计费方式。
 - 单击标签的编辑，可以修改网关绑定的标签信息。
 - **集群分组：**集群分组支持多个集群使用相同的路由配置，方便实现对多个云原生网关实例进行统一管理。每个网关实例创建后均会生成一个默认分组。



- **分组信息：**关于集群分组的详细介绍和操作请参考 [集群分组](#)。
- **网络配置：**
 - **访问端口：**云原生网关的访问端口。
 - **内网管理端口：**云原生网关的管理端口。管理端口没有权限控制，仅支持通过内网私有连接访问。

- 内网私有连接：云原生网关内网地址。客户端可以通过内网节点地址访问云原生网关的网关节点。
- 公网负载均衡：展示目前网关实例配置的公网负载均衡，支持修改公网带宽和配置访问策略，公网访问建议设置访问安全策略。

❗ 说明：

内网和公网负载均衡支持选择负载均衡规格，单击**添加负载均衡**后，设置好负载均衡规格，单击**确定**即可。

添加公网负载均衡

IP 版本

IPV4

IPV6

负载均衡描述

请输入负载均衡描述

最长60个字符，支持字母、数字、中文字符，'-'，'_'，'.'

负载均衡规格

普通型

性能容量型

型号	最大连接数 (个)	每秒新建连接数 (个)	带宽上限 (Mbps)
☒ 标准型规格	100000	10000	2048
☐ 高阶型1规格	200000	20000	4096
☐ 高阶型2规格	500000	50000	6144
☐ 超强型1规格	1000000	100000	10240
☐ 超强型2规格	2000000	200000	20480
☐ 超强型3规格	5000000	500000	40960
☐ 超强型4规格	10000000	1000000	61440

带宽上限

1Mbps

2048Mbps

-

1

+

Mbps

可用区类型

单可用区

多可用区

主可用区

请选择

备可用区

请选择

费用

公网流量费用

性能容量费用

限时免费

确定

取消

3. 在实例详情页面，单击**部署架构**页签，展示当前所选分组下网关节点的状态及可用区信息。

单击切换视图，切换到部署架构图，展示当前所选分组下接入层（CLB）和网关节点的部署情况。

实例信息部署架构参数配置弹性伸缩Konga 控制台

调整节点权重

当前分组默认权重 10

Default group

请输入关键字进行搜索

节点 ID	所属分组	内网地址	状态	可用区	节点权重	操作
group-cf54a449-kong-0	Default group	10.0.0.9	运行中	新加坡二区	10 (默认权重)	重启
group-cf54a449-kong-1	Default group	10.0.0.18	运行中	新加坡一区	10 (默认权重)	重启

共 2 条

10 / 页1 / 1 页

说明：
云原生网关标准版、专业版节点默认为多可用区部署。

升级网关版本

最近更新时间：2026-07-23 17:07:35

操作场景

如果当前的网关版本已进入下线阶段，为保证网关实例的稳定运行，建议您通过控制台将您的网关升级到稳定版本。

 **注意：**

为确保升级成功，请勿在升级前和升级过程中对网关实例执行变更类操作。升级过程中，网关控制台将禁止写入，请妥善安排您的升级计划。

版本升级

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关 > **实例列表**，找到目标网关。
3. 如果网关版本列显示可升级，在**操作** > **更多**，单击**升级网关版本**。
4. 单击**确定**，进入版本升级过程。
5. 升级前将执行升级环境检查，需确保所有检查项均通过后才能进行升级，如存在检查失败项，可修复后单击**重新检查**，重新执行环境检查。检查内容如下：

检查项	检查通过条件	检查失败处理措施
实例状态	状态处于运行中	等待实例变更完成，状态处于运行中后执行重新检查
服务后端状态	不存在后端指向网关 admin 地址的服务	删除服务或修改服务地址后执行重新检查
弹性伸缩状态	所有弹性伸缩规则未启用	关闭启用状态的弹性伸缩规则后执行重新检查
子网 IP 数量	子网 IP 数量足够	前往 VPC，释放子网 IP 后执行重新检查
节点权重	节点权重一致	前往 实例详情 > 基础信息 > 部署架构 ，手动调整节点权重，保持节点权重一致后执行重新检查

6. 确认实例 ID、实例名称、选择升级目标版本，阅读并勾选网关升级相关事宜说明，单击**确定**，开启升级。升级整体耗时与实例当前节点数相关，如节点数较多可能会耗时较长，您可以在网关列表的状态栏查看任务状态。

查看升级任务进度

1. 单击云原生网关 > **实例列表**，找到目标网关，单击**状态**，单击**查看任务状态**。

2. 升级采用灰度策略，分为两个阶段，**版本升级与资源回收**。版本升级主要用于配置升级所用的组件资源，该过程中节点可能增加。完成版本升级后，需要进行资源回收，销毁升级过程中多余的节点，因此需要您手动确认，同意执行资源回收，之后网关会自动触发旧资源回收流程。

任务进度查询

1. 升级采用灰度策略，升级过程中节点数可能增加，属于正常情况，升级完成后多余节点将自动销毁。

2. 升级整体耗时与实例当前节点数相关，如节点数较多可能会耗时较长，请耐心等待。

3. 当进入【引流到新网关实例】阶段时，请及时关注业务状况，如发现异常可撤销升级。

4. 关闭弹窗不影响升级过程，可在任务状态中重新打开查看。

实例IDga12

实例名称te

阶段一：版本升级

100 %

阶段	状态	原因	信息
创建网关初始配置	成功		
初始化监控配置	成功		
创建后端地址同步组件	成功		
创建网关实例	成功		
更新系统插件	成功		
引流到新网关实例	成功		

阶段二：资源回收

0 %

已完成网关版本升级，请观察业务运行日志与流量状态。若业务正常，点击【同意执行】触发旧资源回收

同意执行

阶段	状态	原因	信息
暂无数据			

撤销升级

关闭

撤销升级

如在升级过程中发现业务异常，可撤销升级。

1. 单击云原生网关 > 实例列表，找到目标网关，单击状态，单击查看任务状态。
2. 单击撤销升级。网关将启动撤销升级任务，将流量回切以保证生产的稳定性。

任务进度查询

1. 升级采用灰度策略，升级过程中节点数可能增加，属于正常情况，升级完成后多余节点将自动销毁。

2. 升级整体耗时与实例当前节点数相关，如节点数较多可能会耗时较长，请耐心等待。

3. 当进入【引流到新网关实例】阶段时，请及时关注业务状况，如发现异常可撤销升级。

4. 关闭弹窗不影响升级过程，可在任务状态中重新打开查看。

实例ID

gate

实例名称

tei

撤销升级

100 %

阶段	状态	原因	信息
回切流量	成功		
清理升级创建的资源	成功		
清理监控配置	成功		
回滚系统插件	成功		
更新konga副本数	成功		
回切流量后置处理	成功		

撤销升级

关闭

版权所有：腾讯云计算（北京）有限责任公司

第35 共261页

升级网关产品版本

最近更新时间：2026-07-23 17:07:35

操作场景

如果当前的网关产品版本所提供的功能不能满足您的业务需求，您可以通过控制台将您的网关升级到更高的产品版本，从而体验更丰富的功能。

操作步骤

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏，单击云原生网关 > 实例列表，找到目标网关，单击网关的“ID”，进入实例信息页面。
3. 在实例信息 > 基本信息卡片，找到产品版本，单击升级，选择好目标版本、节点规格和节点数量。



4. 单击确定，等待大约3 - 5分钟完成产品版本升级，您可以在网关列表的状态栏查看任务状态。

版本升级

×

✓ 升级前检查

>

2 确认升级

ⓘ

1. 升级采用灰度策略，升级过程中节点数可能增加，属于正常情况，升级完成后多余节点将自动销毁。

2. 升级整体耗时与实例当前节点数相关，如节点数较多可能会耗时较长，请耐心等待。

3. 升级过程对短连接无损，长连接会有连接重建的过程，如网关存在大量长连接，建议在业务低峰期进行升级操作。

4. 升级过程中将禁止写入操作，升级完成后自动恢复控制台操作。

5. 正则路径匹配优先级策略说明：当多个正则path同时匹配时，将按正则表达式字符串长度排序（原开源版本按创建时间先后排序）。若您的相关配置依赖创建先后顺序，可能影响升级任务，请提前验证。

6. 如有疑问提交工单进行咨询，我们将协助您平滑过渡。

实例 ID

gateway-f4e74d0a

实例名称

grand-aigw-3.9.4

当前网关版本

3.9.4

升级目标版本

3.9.5

☐

我已知晓网关版本升级相关事宜，并同意进行版本升级

上一步

升级

取消

升级网关规格

最近更新时间：2026-07-23 17:07:36

操作场景

如果当前的网关规格不能满足您的业务需求，可以通过控制台提升您的网关节点数量和节点规格。

说明

如需降低网关节点数量和节点规格，请 [提交工单](#) 处理。

前提条件

只有运行中/升级失败状态的实例支持升级规格。

节点数量变更

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏，单击云原生网关 > 实例列表，找到目标网关，点击网关的“ID”，进入基本信息页面。
3. 在分组信息模块，单击节点数量处的变更，进入节点变更页。
4. 在规格变更页，选择目标升级节点数，单击确定，在确认弹窗中选择确定，等待大约3 – 5分钟完成实例升级，您可以在网关列表的状态栏查看任务状态。

节点规格变更

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏，单击云原生网关 > 实例列表，找到目标网关，点击网关的“ID”，进入基本信息页面。
3. 在分组信息模块，点击节点规格处的变更，进入规格变更页。



4. 在规格变更页，选择目标升级节点规格，单击确定，在确认弹窗中选择确定，等待大约3 – 5分钟完成实例升级，您可以在网关列表的状态栏查看任务状态。

集群分组管理

最近更新时间：2026-07-23 17:07:36

操作场景

云原生网关集群分组支持多个分组共用相同的路由配置，方便实现对多个分组进行统一管理。

集群分组的适用场景如下：

1. 为相同业务分配不同规格的资源。不同分组支持配置不同的资源，包括节点数量和节点规格，您可以为开发用途的分组配置较低的节点规格和数量，为生产用途的分组配置较高的节点规格和数量，实现资源按需购买。
2. 为网关提供多个访问入口。不同分组支持使用不同的负载均衡 CLB，提供多个入口，您可以为不同的 CLB 配置不同的访问策略，并将其分配给不同的角色，实现权限控制。

❗ 说明：

每个网关实例创建后均会生成一个默认分组，您可以按需添加自定义分组。

前提条件

[已创建云原生网关实例](#)。

操作步骤

步骤1：添加分组

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏，单击[云原生网关](#) > [实例列表](#)，进入网关实例列表页面。
3. 单击目标实例的“ID/名称”，进入实例详情页面。
4. 在基本信息页面下方的 [集群分组](#) 卡片，展示 [默认分组](#)。每个网关实例创建后均会生成一个默认分组。
5. 单击[添加分组](#)，填写新添加的分组信息。

添加分组

×

计费模式

包年包月

节点规格

1核2G

节点数量

−

2

+

节点可用区类型

随机可用区

指定可用区

节点网络

wyuan-vpc(vpc-pncg3ilp)

wyuan-vpc-sub1(subnet-399iu)

共253个IP，剩240个可用

如果现有的网络不合适，你可以[新建集群网络](#) 或者 [新建子网](#) [刷新](#)

公网负载均衡

☐ 免费分配公网 IP 和负载均衡

名称

请输入分组名称

最长60个字符，支持字母、数字、中文字符，'-'，'_'

描述

请输入分组描述

最长120个字符

网关实例费用

元

确定

取消

参数	是否必填	说明
计费模式	—	默认与网关实例计费模式保持一致，不支持修改。
节点规格	是	集群分组每个节点的规格。
节点数量	是	集群分组节点的数量。
节点可用区类型	是	- 节点数量为2时，仅支持随机分配可用区。 - 节点数量大于等于3时，支持选择随机可用区和指定可用区（最少选择 2 个可用区，最多选择 3 个可用区）。（仅专业版支持指定可用区）
节点网络	是	选择集群分组所部署在的腾讯云 VPC 网络和腾讯云子网。VPC 网络默认与网关实例相同，支持选择 腾讯云子网。
所属子网	是	选择当前云原生网关所部署在的腾讯云子网，子网属于 VPC 网络。

公网带宽	是	选择集群分组的公网带宽。
分组名称	是	输入当前分组的名称。
描述	否	输入当前分组的描述信息。

6. 单击**确定**，耐心等待分组创建完成。

步骤2：修改分组基本信息

1. 在基本信息页面右上角，单击**编辑**。
2. 在修改分组名称/描述弹窗中修改分组名称/描述。
3. 单击**确定**，完成修改。

步骤3：分组规格变更

1. 在分组信息卡片，找到节点数量，单击右侧的**节点数变更**。
2. 在节点数变更弹窗中修改节点数量。
3. 单击**确定**，完成修改。

步骤4：删除分组

当您不再需要集群分组时，可以在控制台手动删除该分组。

1. 在分组信息卡片，单击右侧的**删除**。
2. 在删除分组弹窗中二次确认，即可删除分组。

❗ 说明：

- 只有运行中/失败状态的分组可以被删除。
- 分组删除后，该分组节点及其数据均会被删除，请谨慎操作。

步骤5：访问路由

当您访问路由时，需使用“分组的公网负载均衡/内网私有连接 IP + 路由路径”拼接访问 URL。

集群分组

添加

默认分组

默认分组

默认分组

运行中

编辑

删除

ID/名称

grfc (默认分组)

创建时间

2027:38

节点规格

1核2G 变更

节点数量

5 变更

节点可用区类型

随机可用区 查看节点部署

弹性伸缩

测试 | str59d

描述

-

网络配置

私有网络

H405

子网

te-8t zp8ugg

访问端口

HTTP: 80 HTTPS: 443 TCP: 8000-8009 UDP: 7000-7009 查看当前使用端口 修改

内网管理端口

- 开启

内网私有连接

1012

编辑

删除

ID

network-89ovq65e

规格

普通型

带宽

共享

描述

-

添加

公网负载均衡

19221

查看监控

编辑

删除

ID

network-cio5yoao

规格

普通型

带宽

1 Mbps

访问策略

白名单 : 0.0.0.0

可用区类型

单可用区

可用区

广州三区

描述

-

添加

网关节点管理

最近更新时间：2026-07-23 17:07:36

本文介绍在创建网关实例后，如何查看网关的节点信息以及进行网关节点管理。

说明：
此功能仅专业版支持。

操作步骤

步骤1：查看部署架构

1. 登录 [微服务平台控制台](#)，在左侧导航栏单击云原生网关 > 实例列表。
2. 在实例列表页面，单击网关实例的“ID”，进入网关实例基本信息页面，单击部署架构页签，可以查看网关实例的部署架构。
3. 部署架构图展示当前所选分组下接入层（CLB）和网关节点的部署情况。

说明：
云原生网关标准版、专业版节点默认为多可用区部署。

步骤2：查看节点信息

节点信息展示当前所选分组下网关节点的信息。

- 节点 ID：网关节点 ID。
- 所属分组：网关节点所属集群分组。
- 内网地址：网关节点内网地址。
- 状态：网关节点当前运行状态，包括运行中和修改中两种。
- 可用区：网关节点当前可用区。
- 节点权重：网关节点当前权重，权重越大承载的流量越多，初始状态为分组默认权重值，取值范围 0 – 100。
- 分组默认权重：网关分组内节点的默认权重，在该分组中增加节点后以该值作为初始权重值。

步骤3：修改节点权重

您可以通过控制台调整节点权重进行节点流量管理。

1. 在节点信息处，勾选需要调整权重的节点，单击调整节点权重，在弹窗中填写目标权重值。
2. 单击确定，完成节点权重修改。

调整权重

节点权重调整实时生效，不支持将所有节点权重同时调整为 0

以下网关节点权重修改为

范围: 0-100，流量不会转发到权重为 0 的网关节点，请谨慎操作

节点 ID	所属分组	内网地址	状态	节点权重
	默认分组		运行中	10 (默认权重)

确定

取消

说明：

1. 仅专业版支持调整节点权重。
2. 节点权重调整实时生效，流量不会转发到权重为 0 的节点，请谨慎操作。
3. 为保证网关可用性，不支持所有节点权重同时调整为 0。

步骤4：修改分组默认权重

分组默认权重决定新增节点和未进行自定义权重配置的节点权重值，您可以通过控制台调整分组默认权重进行节点的默认权重管理。

1. 在节点默认权重信息处，点击 **修改**，在弹窗中填写目标节点默认权重，默认为 10，取值范围 0-100。
2. 单击 **确定**，完成分组默认权重修改。

说明：

- 仅专业版支持调整节点权重。
- 分组默认权重调整实时生效。
- 已进行自定义权重配置的节点不受影响。

删除网关

最近更新时间：2026-07-23 17:07:34

当您不再需要云原生网关实例时，您可以删除该实例。

操作步骤

手动删除

1. 登录 [微服务平台控制台](#)，在左侧导航栏单击云原生网关 > 实例列表。
2. 在实例列表页面，找到目标实例，单击其操作列的**更多** > **删除**。
3. 在“确定删除”确认弹框中，请仔细阅读并勾选**确定删除**确认信息。如需同时释放实例关联的资源，可勾选**释放资源**选项（勾选后，该实例的数据及关联资源将被一并释放，不可恢复。可以选择立即释放或2小时后释放）。确认无误后单击**删除**，即可删除实例。

❗ 说明：

- 只有**运行中/失败状态**的实例可以被销毁。
- 实例销毁后，该实例及其数据均会被删除，该操作不可逆，请谨慎操作。

到期/欠费自动删除

您的网关实例在到期未续费/账户欠费后，最多在 TSE 控制台中保留7个自然日。到期后7天内完成续费可继续使用，详情请查看 [欠费说明](#)。

若您的实例在到期7天（包括第7天）后未进行续费，系统将在到期后第8天的0点开始资源释放，到期实例中的数据将被清除且不可恢复。

❗ 说明：

- 处于已隔离状态的实例不能生产和消费，已保存在实例内的数据和配置不会被删除。
- 对于7天内处于已隔离状态的实例，可以在控制台实例列表页，单击操作列的**更多** > **续费**。续费成功后，实例即可恢复运行状态，可正常使用。

服务路由

服务

最近更新时间：2026-07-23 17:07:35

操作场景

您需要将服务添加到云原生智能网关中，以便网关能查找对应的服务地址从而实现动态代理。云原生智能网关支持添加 K8S 服务、注册中心、IP 列表、域名/IP、私有域名、云函数几种类型的服务。本文介绍如何为云原生智能网关添加服务。

操作步骤

添加服务

- 1. 登录 [微服务平台控制台](#)。
- 2. 在左侧导航栏单击云原生网关进入实例列表，单击目标实例 ID/名称进入实例详情页。
- 3. 在左侧导航栏单击服务路由，单击服务页签，在服务列表单击新建。
- 4. 在新建服务窗口中配置相关参数。

参数	是否必填	说明
服务名称	是	输入服务名称。
服务类型	是	服务类型，支持 K8S 服务、注册中心、IP 列表、域名/IP、私有域名、云函数等多种类型。

 注意：

目前云原生网关仅对微服务场景，即作为 Polaris（北极星）服务接入层支持开通白名单新购，其余场景不支持新接入。非上述场景访问控制台时将重定向至 Polaris（北极星）控制台。

K8S 服务

当服务来源为 K8S 服务时，填写以下参数：

参数	是否必填	说明
服务来源	是	选择已创建的容器服务来源。

命名空间	是	输入命名空间名称。
服务列表	是	填写需要访问的容器服务列表，多个服务使用换行分隔。
服务协议	是	网关请求后端服务使用的协议，支持 HTTP/WS、HTTPS/WSS、TCP、UDP。
服务路径	否	后端服务的请求路径。
服务端口	否	后端服务的端口，支持1~65535的任意端口。
超时时间	是	连接建立、写入和读取后端返回数据的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

注册中心

当服务来源为注册中心时，填写以下参数：

参数	是否必填	说明
服务来源	是	选择已创建的服务来源。
命名空间	是	当来源产品为 Nacos/Polaris 时，输入命名空间 ID。注意：Nacos 默认命名空间请填写 empty_ns。当来源产品为 TSF 注册中心时，下拉选择命名空间。
服务列表	是	当来源产品为 Nacos/PolarisMesh/Consul 时填写需要访问的服务列表，多个服务使用换行分隔。针对不同注册中心，服务列表书写方式为： • Nacos: {Nacos 分组}_{服务名称} 注意：Nacos 分组与服务名称之间使用下划线 “_” 分隔。 • Consul: {服务名称} • Polaris Mesh: {服务名称} 当来源产品为 TSF 注册中心时，下拉选择单个微服务名称。
服务协议	是	网关请求后端服务使用的协议，支持 HTTP/WS、HTTPS/WSS、TCP、UDP。
服务路径	否	后端服务的请求路径。
超时时间	是	连接建立、写入和读取后端返回数据的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

IP 列表

当服务类型为 IP 列表时，需要填写以下参数：

参数	是否必填	说明
地址列表	是	填写后端服务节点的 IP、端口、权重信息。
CVM 弹性伸缩	否	开启后，Kong 将自动关联所选伸缩组对应的 CVM 实例。 - 伸缩组：勾选同 VPC 网络下的弹性伸缩组。关联后，Kong 将在弹性伸缩组创建生命周期挂钩，以及在云服务器-我的命令创建弹性伸缩命令，对上述资源的修改和删除将导致伸缩失效。 - 伸缩组端口：填写伸缩组端口。
服务协议	是	网关请求后端服务使用的协议，支持 HTTP/WS、HTTPS/WSS、TCP、UDP。
服务路径	否	后端服务的请求路径。
超时时间	是	连接建立、写入和读取后端返回数据的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

域名/IP

当服务类型为域名/IP 时，需要填写以下参数：

参数	是否必填	说明
服务地址	是	支持填写 IP 或域名，例如：10.0.0.1或 www.tencent.com。
服务端口	是	后端服务的端口，支持1~65535的任意端口。
服务协议	是	网关请求后端服务使用的协议，支持 HTTP/WS、HTTPS/WSS、TCP、UDP。
服务路径	否	后端服务的请求路径。
超时时间	是	连接建立、写入和读取后端返回的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

云函数

当服务类型为云函数时，需要填写以下参数：

参数	是否必填	说明
----	------	----

函数类型	是	事件函数或者 Web 函数。
命名空间	是	选择函数所在的命名空间。
函数	是	选择函数。
版本/别名	是	选择函数所在的服务版本或者别名。
CAM 鉴权	是	默认开启，网关转发请求到云函数进行 CAM 鉴权。
响应集成	否	是否启用响应集成。
Base64 编码	否	是否对响应进行 Base64 编码。
超时时间	是	连接建立、写入和读取后端返回的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

私有域名

当服务来源为 私有域名时，填写以下参数：

参数	是否必填	说明
服务来源	是	选择已创建的腾讯云私有解析来源。
私有域名	是	填写服务的私有域名。
服务端口	否	后端服务的端口，支持1~65535的任意端口。
服务协议	是	网关请求后端服务使用的协议，支持 HTTP/WS、HTTPS/WSS、TCP、UDP。
服务路径	否	后端服务的请求路径。
超时时间	是	连接建立、写入和读取后端返回的最大超时时间。
重试次数	是	与后端服务建立连接的最大重试次数。

1. （可选）配置高级参数。

- 负载均衡算法：支持选择轮询和最小连接数。
- 慢启动：开启后，服务对应的节点将在慢启动时间内，将权重从1逐步增加到目标值。若网关版本较低，可能不支持使用此功能，建议前往实例基本信息页面升级网关版本，详情请参见 [升级网关版本](#)。

❗ 说明:

支持在容器服务集群中修改 YAML 配置慢启动，增加 annotation:

```
annotations:  
  tse.konghq.com/TSE-Upstream-SlowStart: "30"
```

2. 添加后，服务列表中会出现新增的服务，单击**服务 ID/名称**，查看详细的服务信息。

编辑服务

单击**操作** > **编辑**，修改服务信息，确认后即可。

删除服务

单击**操作** > **删除**，确认是否删除服务。确认后将删除服务。

路由

最近更新时间：2026-07-23 17:07:35

操作场景

您需要为需要转发的服务配置路由，以便网关能根据路由匹配条件进行路由匹配，并将请求转发到对应的服务地址从而实现动态代理。本文介绍云原生智能网关路由匹配方式，以及如何为云原生智能网关添加路由并验证转发效果。

路由匹配说明

路由属性

根据不同的协议，云原生智能网关支持配置不同属性用于进行路由匹配：

请求协议	用于路由匹配的属性
HTTP/HTTPS	请求 HTTP 方法（Request HTTP method）
主机名（Host）	—
请求头（Request Header，自定义请求头）	—
请求路径（Request path）	—
请求 SNI（Request SNI），仅用于 HTTPS 场景	—
TCP/TLS 协议	请求目标（Request Destination）
请求 SNI（Request SNI），仅用于 TLS 场景	—
gRPC/gRPCS 协议	主机名（Host）
请求头（Request Header，自定义请求头）	—
请求路径（Request path）	—
请求 SNI（Request SNI），仅用于 gRPCS 场景	—

配置路由时，每个属性都是可选的，但至少需要指定上述属性中的一个。例如，仅指定 Host 为 “`*.example.com`”，则所有来自该域名及其子域名的请求都可能匹配此路由。

路由属性说明

主机名（Host）

云原生智能网关支持以下两种主机名：

- 精确主机名：基于 Host 头进行路由是最常见的方式，可通过路由的 Host 字段配置。例如，配置 Host 为 `["example.com", "service.com"]`，则客户端请求的 Host 头必须为这两个值之一才能匹配该路由。
- 通配符主机名：为了提供灵活性，云原生智能网关支持配置带有通配符的主机名，通配符主机名在域的最左侧 或 最右侧标签中只能包含一个星号。例如 `*.example.com` 将允许主机值（例如 `a.example.com`）与 `x.y.example.com` 匹配。`example.*` 将允许主机值（例如 `example.com`）、与 `example.org` 匹配。

请求头（Request Header）

除 Host 头外，也可以使用其他自定义请求头进行路由。例如，配置 `headers.region=north`，则请求头中包含 `Region: North` 的请求将匹配该路由。

❗ 说明：

多个请求头之间是逻辑“与”关系，而其值是逻辑“或”关系，即请求必须包含所有配置的 headers 键，且对于每个键，其值只需匹配其中一个配置值即可。

请求路径（Request path）

云原生智能网关支持配置以下路径匹配方式：

- 前缀匹配：请求 URL 以配置的路由路径前缀开头即能匹配，例如配置 `/api/v1/`，则 `/api/v1/users`、`/api/v1/orders` 等都可匹配。
- 正则匹配：通过正则表达式定义路由匹配规则，请求 URL 符合正则模式才能匹配，例如 `/api/v1/[a-z]+/[0-9]+` 可匹配 `/api/v1/users/123`。

请求 HTTP 方法（Request HTTP method）

请求方法指根据 HTTP 方法（例如 GET、POST、PUT 等）匹配请求。请求方法支持配置多个值，默认值为空（即不根据方法进行路由）。

例如，配置请求方法为 `["GET", "HEAD"]` 的路由，将匹配 GET 和 HEAD 请求，但不匹配 POST 或 DELETE 请求。您可以为不同的 HTTP 方法配置不同的插件，例如对 GET 请求不进行认证，对 POST 请求应用认证和限流插件。

请求目标（Request Destination）

仅适用于 TCP 和 TLS 路由，通过目标端口进行匹配。

请求 SNI（Request SNI）

在使用安全协议（https、gRPCs、tls）时，Server Name Indication 可作为路由属性。例如，配置 SNI 为 `["foo.test", "example.com"]` 的路由，将匹配在 TLS 连接的 SNI 扩展中设置了相应主机名的传入请求。SNI 路由适用于 TLS 及基于 TLS 的其他协议（如 HTTPS），且在路由匹配时，只要路由中指定的任何一个 SNI 与请求的 SNI 匹配即可。SNI 在 TLS 握手时确定，连接建立后无法修改，因此使用同一 keepalive 连接的多个请求在进行路由匹配时将具有相同的 SNI 主机名，与 Host 头无关。虽然可以创建 SNI 和 Host 头不匹配的路由，但通常不建议这样做。

请求匹配条件

对于一个请求要匹配路由，必须满足以下条件：

1. 请求中必须包含路由配置中设置的所有字段。例如，如果路由配置了 Host 和 paths，则请求中必须有对应的主机名和路径信息。
2. 请求中字段的值必须至少匹配配置值中的一个。例如，路由的 Host 配置为 `["example.com", "test.com"]`，则请求的 Host 头为 “example.com” 或 “test.com” 时满足条件；若 paths 配置为 `["/api/v1/products", "/api/v1/orders"]`，请求路径为 “/api/v1/products/list” 或 “/api/v1/orders/123” 等以配置路径开头的都可匹配。

请求匹配优先级

当一个请求可能匹配多个路由时，按照以下优先级进行匹配：

1. 根据匹配到的路由属性数量进行判断：匹配到的路由属性数量越多，优先级越高。
2. 根据主机名 Host 进行判断：使用精确 Host 的请求>使用通配 Host 的请求。
3. 根据请求头进行判断：匹配到的请求头数量越多，优先级越高。
4. 根据路径进行判断：正则匹配 > 前缀匹配。

对于正则匹配，存在以下优先级：

5. 根据正则优先级（Regex priority）进行匹配：默认为0，值越大优先级越高。
6. 正则优先级相同时，依据创建时间进行判断：创建时间越早，优先级越高。

操作步骤

添加路由

1. 登录 [微服务平台控制台](#)，在左侧导航栏单击云原生网关 > 实例列表。
2. 在实例列表页面，单击目标实例 ID 进入实例详情页，在左侧导航栏单击服务路由，单击服务页签。
3. 点击要配置的服务名称，进入服务路由管理页面。
4. 在左上角单击新建，填写路由基本信息。
 - 当后端请求协议为 HTTP、HTTPS 时，需要配置的参数如下：

参数	是否必选	说明
路由名称	否	路由名称允许为空，支持中文，英文大小写，数字，- _ ~。
请求协议	是	支持 HTTP&HTTPS、HTTPS。
匹配规则	是	请求方法、请求路径、Host 需至少配置一种；Header 可选配置。
Strip Path	否	当通过请求路径匹配到路由时，从后端服务请求 URL 中去除匹配前缀。默认开启。

保留 Host	否	请求后端时是否使用请求 Host，默认关闭。
请求缓冲	否	默认开启。开启时，网关会缓存客户端请求，再分块转发到后端服务；关闭后，网关在接收到客户端请求后将直接转发给后端服务。如果配置的插件需读取请求 body，不支持关闭缓冲。
响应缓冲	否	默认开启。开启时，网关会先缓存后端服务响应，再统一返回客户端；关闭后，网关直接将响应数据返回客户端。如果关联的插件需读取响应 body，不支持关闭缓冲。

- ❗ 说明：
1. 如需配置正则路由，只需将请求路径配置为正则表达式即可。

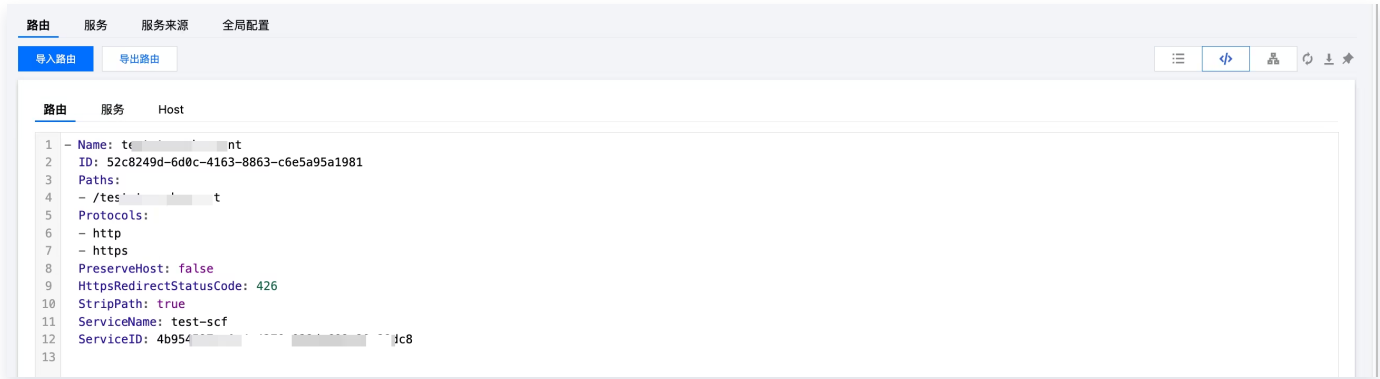
2. Host 大小写敏感。

5. 单击**下一步**，进入路由后端配置，确认后端服务信息无误后提交。
- ☐ 单服务：表示当前路由转发请求到后端单个服务。

☐ 多服务：暂不支持多服务。
6. 单击提交，完成路由创建，在**服务路由 > 路由列表**页面，可以看到创建好的路由，代表创建成功。



7. 在路由列表右上角，您可以切换至 **YAML 视图** 查看信息，



路由优先级检测（可选）

当同一网关下配置了多条路由时，可能存在路由匹配条件重叠，导致请求匹配到非预期路由的情况。为帮助您提前发现此类问题，云原生智能网关提供路由优先级检测功能，用于检测路由之间是否存在优先级冲突。

在**服务路由 > 路由列表**页面，单击**路由优先级检测**，系统将根据当前路由的匹配属性、Host、请求头、路径及正则优先级等，检测是否存在优先级冲突并给出提示。该操作作为可选操作，您可根据检测结果调整路由的匹配条件或正则优先级，以确保请求按预期路由转发。

验证请求转发

发起 API 请求，验证转发到注册中心/容器集群中的服务。

应用场景1：强制 HTTPS

场景说明

HTTPS 通过数字证书、加密算法以及非对称密钥等技术对信息进行加密处理，有效的保证了数据在传输过程中的安全性和完整性。在一些信息安全要求极高的场景下，需要实现只有 HTTPS 请求才能访问服务，如果是 HTTP 请求则需要完成 HTTP 到 HTTPS 的自动跳转，整个过程对用户来讲是透明的、无感知的。

操作方式

1. 创建/编辑路由指定请求协议为 HTTPS，设置重定向状态码为 301（即 Moved Permanently，永久重定向）。

新建路由

1 基本信息配置

>

2 路由后端配置

路由名称

路由名称允许为空，支持中文，英文大小写，数字，.-_~

请求协议

HTTP&HTTPS

HTTPS

匹配规则

请求方法

请选择

请求路径

请输入请求路径

添加路径

Host

请输入 Host

大小写敏感

添加 Host

Header

Header 键

Header 值

支持数字，大小写字母，_-

新增 Header

高级配置

重定向状态码

301

Strip Path

开启

保留 Host

开启

请求缓冲

开启

响应缓冲

开启

下一步

取消

2. 使用 HTTP 请求路由，可以看到请求返回 301 状态码（表示跳转），Location 重定向为 HTTPS 请求。

```
o-3 ~ % curl -i http://1.15.80/httpbin
HTTP/1.1 301 Moved Permanently
Connection: keep-alive
Content-Length: 134
Content-Type: text/html
Date: Wed, 09 Aug 2023 12:21:36 GMT
Location: https://1.15.80/httpbin
Server: kong/2.5.1
X-Kong-Response-Latency: 0

<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
</body>
</html>
```

服务来源

最近更新时间：2026-07-23 17:07:35

操作场景

目前云原生网关支持注册中心（Polaris）、容器服务（TKE）以及私有 DNS 三类服务来源，本文介绍如何将部署了后端服务的容器集群或注册有后端服务的注册中心添加到网关服务来源中，以便网关能获取服务列表，查找对应的服务地址从而实现动态代理。如您的服务使用私有 DNS，可以使用腾讯云私有解析（Private DNS）作为服务来源，从而实现服务私有域名解析。

前置条件

已购买云原生网关实例，相关操作详情请参见 [新建网关](#)。

操作步骤

添加服务来源

1. 登录 [微服务平台控制台](#)。
2. 在左侧导航栏单击云原生网关 > 实例管理，选择网关实例进入详情页，选择左侧服务路由，单击服务来源页签，在服务来源页面单击新建。
3. 在新建服务来源窗口中配置相关参数。

参数	是否必填	说明
来源类型	是	服务来源类型，支持注册中心、容器服务和私有 DNS 三类来源。

注意：
目前云原生网关仅对微服务场景，即作为 Polaris（北极星）服务接入层支持开通白名单新购，其余场景不支持新接入。非上述场景访问控制台时将重定向至 Polaris（北极星）控制台。

- 当服务来源为注册中心时，填写以下参数：

参数	是否必填	说明
来源产品	是	来源产品，支持 TSF 注册中心和自建注册中心，注册中心类型包括 Nacos，Consul，PolarisMesh 和 TSF 注册中心。 说明： 推荐使用 TSF PolarisMesh。

选择实例	是	当来源产品为 TSF 注册中心时，选择后端服务所在的注册中心实例。
来源名称	是	当来源产品为自建注册中心时，填写来源名称。
访问地址	是	当来源产品为自建注册中心时，填写访问地址，多个访问地址使用换行分隔。针对不同注册中心，访问地址书写方式为： - 自建 Nacos：填写域名或 IP：Port - 自建 Consul：填写域名或 IP：Port - 自建 PolarisMesh：填写 http 和 grpc 访问地址作为一组访问地址，每组格式为：<code>http://IP:{HTTP_PORT}</code>、<code>grpc://IP:{GRPC_PORT}</code>
用户名	否	对于类型为 Nacos 的注册中心，如开启了注册中心鉴权功能，填写访问用户名。
密码	否	对于类型为 Nacos 的注册中心，如开启了注册中心鉴权功能，填写访问密码。
访问 Token	否	对于类型为 Consul 和 Polaris 的注册中心，如开启了注册中心鉴权功能，填写访问 Token。

○ 当服务来源为容器服务时，填写以下参数：

参数	是否必填	说明
来源产品	是	来源产品，支持 TKE 容器服务。
选择实例	是	选择后端服务所在的容器集群。

○ 当服务来源为私有 DNS 时，选择腾讯云私有解析即可使用 Private DNS 进行域名解析。

4. 添加后，查看服务来源列表，包含新建的服务来源信息，则说明服务来源创建成功。

删除服务来源

单击操作 > 删除，确认是否删除服务来源。确认后将删除服务来源。

 注意

如果当前服务来源已有服务关联，则不允许删除，需删除服务后重试。

灰度发布

最近更新时间：2026-07-23 17:07:35

操作场景

灰度发布是软件上线过程中常见的上线方式，是指在发布过程中，将具有一定特征或者比例的流量分配到需要被验证的版本中，用来观察新的验证版本的线上运行状态。相比全量上线，灰度发布是更加谨慎的发布形式。当线上调用链路较为复杂时，全链路灰度发布可以将线上的各个服务隔离出一个单独的运行环境。灰度发布功能仅专业版支持。使用云原生智能网关的灰度策略能力，可以在不修改任何业务代码的情况下，可视化配置灰度规则，实现云上轻松易用的全链路灰度发布。本文介绍如何在云原生智能网关控制台上配置灰度发布策略，包括以下两种场景：

- 配置参数路由，以实现根据请求特征（header、query、cookie、path、method 等）进行请求转发。
- 配置百分比路由，以实现根据比例转发到不同的服务。

前置条件

- 已购买云原生智能网关实例，灰度发布功能仅专业版支持说明，详情请参见 [操作文档](#)。
- 配置了后端 [服务（Service）](#) 以及 [路由（Route）](#)。

操作步骤

步骤1：更新路由插件

在配置灰度发布策略前，确保您的 tse-route（路由转发）插件已升级到最新版本。

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关 > **实例列表**，单击目标实例 ID 进入详情页，在左侧单击**插件管理**。
3. 单击**系统插件**页签，查看插件是否已更新到最新版本，若未更新，单击插件名称进入插件详情页，在详情页中单击**安装最新版本**即可。

步骤2：创建全局配置（可选）

说明：
当灰度策略中的逻辑关系选择了包含或者不包含，参数值可选择全局配置，此时需要提前创建好全局配置。

1. 在左侧导航栏选择**服务路由** > **全局配置**页签，单击**新建**。
2. 在新建配置页面，输入配置名称和配置内容，配置内容可以选择**直接导入文件**。

新建配置

配置名称

请输入配置名称

类型

限流配置

灰度配置

Redis 配置

JSON 格式，用于配置精细化限流阈值。[下载模版](#)

配置内容

请输入配置内容或上传文件

文件导入

提交

关闭

3. 单击提交，完成创建。

步骤3：配置灰度策略

1. 在左侧导航栏选择服务路由 > 服务页签，单击服务名称，进入服务详情页面。
2. 选择灰度策略页签，选择标准灰度规则，单击创建规则，配置规则信息。
 - 优先级：支持0-100，数值越大优先级越高，各规则优先级不允许重复。支持配置多条规则，网关将按优先级进行规则匹配，匹配到了即转发到对应的目的后端。
 - 是否启用：启用后，创建好的灰度策略将生效。
 - 匹配条件：路由规则内的多个条件为逻辑与（AND）关系，即一个路由规则内条件都满足了，才会转发到对应的后端服务。

参数	说明
参数类型	选择参数类型，支持的参数类型说明如下： - 支持特定位置的请求参数，目前只支持 Header、Path、Body、Query、Cookie 参数。 ⚠ 注意： - Body 参数为请求 JSON 中的参数，仅支持 content-type 为 application/json 类型。 - Body 参数仅对 JSON 第一层进行解析，当存在同名键，会选择后一个参数值。 - 支持请求方法（method）匹配，满足数组中的一个方法即可。 - 支持以下常量类型： - STRING: 字符串类型，支持单引号或双引号，如："Hello"、'hello'。

	○ NUMBER: 数字类型，如： 0.1, 100.0, 1。 ● 支持以下系统参数： ○ domain: 请求的域名。 ○ clientIp: 客户端 IP。 ○ httpScheme: 请求的协议 HTTP,HTTPS,WS,WSS。 ○ clientUa: 客户端上传的 UserAgent 字段。
参数名	输入参数名称。
逻辑关系	选择逻辑关系： ● 支持存在与正则表达式 ● 支持<, <=, >, >=, ==, != 运算符。 ● 支持包含和不包含。 ● 不支持双目运算。 ● 不支持数学运算。
参数值	输入参数值。当逻辑关系选择了 包含或者不包含 ，参数值可选择 手动输入或者全局配置 ，若无全局配置可参考全局配置文档创建配置。

- 目标服务：路由规则内条件都满足后，路由转发到的目标后端服务。支持一个或多个后端服务，所有后端服务百分比之和必须为100%。

3. 单击**确定**，完成规则创建。

创建规则

① 流量将按照优先级进行匹配，优先级数字越大越先匹配。

▼ 优先级 ①

是否启用

规则

匹配条件

目的服务

Query参数 | name == test

+ 添加条件

100% → pod

+ 添加目的服务

+ 添加规则

步骤4：验证策略是否生效

请求服务，网关将根据请求参数动态路由到对应的目的后端。

限流策略

最近更新时间：2026-07-23 17:07:34

操作背景

限流主要是通过设置限流规则保护后端服务，防止瞬时流量过大造成服务和数据崩溃，导致服务不可用。当资源成为瓶颈时，需要对请求做限流，启动流控保护机制。TSF 云原生智能网关支持**服务级**和**路由级别**的限流，本文介绍限流规则的配置方式。

前提条件

在配置限流策略前，请确保您的 tse-rate-limiting（流量控制）插件已升级到最新版本。如果需对限流阈值进行精细化配置，请确保您的 tse-global-configuration（全局配置）插件已升级到最新版本。

1. 登录 [TSF 控制台](#)，在左侧导航栏单击**云原生网关 > 实例管理**，进入实例列表页面。
2. 单击实例 ID/名称进入实例详情页。
3. 选择**插件管理 > 系统插件**页签，查看插件是否已更新到最新版本。若未更新，单击插件名称进入插件详情页，在详情页中单击**安装最新版本**即可。

服务级限流

服务级限流对服务下所有路由均生效，如需对路由配置限流，您可以前往服务路由页配置。

步骤1：配置限流策略

1. 登录 [TSF 控制台](#)，在左侧导航栏单击**云原生网关 > 实例管理**，进入实例列表页面。
2. 单击实例 ID/名称进入实例详情页。
3. 单击**服务路由**，选择**服务**页签，单击服务名称，进入服务详情页面。
4. 在页面上方选择**限流策略**页签，单击**配置限流规则**。

参数	是否必选	说明
是否启用	是	开启后，限流规则将会生效。
限流配置	是	● 基础限流：根据请求数阈值进行流量控制。当该服务上的请求数量在设定的时间周期内大于阈值将触发限流。支持秒、分钟、小时、天、月、年六种时间维度。 ● 参数限流：根据用户请求参数及指定条件进行流量控制。 ○ 过滤条件（可选）：仅对符合一定过滤条件（IP 或 Path）的请求才进行限流统计。 ○ 限流条件：统计的限流参数达到指定阈值即触发限流，支持依据 Header、Cookie、Path、IP、Query 参数进行流量控制，支

		持设置固定限流阈值以及使用全局配置针对不同限流参数值进行精细化配置。
计数策略	是	- 当选择默认 Redis 时，计数器保存在 Kong 实例自带的 Redis 服务节点实现分布式限流，支持300万个 Key。 - 当选择外部 Redis 时，可配置 Redis 访问地址（服务器地址、连接密码、超时时间），实现更大规模的分布式限流。
限流处理策略	是	- 直接返回：触发限流直接返回 429。 - 自定义返回：触发限流后，自定义 HTTP 应答行为，需输入状态码、响应头和响应内容。 - 请求转发：触发限流后，跳转到指定页面，需输入请求转发地址。
请求排队	否	请求处理的最大排队时间，支持 0–15 秒。配置为 0，代表不延迟处理。配置大于 0，每隔 1 秒网关重试处理该请求，如果达到最大排队时间还未被处理，则请求被限流。
隐藏限流响应头	否	默认开启，隐藏后客户端将不会收到限流相关响应头。

新建限流规则

服务 ID: cd7

服务名称:

是否启用

☒

限流配置

符合基础限流或参数限流 任意 规则即进行限流:

☒ 基础限流

限流阈值

每

分钟

最多

3

个请求

添加规则

☒ 参数限流

过滤条件

符合以下 所有 条件的请求才会进行限流统计:

限流参数

Header.name

添加

普通限流阈值

每秒最多2个请求

添加

精细限流阈值

秒: clare

添加

添加规则

计数策略

默认 Redis

限流处理策略

直接返回

自定义返回

请求转发

触发限流直接返回 429

说明:

- 精细限流阈值适用于不同限流参数有不同限流阈值的场景，例如，根据 Query 请求参数 appid 来进行限流，当 appid = 1时，每分钟限流100，当 appid = 2时，每分钟限流200。
- 精细限流阈值仅在配置了普通限流阈值后才支持配置，普通限流阈值会作为未匹配到指定请求参数的默认限流阈值。

3. 精细限流阈值使用全局配置定义，使用 JSON 格式描述，请按照限流 key:限流阈值格式配置，使用下划线(_)按顺序连接全部限流参数值作为限流 key。例如，当限流依据为 Query 参数 appid，当 appid=1 时，限流阈值为 100，appid=2 时，限流阈值为 200，全局配置如下：

```
{
  "1": 100,
  "2": 200
}
```

5. 单击**确定**，完成规则创建。

步骤2：验证策略是否生效

发起请求触发限流规则，查看限流是否生效。

路由级限流

路由级限流仅对本路由生效，如需配置服务限流，您可以前往服务级限流策略页配置。

步骤1：配置限流策略

1. 登录 [TSF 控制台](#)，在左侧导航栏单击云原生网关 > **实例管理**，进入实例列表页面。
2. 单击实例 ID/名称进入实例详情页。
3. 单击**服务路由**，选择**路由**页签，单击路由名称进入路由详情页。
4. 选择**限流策略**页签，进入路由级限流页面，单击**配置限流规则**。

参数	是否必选	说明
是否启用	是	开启后，限流规则将会生效。
限流配置	是	● 基础限流：根据请求数阈值进行流量控制。当该服务上的请求数量在设定的时间周期内大于阈值将触发限流。支持秒、分钟、小时、天、月、年六种时间维度。 ● 参数限流：根据用户请求参数及指定条件进行流量控制。 ○ 过滤条件（可选）：仅对符合一定过滤条件（IP 或 Path）的请求才进行限流统计。 ○ 限流条件：统计的限流参数达到指定阈值即触发限流，支持依据 Header、Cookie、Path、IP、Query 参数进行流量控制，支持设置固定限流阈值以及使用全局配置针对不同限流参数值进行精细化配置。

计数策略	是	- 当选择默认 Redis 时，计数器保存在 Kong 实例自带的 Redis 服务节点实现分布式限流，支持300万个 Key。 - 当选择外部 Redis 时，可配置 Redis 访问地址（服务器地址、连接密码、超时时间），实现更大规模的分布式限流。
限流处理策略	是	- 直接返回：触发限流直接返回 429。 - 自定义返回：触发限流后，自定义 HTTP 应答行为，需输入状态码、响应头和响应内容。 - 请求转发：触发限流后，跳转到指定页面，需输入请求转发地址。
请求排队	否	请求处理的最大排队时间，支持 0–15 秒。配置为 0，代表不延迟处理。配置大于 0，每隔 1 秒网关重试处理该请求，如果达到最大排队时间还未被处理，则请求被限流。
隐藏限流响应头	否	默认开启，隐藏后客户端将不会收到限流相关响应头。

5. 单击**确定**，完成规则创建。

步骤2：验证策略是否生效

发起请求触发限流规则，查看限流是否生效。

健康检查

最近更新时间：2026-07-23 17:07:34

操作背景

云原生网关提供健康检查机制，通过配置健康检查相关策略，网关会对服务节点进行“探活”、“探死”，当服务节点发生故障时，可将请求代理到健康的节点上，以增强网关服务的连续性。

⚠ 注意：

云原生网关健康检查机制仅适用于 IP 列表类型的服务，K8S 服务建议使用容器健康检查机制。

健康检查的方式

云原生网关提供两种健康检查的方式：主动检查和被动检查。

主动检查（active checks）

主动检查是指网关通过预设的探针类型，主动探测服务节点的存活性。当启用主动健康检查后，网关会按照预设的探测周期，向每个服务节点配置的路径发出 HTTP 或 HTTPS 请求，网关将根据探测结果自动启用处于健康状态的节点，并禁用不健康的节点。

被动检查（passive checks）

被动检查也被称为断路器模式，该模式下网关会基于网关代理的（HTTP/HTTPS/TCP）进行检查，不会主动发起对节点的探测。当节点无响应时，被动健康检查将检测该目标并将其标记为不健康。

❗ 说明：

由于该模式下不健康的节点无法收到请求，即使节点恢复健康，网关也无法探测到该节点恢复进而重新将该节点标记为健康，只能通过管理端 API 手动触发并通知健康检查器再次启用该节点。因此通常需要结合主动健康检查策略。

如何配置健康检查

1. 登录 [TSF 控制台](#)，在左侧导航栏单击云原生网关 > 实例管理，进入实例列表页面。
2. 单击实例 ID/名称进入实例详情页。
3. 单击服务路由，选择服务页签，可在服务列表页或服务详情页配置健康检查。

路由	服务	服务来源	全局配置
新建			
ID/名称	健康检查	服务类型	
Odf419ea-16c8-4654-a1... testttt	未开启	IP 列表	

配置主动健康检查

在网关服务列表页面，单击健康检查列处的编辑按钮，进行健康检查配置。

- 被动健康检查：通过判断上游节点的响应状态确定节点健康状态。
- 主动健康检查：网关按照预设的探测周期，主动探测服务节点的存活性。
 - 请求路径：网关向目标节点发出主动探测请求时使用的路径，默认为“/”。
 - 健康检查间隔：两次对健康节点主动进行健康检查的间隔时间，支持 2–300 秒。
 - 异常检查间隔：两次对异常节点主动进行健康检查的间隔时间，支持 2–300 秒。
- 健康 HTTP 状态码：用于判断健康检查正常的 HTTP 状态码，返回上述 HTTP 状态码，节点将被标记为健康。健康 HTTP 状态码自动填充以下健康 HTTP 状态码：
 - 2xx：200, 201, 202, 204
 - 3xx：301, 302, 303, 304, 307, 308
 - 4xx：400, 401, 403, 404, 410, 411, 429, 499
- 异常 HTTP 状态码：用于判断健康检查异常的 HTTP 状态码，返回上述 HTTP 状态码，节点将被标记为异常。异常 HTTP 状态码自动填充以下异常 HTTP 状态码：
 - 5xx：500, 502, 503, 504
- 超时阈值：连续健康检查超时次数，超出该值认为该节点不健康，默认 10 次，可配置 1–10 次。
- 连续健康检查阈值：连续健康检查成功次数，超出该值认为该节点健康，默认 1 次，可配置 1–10 次。
- 连续异常阈值：连续健康检查失败次数，超出该值认为该节点异常，默认 10 次，可配置 1–10 次。

说明：

首次开启健康检查，网关将根据健康 HTTP 状态码范围进行节点健康状态标记，请确保 HTTP 状态码配置正确。

配置健康检查



首次开启健康检查，网关将根据健康 HTTP 状态码范围进行节点健康状态标记，请确保 HTTP 状态码配置正确。

服务名称

大模型

被动健康检查

为保证节点可用性，被动健康检查需与主动健康检查配合使用

健康检查协议

HTTP TCP

主动健康检查



请求路径

/

1-200个字符，以 / 开头，支持a-zA-Z0-9._/=?

健康检查间隔

2 秒

支持 2-300 秒

异常检查间隔

2 秒

支持 2-300 秒

健康 HTTP 状态码

200 201 202 204 301 302 303 304 307 308 400 401 403 404 410 411 429 499

返回上述 HTTP 状态码，节点将被标记为健康，请确保添加全部认为节点为健康状态的 HTTP 状态码

异常 HTTP 状态码

500 502 503 504

返回上述 HTTP 状态码，节点将被标记为异常，请确保添加全部认为节点为异常状态的 HTTP 状态码

超时阈值

10 次

支持 1-10 次

连续健康阈值

1 次

支持 1-10 次

连续异常阈值

10 次

支持 1-10 次

确定

取消

监控健康检查状态

配置完健康检查后，在左侧导航栏选择数据观测，页面顶部选择业务监控 > 业务指标：

- 通过服务筛选，可查看该服务下的节点健康状况。
- 通过实例视图筛选，可查看该实例下所有服务的节点健康状况聚合情况。
- 支持为健康检查健康指标配置告警。



查看节点状态

在服务详情页面的**服务信息 > 节点信息**模块，可以查看到该服务下各节点的健康状况。您可以在节点列表的操作栏手动修改节点健康状态。

节点信息				
节点地址	权重	健康状况	创建时间	操作
47.22.22.2222	100	异常	2023-11-02 02:29:03	标记为健康
7.22.22.2222	100	异常	2023-11-02 02:29:03	标记为健康
4.22.22.2222	100	异常	2023-11-02 02:29:03	标记为健康

服务不支持健康检查的原因

- 域名/IP 类型不支持
原因：节点地址是动态 DNS 解析的域名，网关拿到的不是固定的 IP 列表，而是域名。每次请求可能解析到不同的后端 IP，健康检查的"探活/探死"目标无法稳定指向具体节点。
- 注册中心（TSF PolarisMesh）类型不支持
原因：节点上下线、存活性由注册中心自身通过心跳机制维护，PolarisMesh 会主动剔除不健康实例，网关拿到的是注册中心过滤后的"已健康"节点列表，再做一层健康检查是冗余且会与注册中心判定结果冲突。

全局配置

最近更新时间：2026-07-23 17:07:34

云原生网关支持使用全局配置存放配置内容，用于配置限流规则或者灰度策略。

- 限流配置：使用全局配置定义不同参数值的限流阈值，以实现更精细化的流控。
- 灰度配置：用于配置灰度名单，当灰度策略的逻辑关系选择了**包含**或者**不包含**，参数值可选择全局配置。

操作步骤

新建全局配置

1. 登录 [微服务平台控制台](#)。
2. 在左侧导航栏单击**云原生网关**进入实例列表，单击目标实例 ID/名称进入实例详情页。
3. 单击左侧**服务路由**。
4. 选择**全局配置**页签，单击**新建**，您可以直接输入配置内容，也可以下载模板填写后，导入控制台进行提交。
 - 限流配置：JSON 格式，用于配置精细化限流阈值。
 - 精细限流阈值适用于不同限流参数有不同限流阈值的场景，例如，根据 Query 请求参数 appid 来进行限流，当 appid=1 时，每分钟限流100，当 appid=2 时，每分钟限流200。
 - 精细限流阈值仅在配置了普通限流阈值后才支持配置，普通限流阈值会作为未匹配到指定请求参数的默认限流阈值。
 - 精细限流阈值使用全局配置定义，使用 JSON 格式描述，请按照限流 key:限流阈值格式配置，使用下划线(_)按顺序连接全部限流参数值作为限流 key。例如，当限流依据为 Query 参数 appid，当 appid=1 时，限流阈值为 10，appid=2 时，限流阈值为 20，全局配置如下：

```
{
  "headername1_cookie1_10.0.0.1_/path1_queryname1": 10,
  "headername2_cookie2_10.0.0.2_/path2_queryname2": 20
}
```

- 灰度配置：文本格式，用于配置灰度名单，支持英文逗号，英文分号，空格，换行符作为分隔符。配置灰度规则时，当逻辑关系选择了**包含**或者**不包含**，参数值可选择手动输入或者**全局配置**。
 - Redis 配置：Redis 服务配置，网关实例中使用的 Redis 服务信息，包括服务地址，服务用户名与密码。配置后网关实例可直接使用此 Redis 服务。
5. 单击**确定**，完成规则创建。

后续步骤

全局配置创建完成后，可用于配置限流规则或者灰度策略，具体操作步骤请参考：

- [配置限流规则](#)
- [配置灰度策略](#)

证书管理

最近更新时间：2026-07-23 17:07:32

操作场景

本文档介绍如何在微服务平台 TSF 控制台上管理您的 SSL 证书，包括新建、删除、编辑证书等操作。

前提条件

1. 已经完成域名注册、实名认证、备案等流程，若您尚未完成，可以使用腾讯云提供的域名服务，详情请参见 [快速注册域名及实名认证](#)。
2. 已经拥有 SSL 证书，若您暂无证书，可以使用腾讯云提供的证书服务，详情请参见 [证书申请指南](#)。

新建证书

步骤1：配置 DNS 解析

1. 前往 [TSF 控制台](#)。
2. 选择云原生网关 > 实例管理，选择需要访问的实例，进入实例详情页，选择需要访问的集群分组，查看[网络配置](#) > 公网负载均衡，获取访问 IP。
3. 根据 [快速添加域名解析](#) 指引，将您的域名解析至网关公网代理地址中的 IP。

步骤2：新建证书

1. 在左侧导航栏选择证书管理，单击新建。
2. 填写证书名称，选择证书并填写您的域名。仅支持 SVR 类型证书。

新建证书

证书名称

1-80个字符，支持中文、英文、数字、下划线、分隔符“-”、小数点

证书来源

腾讯云 SSL 平台

查看托管证书详情或申请免费证书，前往 [SSL证书管理](#)

选择证书

请输入ID、备注或域名进行搜索

已选择：暂未选择

证书 ID	备注	域名	到期时间 ↓
BmsPZy5Q			2024-09-07 15:45:17
Bms6yRJG			2024-01-21 16:22:11
OyVuaCkv			2025-01-02 13:44:17
OnvEihwV			2025-01-02 13:44:17

仅支持 SVR 类型证书

绑定域名

请输入域名

添加域名

确定

取消

3. 单击**确定**，完成证书新建，返回证书列表查看创建好的证书。

步骤3：验证证书是否生效

访问 `https://域名` 验证是否生效。

更新证书

在证书过期前，您可以更新证书。

1. 在证书列表页面，选择您要更新的证书，单击操作栏的**更新证书**。
2. 在弹窗中选择更新的证书，确认更新后的证书信息，单击**确定**，完成更新。

编辑证书

您可以修改证书的证书名称和绑定的域名。

1. 在证书列表页面，选择您要更新的证书，单击操作栏的**编辑**。
2. 在弹窗中输入证书名称和绑定域名，单击**确定**，完成修改。

编辑证书

证书名称

test

1-80个字符，支持中文、英文、数字、下划线、分隔符“-”、小数点

证书来源

4myM0Xdd (腾讯云 SSL 平台)

绑定域名

test.com

添加域名

确定

取消

删除证书

⚠ 注意：

删除后，该证书绑定的域名将无法使用 HTTPS 访问网关。

1. 在证书列表页面，单击您要更新证书操作栏的删除。
2. 在弹窗中单击**确定**，完成删除。

域名管理

最近更新时间：2026-07-23 17:07:32

本文将快速引导您如何在云原生网关中，使用自定义的域名访问 API 网关。云原生网关提供了 IP 访问的方式，并没有提供域名接入点。因此您需要在自己的域名 DNS 添加一个 A 记录，指向云原生网关提供的 IP。

概述

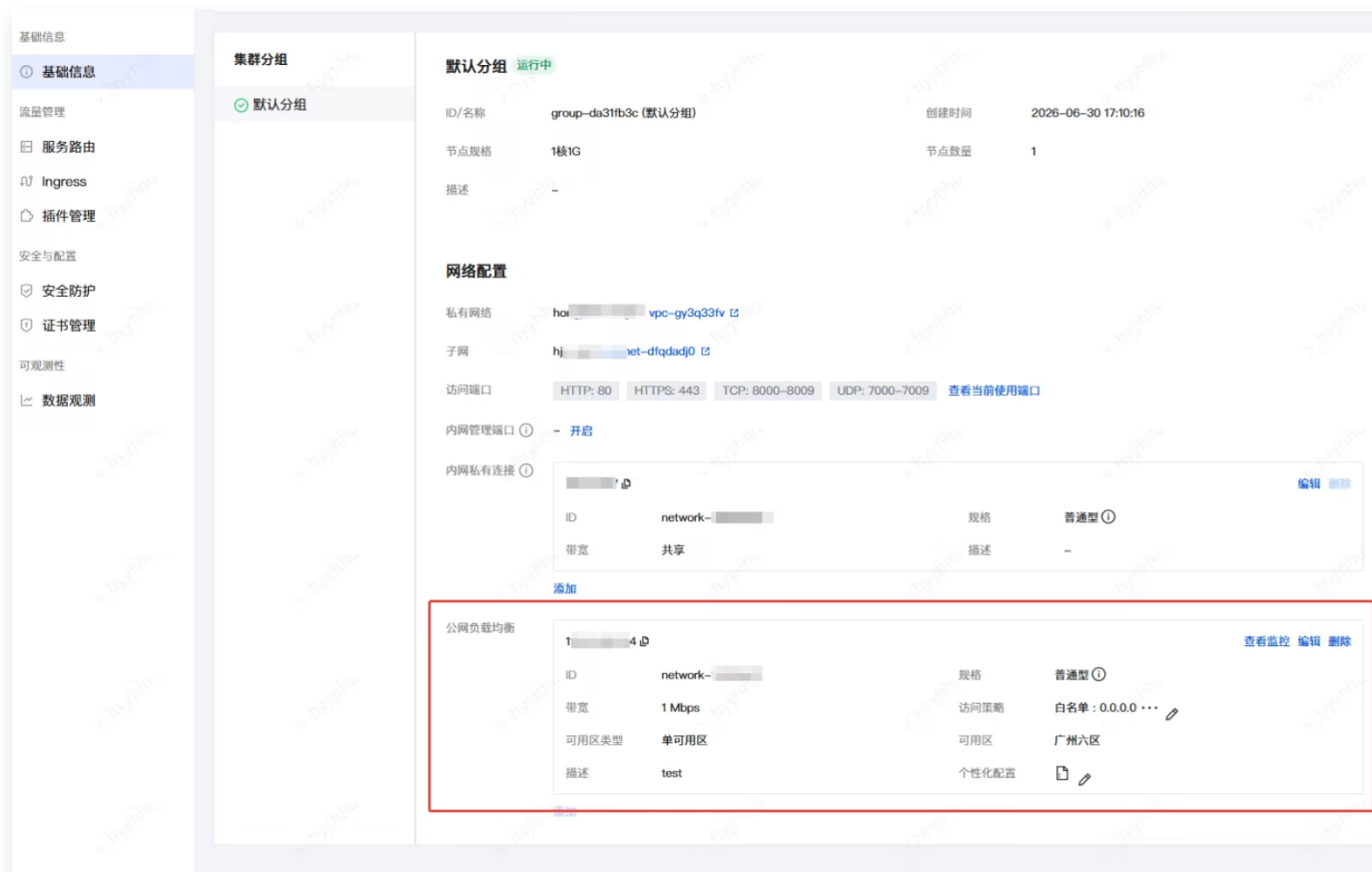
您需要依次完成以下步骤：

1. 在 DNS 中 添加 A 记录，绑定云原生网关提供的 IP。
2. 绑定证书（如果有使用 HTTPS 访问网关的需要）。

操作步骤

步骤1：在 DNS 中添加 A 记录，绑定云原生网关提供的 IP

1. 前往 [云原生网关控制台](#)，选择 **云原生网关 > 实例管理**，选择需要访问的实例，进入实例详情页。
2. 选择需要访问的集群分组，查看 **网络配置 > 公网负载均衡**，获取访问 IP。



3. 在 DNS 中添加记录，将域名绑定该公网 IP。

步骤2：绑定证书（如果有 HTTPS 访问的需要）

如果您有使用 HTTPS 访问网关资源的需要，请参考 [证书管理](#) 步骤绑定证书。

 注意：

如果没有绑定证书，HTTPS 协议访问自定义域名的时候，会出现证书跟自定义域名不一致的情况。如果您是使用 curl 工具进行测试，使用 `-k` 的参数跳过对服务端证书的校验。在生产的环境，使用 HTTPS 需要绑定证书。

Ingress

Kong Ingress介绍

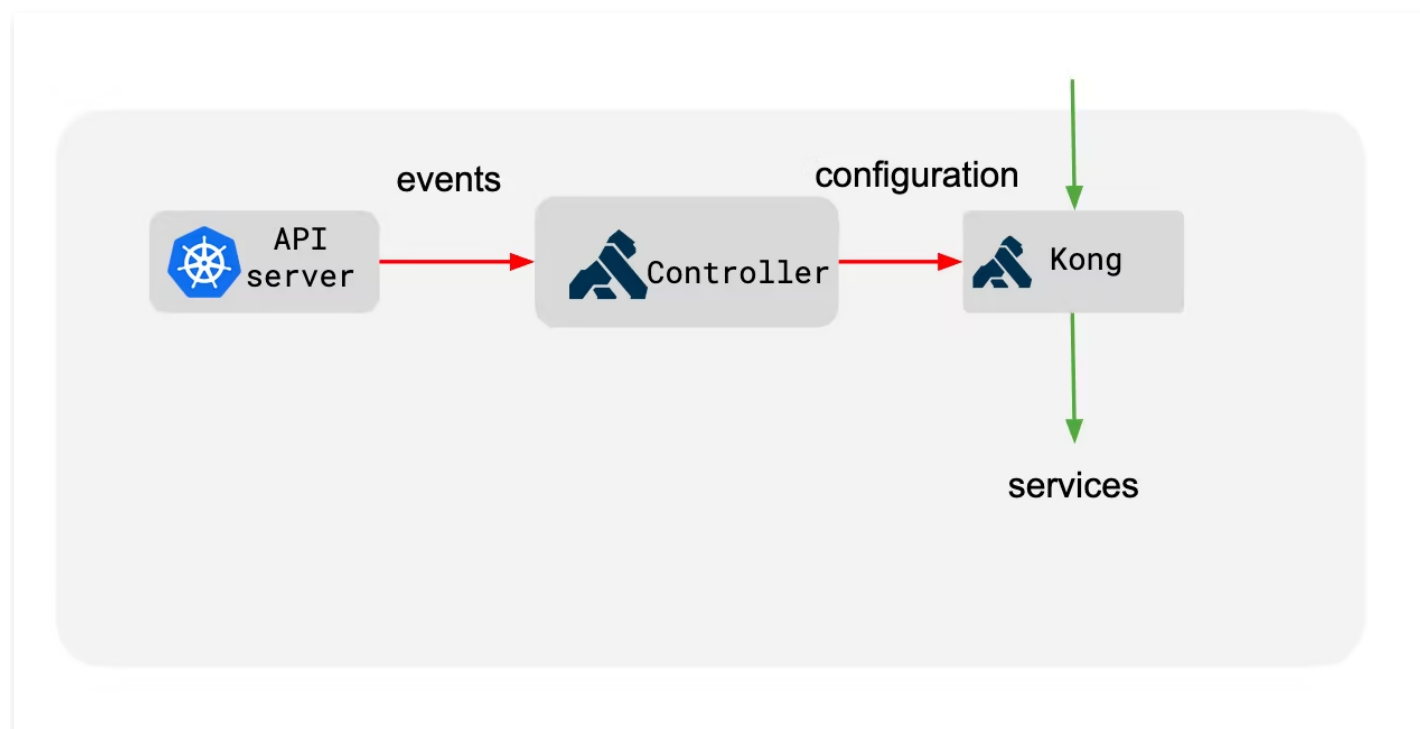
最近更新时间：2026-07-23 17:07:34

Kong Ingress Controller 介绍

Kong 网关一般通过 Admin API 进行资源管理，可管理的资源包括 Service、Route、Plugin、Consumer 等，在定义资源规则后，Kong 将按照这些配置规则进行对上游服务的请求进行路由分发和控制。

在 Kubernetes 集群环境下，Admin API 方式不是很适应 Kubernetes 声明式管理方式。所以 Kong 在 Kubernetes 集群环境下推出 Kong Ingress Controller，涵盖了原 Admin API 的各个方面。因此，Kong Ingress Controller 的作用不仅仅是代理进入 Kubernetes 集群的流量，也可以实现插件配置、负载平衡、健康检查等 Kong 提供的所有功能。

下图展示了它的工作原理：



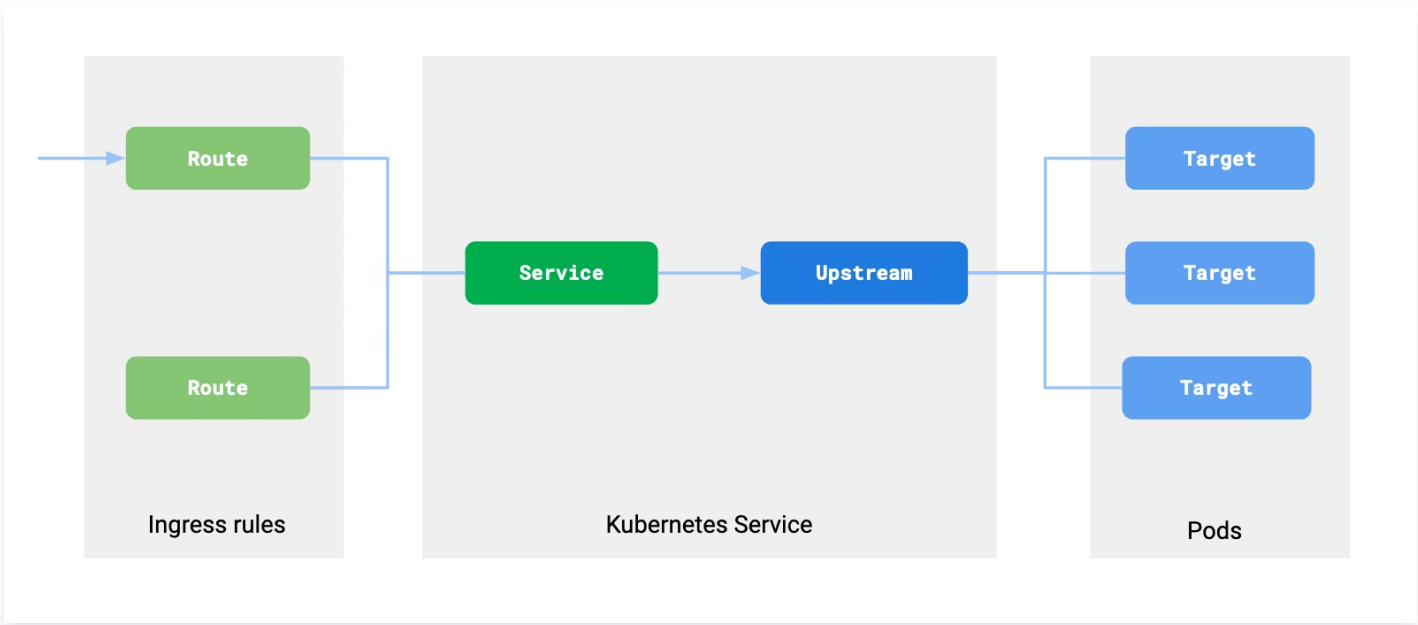
当您安装了 Kong Ingress Controller，Controller 管理器将监听 Kubernetes 集群内部发生的变化，并更新 Kong 以响应这些变化，从而能够正确代理所有流量。与此同时，Kong 也会根据 Controller 同步的信息进行动态更新，以响应 Kubernetes 集群内发生的扩展、配置更改和故障等变化。

Kubernetes 资源与 Kong 资源对应关系

Kong Ingress Controller 旨在将您配置在 Kubernetes 集群中的资源同步到 Kong，并生成对应的 Kong 资源以便能够使用 Kong 原有机制进行网关操作。两者资源的对应关系如下：

- Kubernetes Service：对应 Kong Service 和 Kong Upstream
- Kubernetes Pod：对应 Kong Target

- Kubernetes Ingress：对应 Kong Route



Kubernetes 版本与 Kong Ingress Controller 版本对应关系

Ingress 版本	1.3.4	2.5.0	2.7.0
Kubernetes 1.14	✓	—	—
Kubernetes 1.16	✓	—	—
Kubernetes 1.17	—	—	—
Kubernetes 1.18	✓	✓	—
Kubernetes 1.19	—	—	—
Kubernetes 1.20	✓	✓	—
Kubernetes 1.21	—	—	—
Kubernetes 1.22	—	✓	✓

Kubernetes 1.23	—	—	—
Kubernetes 1.24	—	✓	✓
Kubernetes 1.25	—	—	—
Kubernetes 1.26	—	—	✓
Kubernetes 1.27	—	—	—
Kubernetes 1.28	—	—	✓

使用 Kong Ingress Controller

最近更新时间：2026-07-23 17:07:35

操作场景

Kong 通过 Ingress Controller 实现为 Kubernetes Service 配置插件、健康检查、负载均衡等功能。通过整合 Kong Ingress Controller，Kong 可直接关联到 Kubernetes 整个生命周期，Ingress Controller 将监听容器集群变化，并更新 Kong 的配置，以便能正确代理所有的流量，免去人工管理的困扰。

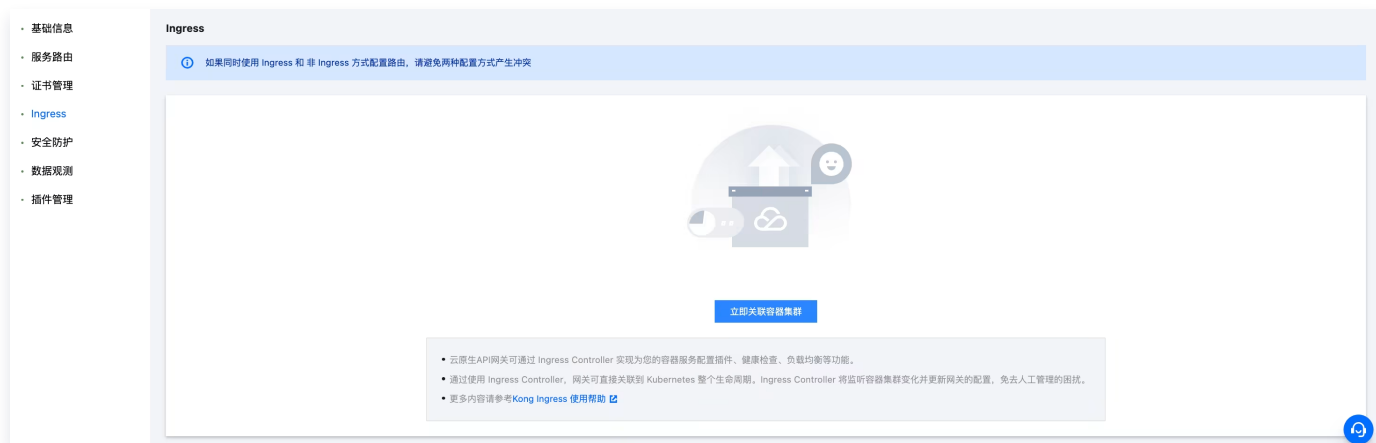
前提条件

- 已创建云原生网关实例。
- 已购买腾讯云容器服务，包括 [TKE 标准集群](#) 或 [TKE Serverless 集群](#)。

操作步骤

步骤1：启用 Kong Ingress Controller

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击 [云原生网关](#) > [云原生网关](#)，单击目标实例进入实例详情，点击左侧 [Ingress](#)
3. 未关联容器集群时，页面提示需关联容器集群。



4. 单击[立即关联容器集群](#)，选择对接集群类型和集群信息，单击[确定](#)，即启用 Kong Ingress Controller。

关联容器集群

所属 VPC

sh-test | vpc-2r

仅支持与网关相同的VPC或使用云联网、对等连接打通的VPC

Ingress 版本

2.7.02.5.01.3.4

2.7 支持 K8s 1.22-25，支持配置多个容器集群，请提交工单升级实例后重试

选择集群

TKE 标准集群TKE Serverless集群

请输入集群名称

新建集群

ID/名称	集群版本	状态	所属 VPC	连通性检测
cl-06	1.26.1	运行中	vp-01	未通过
cl-e	1.22.5	运行中	vp-p	未通过
cl-ck	1.22.5	运行中	vp-f	未通过

已选择 (0)

ID/名称	集群类型
暂无数据	

高级配置

IngressClass

kong

不超过 253 个字符，只能包含小写字母、数字、 '-' 和 '.'，须以字母数字开头和结尾

确定

取消

说明：

请选择与网关实例网络连通的容器集群，包括在相同 VPC，使用云联网或对等连接打通的 VPC。

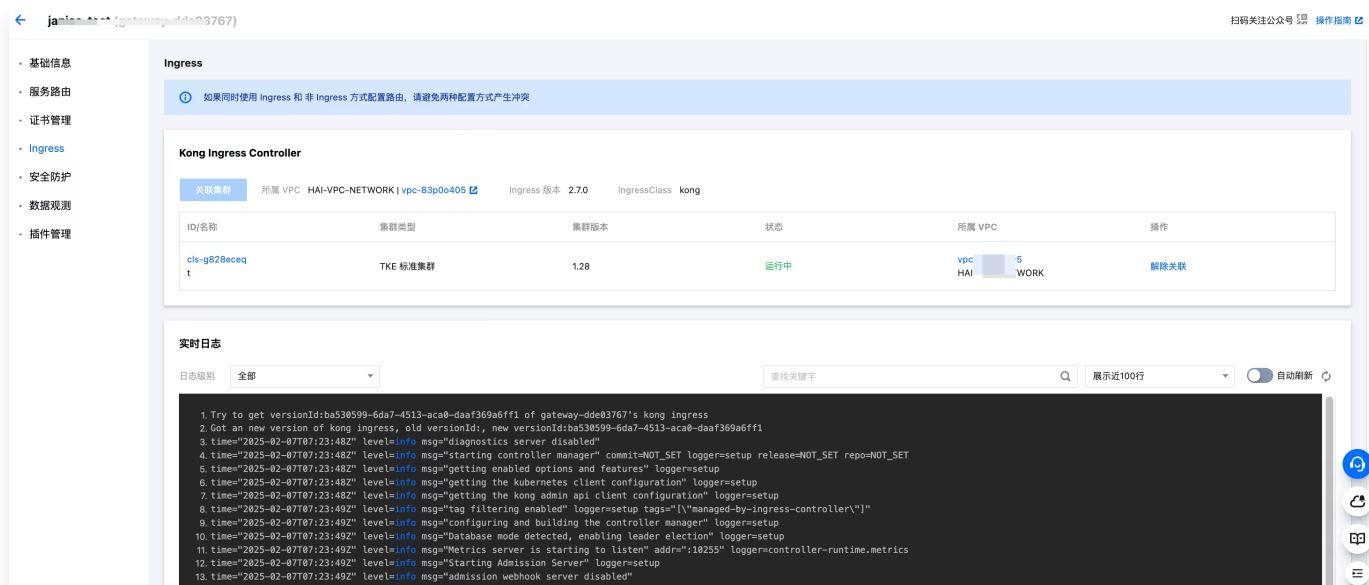
Ingress 版本支持 2.7.0，2.12.0，2.5.0 和 1.3.4，仅 Ingress 2.7.0 版本支持关联多容器集群。

IngressClass 默认为 kong，支持配置自定义 IngressClass，用来标识网关实例。

5. 确认已展示选择的容器集群信息。

版权所有：腾讯云计算（北京）有限责任公司

第84 共261页



6. 创建对应 Ingress 规则（按照下方信息创建），请配置 Kubernetes Ingress 资源中的 ingress.class 为步骤 4 中配置的 IngressClass 名称。

⚠ 注意：

对于 apiVersion 为 v1beta1 和 v1 版本的 Ingress 资源，配置方式有所不同，请参考如下示例进行配置。有关不同版本的详细区别，请参见 [Kong 官方文档说明](#)。

- 对于 apiVersion 为 v1beta1 版本的 Ingress 资源，请参考如下示例进行配置：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: demo
  annotations:
    kubernetes.io/ingress.class: kong
spec:
  rules:
    - http:
        paths:
          - path: /tse
            backend:
              serviceName: nginx
              servicePort: 80
```

- 对于 apiVersion 为 v1 版本的 Ingress 资源，请参考如下示例进行配置：

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: demo-v1
  annotations:
    konghq.com/plugins: "httpbin-auth"
spec:
  ingressClassName: kong
  rules:
  - http:
      paths:
      - path: /demo-v1
        pathType: Prefix
        backend:
          service:
            name: nginx
            port:
              number: 80
```

7. 单击左侧**服务路由** > 服务进入服务列表，查看是否已生成对应的服务。

8. 若已生成对应的服务，单击服务的名称，进入服务详情页面，单击**服务信息**页签，查看是否有对应的节点信息。

← 路由管理 / 服务 / default.nginx-v1.pnum-80

路由管理 灰度策略 限流策略 服务信息

基本信息

服务 ID e94713bc-8a48-47f1-ac43-a8582cf924ab

服务名称 default.nginx-v1.pnum-80

创建时间 2023-07-20 20:26:21

标签 managed-by-ingress-controller

服务信息

编辑

服务类型 IP 列表

请求协议 HTTP

请求路径 /

弹性伸缩 未开启

超时时间 60000 毫秒

重试次数 5 次

负载均衡算法 轮询

慢启动 未开启

慢启动时间 0秒

节点信息

节点地址	权重	健康状况	创建时间
10.0.0.23:80	100	健康	2023-07-20 20:27:12

步骤2：停止 Kong Ingress Controller

1. 在 Kong 实例详情页，查看基本信息 > Kong Ingress Controller 卡片。

Kong Ingress Controller

关联集群

所属 VPC sh-gz-test | vpc-2eppqc2r

Ingress 版本 2.7.0

IngressClass kong

ID/名称	集群类型	集群版本	状态	所属 VPC	关联云联网	关联对等连接	操作
cls- test-tre	TKE Serverless 集群	1.24	运行中	vpc-2eppqc2r sh-gz-test	-	-	解除关联

2. 单击解除关联 操作，确认后解除集群关联。

解除集群关联

您确定要解除关联集群 test-tre (cls-cc4anjxu) 吗？

解除关联后，Ingress Controller 不再同步该集群下资源变更，相关请求可能出现异常。

确定

取消

 注意:

停止后，Kong 不再监听容器集群变化，资源变更无法同步到网关实例中，通过网关访问容器服务可能出现异常。

安全防护

使用 Web 安全防火墙

Web 安全防护概述

最近更新时间：2026-07-23 17:07:36

什么是 Web 安全防火墙？

Web 应用防火墙（Web Application Firewall，WAF）能够帮助腾讯云内及云外用户应对 Web 攻击、入侵、漏洞利用、挂马、篡改、后门、爬虫等网站及 Web 业务安全防护问题。企业通过部署腾讯云 WAF 服务，将 Web 攻击威胁压力转移到腾讯云 WAF 防护集群节点，分钟级获取腾讯 Web 业务防护能力，为网站及 Web 业务安全运营保驾护航。

WAF 支持两种类型：SaaS 型和负载均衡型，这两种类型的 WAF 都支持接入防护，对比如下：

WAF 类型	负载均衡型 WAF（推荐）	SaaS 型 WAF
是否需要网关控制台操作	需要。	不需要。
接入方式	支持 域名防护 和 对象防护 两种方式，也可同时使用。	参考 接入 SaaS 型 WAF 操作指南 ，源站地址填写一个网关公网负载均衡的公网 IP。
WAF 到网关的流量路径	旁路对接。流量先到网关，再到 WAF，旁路送审。	流量先到 WAF，再到网关。
WAF 到网关流量类型	内网。	公网。
如何选择	- 旁路对接，业务转发与安全防护分离，最大限度减少安全防护对业务的影响。 - 使用内网对接，更省流量。	- 只能绑定一个公网负载均衡的公网 IP。 - 如需使用网页防篡改和数据防泄漏功能，仅 SaaS 型 WAF 可支持。

云原生网关推荐使用负载均衡型 WAF，可将网关收到的 HTTP/HTTPS 流量镜像到 WAF 集群，WAF 进行旁路威胁检测和清洗，将用户请求的可信状态同步到云原生网关进行威胁拦截或放行，以保护您的业务安全。通过旁路对接的方式，具备如下优势：

- 无感知接入，域名接入 WAF 无需调整现有网络架构。
- 网站业务转发和安全防护分离，保障网站业务安全、稳定可靠。

负载均衡型 WAF 接入方式

负载均衡型 WAF 提供两种接入方式：

方式一：域名接入

通过添加域名的方式与网关资源进行绑定，实现对经过网关的 HTTP 或 HTTPS 流量进行检测和拦截，需要在网关侧进行：

1. 添加接入域名
2. 配置需要防护的资源

方式二：对象接入

通过 WAF 资产中心的实例管理模块，与网关实例进行关联，实现对象维度的接入，网关实例的所有流量都进行防护。

云原生对象接入支持基于云原生网关实例对象维度接入防护，将网关实例 ID 的 Web 业务流量镜像，通过协商协议转发到 WAF 集群接入防护。接入后，自动生成基于网关实例 ID 对象的防护集合，对未配置域名接入的 Web 流量启动防护，并允许客户自定义修改防护策略规则。

两种接入方式的对比如下：

对比项	域名接入	对象接入
是否需要配置域名	需要	不需要
防护粒度	资源级（支持路由级和服务级）	实例级
接入复杂度	较复杂	简单
网关与 WAF 流量方式	旁路对接	

两种相关防护生效顺序如下：

对象接入以网关实例为维度进行安全防护接入，如您需要更精细的防护，可使用域名接入，通过配置防护域名，精准实现资源防护。您也可以同时配置两种防护策略，相关防护生效顺序如下：

- 精准域名接入防护：优先级最高，命中后安全防护策略优先生效。
- 对象接入防护：没有命中任何域名防护策略后的流量，继续执行对象默认防护策略。

说明：

- 对象接入：仅企业版及以上版本支持 WAF 对象接入。
- 域名接入：不同 WAF 版本支持域名数量不同，详细参考 [WAF 套餐与版本说明](#)。

使用场景

场景一：实例全防护

场景说明：需要访问该网关实例的所有请求都进行 WAF 防护，存量路由和后续新增路由不用单独配置。适用于安全性要求极高的生产场景。

使用方式：开启对象接入防护。

场景二：仅部分服务需要防护

场景说明：仅访问某些后端服务的请求需要接入防护，其他请求不需要接入防护，适用于测试场景或安全性要求较高的生产场景。

使用方式：对需要防护的服务开启域名防护。

场景三：仅部分路由需要防护

场景说明：仅访问某些路由需要接入防护，其他请求不需要接入防护，适用于测试场景或安全性要求较低的生产场景。

使用方式：对需要防护的路由开启域名防护。

场景四：开启了服务级防护，需要关闭服务下部分路由的防护

场景说明：已开启服务级防护，新增了测试路由需要取消防护进行测试。

使用方式：对需要测试的路由关闭防护，此时服务防护将由全部开启变为部分开启状态，待测试完成后，可选择性开启路由防护，此时服务防护会再次变为全部开启。

域名防护

最近更新时间：2026-07-23 17:07:36

本文介绍云原生网关如何结合 WAF 通过添加域名的方式与云原生网关的资源（包括服务、路由）进行绑定，实现对经过云原生 API 的流量进行检测和拦截。

域名防护提供服务级、路由级别的防护，分别适用于不同粒度的防护，您可根据网关业务特性和 API 使用情况分别配置，在节约成本的同时保障 API 安全。对应的防护的区别为：

防护类型	防护说明	状态转换
服务级防护	访问该服务下所有路由的请求都将接入 WAF 默认规则的防护引擎，如您需要针对具体服务开启 WAF 防护，可使用该粒度防护。	默认关闭。 - 服务下全部路由开启：自动变为全部开启 - 服务下部分路由关闭：自动变为部分开启
路由级防护	仅访问该路由的请求都将接入 WAF 默认规则的防护引擎，如您只需要对具体路由开启 WAF 防护，可使用该粒度防护。	默认关闭。 服务防护开启：自动变为全部开启

说明：
不同 WAF 版本支持域名数量不同，详情请参见 [WAF 套餐与版本说明](#)。

前提条件

- 在开启 WAF 前，确保您的 tse-global-configuration 插件和 tse-cloud-waf 插件已升级到最新版本。
- 登录 [云原生网关控制台](#)，点击实例名称进入实例详情页
 - 在左侧导航栏单击 **插件管理**。
 - 选择系统插件页签，查看插件是否已更新到最新版本

操作步骤

步骤 1：配置 Web 应用防火墙

- 登录 [Web 应用防火墙控制台](#)，在左侧导航栏中，选择 **接入管理 > 域名接入**。
- 在域名接入页面，单击**添加域名**，填写相关配置参数，单击确定即可。

添加域名

所属实例

SaaS型

负载均衡型

域名

test.com

流量来源

CLB

云原生API网关

APISIX网关

其他应用网关

代理情况

否

是

WAF前是否有七层代理服务(高防/CDN等)?

国内地域

北京

广州

深圳金融

北京金融

重庆

上海金融

南京

上海

成都

福州

参数	说明
所属实例	负载均衡型。
域名	在域名输入框中添加需要防护的域名。
流量来源	选择云原生网关。
代理情况	选择“是”，WAF 将通过 XFF 字段获取客户真实 IP 地址作为源地址，勾选可能存在源 IP 被伪造的风险。
国内地域	选择需要防护的地域。

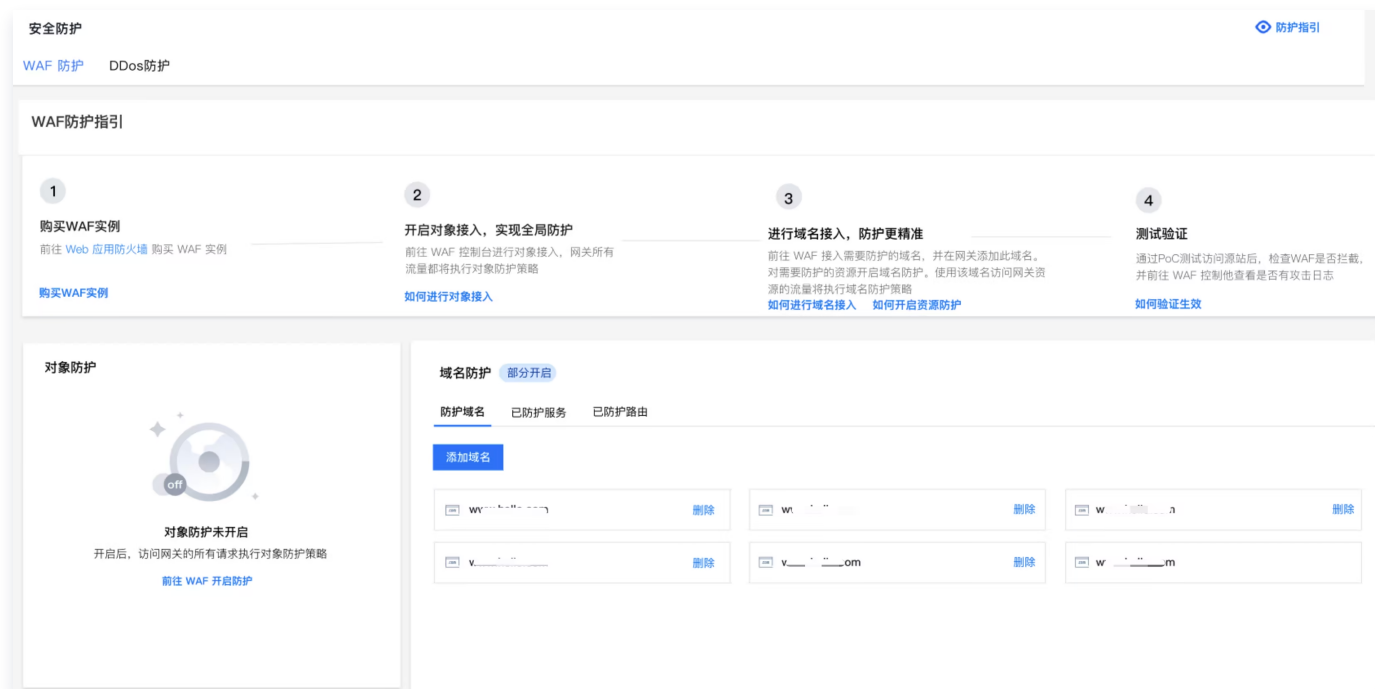
3. 单击**确定**，即可返回域名接入。在域名接入页面可以查看到防护域名和网关实例 ID、名称等信息。

步骤 2：添加防护域名

1. 登录 [云原生网关控制台](#)，点击实例名称进入实例详情页，在左侧导航栏选择**安全防护 > WAF 防护**，在防护域名 Tab，单击添加域名，选择或填写 [步骤 1](#) 中已经接入 WAF 的域名，支持选择在证书管理中已添加证书对应的域名，也支持自行输入域名进行添加。
2. 单击**确定**，确认防护域名已添加完成。

⚠ 注意：

您需要确保防护域名已完成 WAF 接入，否则将无法将对应域名的请求进行送审。



步骤 3：开启资源防护

云原生网关支持服务级和路由级粒度的防护，

开启服务级防护

1. 进入防护服务 Tab，默认展示当前实例下所有服务。
2. 选择需要进行 WAF 接入的服务，单击**开启防护**，在开启 WAF 防护的弹窗中，单击**确定**，开启该服务的 WAF 防护。

❗ 说明：

- 服务级防护开启后，该服务以及服务下所有的路由都对应开启防护。
- 如将服务下部分路由防护关闭，服务防护状态和实例防护状态都将变为部分开启。

开启路由级防护

1. 进入防护路由 Tab，点击防护路由，默认展示当前实例下所有路由。
2. 选择需要进行 WAF 接入的路由，单击**开启防护**，在开启 WAF 防护的弹窗中，单击**确定**，开启该路由的 WAF 防护。

关闭服务级防护

1. 进入防护服务 Tab，点击防护服务，默认展示当前实例下所有服务。
2. 选择需要关闭 WAF 接入的服务，单击**关闭防护**，在关闭 WAF 防护的弹窗中，单击**确定**，关闭该服务的 WAF 防护。

⚠ 注意：

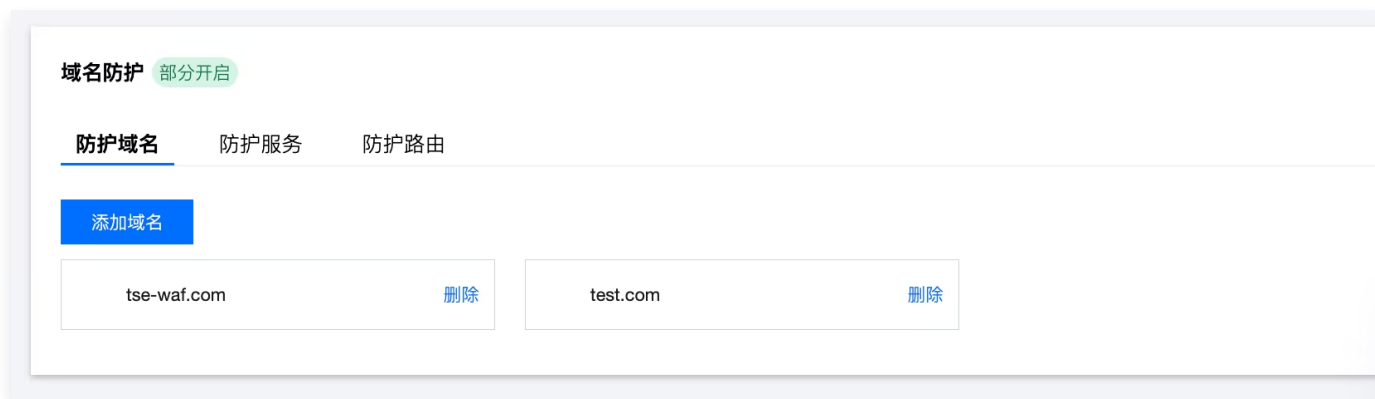
服务级防护关闭后，该服务以及服务下所有的路由都对应关闭防护。

关闭路由级防护

1. 进入防护路由 Tab，点击防护路由，默认展示当前实例下所有路由。
2. 选择需要进行 WAF 接入的路由，单击**关闭防护**，在关闭 WAF 防护的弹窗中，单击确定，关闭该路由的 WAF 防护。

步骤4: 测试验证

1. 登录 [云原生网关控制台](#)，点击实例名称进入实例详情页，在左侧导航栏中，选择安全防护。
2. 确认域名防护状态为部分开启或全部开启。



3. 在浏览器中输入网址 `http://<网关域名或IP>/?test=alert(123)` 并访问，浏览器返回阻断页面，说明 Web 应用防火墙防护功能正常。



很抱歉，您提交的请求可能对网站造成威胁，请求已被管理员设置的策略阻断

本页面为[腾讯T-Sec Web应用防火墙\(WAF\)](#)默认提示页面，如有疑问请联系网站管理员并提供UUID信息

您的请求UUID为 

对象防护

最近更新时间：2026-07-28 15:21:00

本文介绍云原生网关如何结合 WAF 通过对象防护的方式实现网关安全防护。对象防护开启后，所有访问网关的请求都默认进行防护。

❗ 说明：

仅 WAF 企业版及以上版本支持 WAF 对象接入。

前提条件

在开启 WAF 前，确保您的 tse-global-configuration 插件和 tse-cloud-waf 插件已升级到最新版本。

1. 登录 [云原生网关控制台](#)。
2. 在左侧导航栏单击云原生网关，点击实例名称进入实例详情页，单击左侧**插件管理**。
3. 选择系统插件页签，查看插件是否已更新到最新版本。

操作步骤

1. 登录 [Web 应用防火墙控制台](#)，在左侧导航栏中，选择**接入管理 > 对象接入**。
2. 在对象接入列表，选择需要进行对象接入的云原生网关实例，开启 WAF 开关。
3. 登录 [云原生网关控制台](#)，在左侧导航栏单击云原生网关，点击实例名称进入实例详情页。
4. 在左侧导航中选择**安全防护**，进入 WAF 防护页签，确认对象防护已开启。
5. 在浏览器中输入网址 `http://<网关域名或IP>/?test=alert(123)` 并访问，浏览器返回阻断页面，说明 Web 应用防火墙防护功能正常。

使用 DDoS 防护

最近更新时间：2026-07-23 17:07:34

操作背景

DDoS 防护（Anti-DDoS）具有全面、高效、专业的 DDoS 防护能力，为企业组织提供 DDoS 高防包、DDoS 高防 IP 等多种 DDoS 解决方案，应对 DDoS 攻击问题。通过充足、优质的 DDoS 防护资源，结合持续进化的“自研+AI 智能识别”清洗算法，保障用户业务的稳定、安全运行。

云原生网关公网负载均衡默认提供基础 DDoS 防护服务，如需更高级别防护，可购买 DDoS 高防产品，并将其绑定云原生网关，即可享受更安全的服务。

操作步骤

步骤 1：购买 DDoS 防护实例

网关默认为所有已添加的公网负载均衡提供最高不超过 2Gbps 的基础 DDoS 防护，如需更高级的使用 DDoS 防护，可购买 DDoS 高防包或高防 IP。DDoS 根据不同场景提供了多种防护解决方案，请参考 [DDoS 防护解决方案对比](#) 进行选择。

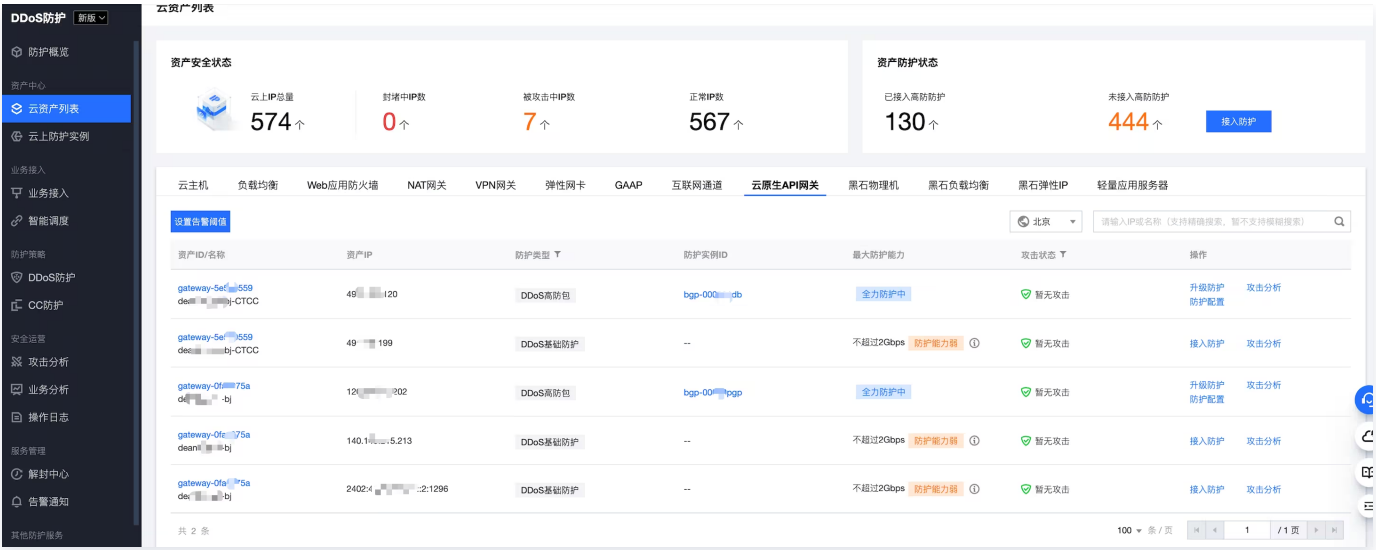
步骤 2：防护实例绑定网关资源

场景一：使用高防 IP

可参考 [DDoS 高防 IP 操作指南](#) 进行业务接入。

场景二：使用高防包

1. 进入 [DDoS 防护控制台](#)，单击云资产列表，选择云原生API网关，确认地域。
2. 在资产 IP 中，选择需要进行 DDoS 防护的公网负载均衡 IP，点击右侧的接入防护。
3. 在接入防护对话框中，选择需要绑定的 DDoS 高防包，点击确定。
4. 确认在资产列表中，防护类型已经更新为 DDoS 高防包，最大防护能力为全力防护中。



5. 在 [云原生网关控制台](#)，点击实例 ID，进入实例详情页，选择安全防护，进入 DDoS 防护页签，确认当前高防包关联资源。

步骤 3：查看/配置防护策略

可参考 [DDoS 防护配置](#) 进行防护策略查看和配置。

结合 WAF 和 DDoS实现安全防护

最近更新时间：2026-07-23 17:07:35

应用场景

云原生网关可结合 Web 应用防火墙与 DDoS 高防包联动，为用户提供全方位的安全防护。

- Web 应用防火墙提供实时防护能力，可有效拦截 Web 攻击，保障用户业务的数据和信息安全。
- DDoS 高防包可以提供上百 Gbps 的 DDoS 防护能力，轻松应对 DDoS 攻击，保障业务稳定运行。

本文介绍云原生网关如何结合 WAF 和 DDoS 高防包，实现网关安全防护。

操作步骤

说明：

步骤1和步骤2没有严格的先后次序，您可以按照您的使用习惯完成配置。

步骤1：配置 Web 应用防火墙

云原生网关支持域名接入和对象接入两种方式。对象接入以网关实例为维度进行安全防护接入，如您需要更精细的防护，可使用域名接入，通过配置防护域名，精准实现资源防护。您也可以同时配置两种防护策略，相关防护生效顺序如下：

- 精准域名接入防护：优先级最高，命中后安全防护策略优先生效。
- 对象接入防护：没有命中任何域名防护策略后的流量，继续执行对象默认防护策略。

说明：

- 对象接入：仅企业版及以上版本支持 WAF 对象接入。
- 域名接入：不同 WAF 版本支持域名数量不同，详细参见 [WAF 套餐与版本说明](#)。

对象接入

1. 登录 [Web 应用防火墙控制台](#)，在左侧导航栏中选择接入管理。



2. 在对象接入页签，选择需要进行对象接入的云原生网关实例，开启 WAF 开关。

域名接入

1. 登录 **Web 应用防火墙控制台**，在左侧导航栏中选择**接入管理**。2. 在**域名接入**页签，单击**添加域名**，填写相关参数，单击**确定**。

- 所属实例：选择云原生型。
- 域名：输入需要防护的域名。
- 流量来源：云原生API网关。
- 代理情况：选择是否进行代理，选择“是”，WAF 将通过 XFF 字段获取客户真实 IP 地址作为源地址，勾选可能存在源 IP 被伪造的风险。
- 国内地域：选择需要防护的地域。

添加域名

所属实例

SaaS型云原生型

域名

请输入域名

流量来源

☒ CLB ☐ 云原生API网关 ☐ 云函数 ☐ 云上自定义网关 ☐ 混合云网关

代理情况

☒ 否 ☐ 是

WAF前是否有七层代理服务(高防/CDN等)?

国内地域

广州

南京

上海金融

北京

重庆

上海

深圳

北京金融

成都

深圳金融

福州

手工刷新

防护对象组

暂无可用防护对象组

暂无可用防护对象组，前往【自定义防护对象组】创建

步骤2：配置 DDoS 高防包

1. 登录 **DDoS 防护控制台**，在左侧导航栏中，选择**云资产列表**。2. 在云资产列表页面，选择需要接入防护的云原生网关实例，单击操作列的**接入防护**。

DDoS防护

防护概览

资产中心

云资产列表

云上防护实例

业务接入

业务接入

智能调度

防护策略

DDoS防护

CC防护

安全运营

攻击分析

业务分析

云资产列表

资产安全状态

云上IP总量

92个

封禁中IP数

0个

被攻击中IP数

0个

正常IP数

92个

资产防护状态

已接入高防防护

0个

未接入高防防护

92个

接入防护

云主机

负载均衡

Web应用防火墙

NAT网关

VPN网关

弹性网卡

GAAP

互联网通道

云原生API网关

黑石物理机

黑石负载均衡

黑石弹性IP

轻量应用服务器

设置告警策略

广州

请输入IP或名称（支持精确搜索，暂不支持模糊搜索）

资产ID/名称	资产IP	防护类型	防护实例ID	最大防护能力	攻击状态	操作
ins-4d...ve	1.179	DDoS基础防护	--	不超过2Gbps 防护能力弱	暂无攻击	接入防护 攻击分析
ins-km4ws4zm	10...5	DDoS基础防护	--	不超过2Gbps 防护能力弱	暂无攻击	接入防护 攻击分析
ins-88rbdqe	1...242	DDoS基础防护	--	不超过2Gbps 防护能力弱	暂无攻击	接入防护 攻击分析

- 在接入防护对话框中，选择需要绑定的 DDoS 高防包，单击确定。
- 确认在资产列表中，防护类型已经更新为 DDoS 高防包，最大防护能力为全力防护中。

步骤3：配置云原生网关

说明：

如您使用对象接入，此步骤可忽略。如您使用域名接入，请参考如下步骤添加域名。

- 登录 [云原生网关控制台](#)，单击目标实例进入实例详情，在左侧导航栏中，选择**安全防护**。
- 进入 **WAF 防护**页签，在域名防护模块，选择**添加域名**，填写需要进行防护的域名。

域名防护 部分开启

防护域名

防护服务

防护路由

添加域名

.com

tse-waf.com

删除

.com

test.com

删除

- 选择需要防护的服务或路由开启防护。



步骤4：测试验证

1. 登录 [云原生网关控制台](#)，单击目标实例进入实例详情，在左侧导航中，选择安全防护。
2. 进入 DDoS 防护 Tab，确认当前网关已关联防护实例。



3. 进入 WAF 防护 Tab，确认防护状态已开启。

对象接入

在对象防护模块，确认对象防护已开启。

对象防护



对象防护已开启

访问网关的所有请求执行对象防护策略
已关联 WAF 实例：[gz-Default](#)

[查看防护配置](#) [关闭防护](#)

域名接入

确认域名防护状态为部分开启或全部开启。

域名防护 部分开启

防护域名 防护服务 防护路由

添加域名

tse-waf.com

[删除](#)

test.com

[删除](#)

4. 在浏览器中输入网址 `http://<网关域名或IP>/?test=alert(123)` 并访问，浏览器返回阻断页面，说明 Web 应用防火墙防护功能正常。



很抱歉，您提交的请求可能对网站造成威胁，请求已被管理员设置的策略阻断

本页面为[腾讯T-Sec Web应用防火墙\(WAF\)](#)默认提示页面，如有疑问请联系网站管理员并提供UUID信息

您的请求UUID为   

监控告警

查看默认监控

最近更新时间：2026-07-23 17:07:35

操作场景

云原生智能网关对运行的网关实例提供了多项监控指标，用以监测实例运行情况，例如：4xx请求数、5xx请求数等业务指标，CPU 利用率、内存使用率等系统指标。

您可以根据这些指标实时了解网关实例的运行状况，针对可能存在的风险及时处理，保障系统的稳定运行。本文为您介绍通过 TSF 控制台查看网关实例监控的操作。

查看监控入口

1. 登录 [TSF 控制台](#)，在左侧导航栏单击云原生网关。
2. 单击 ID/名称，进入实例详情页。
3. 单击数据观测页签，设置要查看的视图和时间范围，即可查看以下监控指标信息。

支持监控指标及含义

请求监控

实例/节点/路由

指标名	指标含义
总请求数	总请求数。按照所选择的时间粒度统计求和。
请求平均时延	请求平均时延。按照所选择的时间粒度统计求平均值。
请求最大时延	请求最大时延。按照所选择的时间粒度统计求最大值。
网关直接返回的请求数	网关未转发到后端，直接返回响应的请求量（如鉴权失败、触发限流时）。按照所选择的时间粒度统计求和。
网关平均时延	网关自身处理请求的平均耗时。
网关最大时延	网关自身处理请求的最大耗时。
2xx请求数	客户端发送到云原生网关，请求成功的次数（如 200 OK），按照所选择的时间粒度统计求和。
3xx请求数	客户端发送到云原生网关，请求重定向的次数，按照所选择的时间粒度统计求和。

4xx请求数	客户端发送到云原生网关的非法请求，如鉴权不通过或者超过限流值的错误个数，网关直接返回的客户端错误的个数（如 401 鉴权失败、403 权限不足、429 限流）。按照所选择的时间粒度统计求和。
5xx请求数	云原生网关将消息转发到后端服务，后端服务返回的服务端错误的个数（如 500 后端异常、502 后端无效响应、504 后端不可达）。按照所选择的时间粒度统计求和。
404请求数	请求后端服务失败，请求所希望的资源未在后端服务器上发现，此类错误的个数的统计，按照所选择的时间粒度统计求和。
429请求数	请求后端服务失败，请求被限流，此类错误的个数的统计，按照所选择的时间粒度统计求和。
499请求数	请求后端服务失败，客户端在后端响应前主动断开连接，此类错误的个数的统计，按照所选择的时间粒度统计求和。
502请求数	网关尝试执行后端请求时，从后端服务器接收到无效的响应（通常连接服务失败），此类错误的个数的统计，按照所选择的时间粒度统计求和。
504请求数	API 网关尝试执行后端请求时，后端机器不可达，此类错误的个数的统计，按照所选择的时间粒度统计求和。
网关转到后端的请求数	网关成功转发到后端服务的请求量。按照所选择的时间粒度统计求和。
后端平均时延	后端服务处理请求的平均耗时。按照所选择的时间粒度统计求平均值。
后端最大时延	后端服务处理请求的最大耗时。按照所选择的时间粒度统计求最大值。
后端2xx请求数	后端服务请求成功的次数（如 200 OK），按照所选择的时间粒度统计求和。
后端3xx请求数	后端服务请求重定向的次数，按照所选择的时间粒度统计求和。
后端4xx请求数	后端服务非法请求的次数。按照所选择的时间粒度统计求和。
后端5xx请求数	后端服务返回的服务端错误的个数（如 500 后端异常、502 后端无效响应、504 后端不可达）。按照所选择的时间粒度统计求和。
后端404请求数	后端服务资源未在后端服务器上发现，此类错误的个数的统计，按照所选择的时间粒度统计求和。
后端429请求数	后端服务请求失败，请求被限流，此类错误的个数的统计，按照所选择的时间粒度统计求和。
后端499请求数	后端服务请求失败，客户端在后端响应前主动断开连接，此类错误的个数的统计，按照所选择的时间粒度统计求和。

后端502请求数	后端服务请求失败，后端服务接收到无效的响应，此类错误的个数的统计，按照所选择的时间粒度统计求和。
后端504请求数	后端服务请求失败，后端机器不可达，此类错误的个数的统计，按照所选择的时间粒度统计求和。

系统监控

实例/节点监控指标

指标名	指标含义
CPU 使用率	云原生网关集群的 CPU 使用率，按照所选择的时间粒度统计求平均值。
内存使用率	云原生网关集群的内存使用率，按照所选择的时间粒度统计求平均值。
入口带宽流量	云原生网关集群的入口带宽流量，按照所选择的时间粒度统计求平均值。
出口带宽流量	云原生网关集群的出口带宽流量，按照所选择的时间粒度统计求平均值。
TCP 入连接数	云原生网关集群的 TCP 连接数，按照所选择的时间粒度统计求平均值。
最大内存使用率	云原生网关集群在所选时间粒度内的内存使用率最大值。用于观测内存使用峰值，判断是否存在内存突增风险（如内存泄漏、突发流量压力）。
最大 CPU 使用率	云原生网关集群在所选时间粒度内的 CPU 使用率最大值。用于发现 CPU 负载峰值波动，定位计算密集型操作（如复杂鉴权、协议转换）导致的性能突增。
运行节点数	所选时间粒度内，云原生网关集群中正常运行的节点数量。反映集群部署规模与可用节点状态，节点数异常减少可能代表集群故障或伸缩操作。
客户端到网关进程的新建连接数	所选时间粒度内，客户端与网关进程之间新建立的 TCP 连接数量。观测短时间内连接建立频率，判断客户端连接活跃度。
客户端到网关进程的活跃连接数	所选时间粒度内，客户端与网关进程之间处于活跃通信状态的 TCP 连接数量。反映网关当前承载的有效连接负载。
客户端到网关进程的非活跃连接数	所选时间粒度内，客户端与网关进程之间建立但无活跃通信的 TCP 连接数量。辅助判断连接资源闲置情况，过多可能意味着连接回收 / 管理机制需优化。
客户端到网关进程的并发连接数	所选时间粒度内，客户端与网关进程之间同时存在的 TCP 连接总数（含活跃、非活跃）。直接反映网关的连接并发压力，是评估网关连接容量的关键指标。
客户端到网关进程的入流量	所选时间粒度内，从客户端发送到网关进程的总数据量。
网关进程到客户端的出	所选时间粒度内，从网关进程发送到客户端的总数据量。

流量	
客户端到网关进程的入带宽	所选时间粒度内，客户端到网关进程的平均带宽占用量（单位时间内的流量传输速率）。评估客户端到网关的带宽压力，避免带宽瓶颈导致连接 / 传输延迟。
网关进程到客户端的出带宽	所选时间粒度内，网关进程到客户端的平均带宽占用量（单位时间内的流量传输速率）。与“入带宽”结合，分析网关对外带宽负载，防止带宽瓶颈影响响应传输。

公网负载均衡监控指标

1. 客户端到 LB 的监控

指标名	指标含义
入流量	在统计粒度内，客户端流入到负载均衡的流量。
出流量	在统计粒度内，负载均衡流出到客户端的流量。
入包量	在统计粒度内，客户端向负载均衡每秒发送的数据包数量。
出包量	在统计粒度内，负载均衡向客户端每秒发送的数据包数量。
入带宽	在统计粒度内，客户端流入到负载均衡所用的带宽。
出带宽	在统计粒度内，负载均衡流出到客户端所用的带宽。
活跃连接数	在统计粒度内，从客户端到负载均衡的活跃连接数。
非活跃连接数	在统计粒度内，从客户端到负载均衡的非活跃连接数。
并发连接数	在统计粒度内，从客户端到负载均衡的并发连接数。
新建连接数	在统计粒度内，从客户端到负载均衡的新建连接数。

2. 丢弃/利用率监控

指标名	指标含义
入带宽利用率	在统计粒度内，客户端通过外网访问负载均衡所用的带宽利用率。
出带宽利用率	在统计粒度内，负载均衡访问外网所用的带宽使用率。
并发连接数利用率	在统计粒度内的某一时刻，从客户端到负载均衡的并发连接数相比规格的并发连接数性能上限的利用率。
新建连接数利用率	在统计粒度内，从客户端到负载均衡的新建连接数相比负载均衡规格的新建连接数性能上限的利用率。

丢弃连接数	在统计粒度内，负载均衡丢弃的连接数。
丢弃入带宽	在统计粒度内，客户端通过外网访问负载均衡时丢弃的带宽。
丢弃出带宽	在统计粒度内，负载均衡访问外网时丢弃的带宽。
丢弃流入数据包	在统计粒度内，客户端通过外网访问负载均衡时丢弃的数据包。
丢弃流出数据包	在统计粒度内，负载均衡访问外网时丢弃的数据包。
丢弃 QPS	在统计粒度内，负载均衡丢弃的请求数。
QPS 利用率	在统计粒度内，负载均衡的 QPS 相比负载均衡规格的 QPS 性能上限的利用率。

3. LB 到后端的监控

指标名	指标含义
出流量	在统计粒度内，后端服务器流出到负载均衡的流量。
入带宽	在统计粒度内，负载均衡流入到后端服务器所用的带宽。
出带宽	在统计粒度内，后端服务器流出到负载均衡所用的带宽。

4. 七层协议监控

指标名	指标含义
CLB 返回的 3xx 状态码	在统计粒度内，负载均衡返回 3xx 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 4xx 状态码	在统计粒度内，负载均衡返回 4xx 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 5xx 状态码	在统计粒度内，负载均衡返回 5xx 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 404 状态码	在统计粒度内，负载均衡返回 404 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 499 状态码	在统计粒度内，负载均衡返回 499 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 502 状态码	在统计粒度内，负载均衡返回 502 状态码的个数（负载均衡和网关节点返回码之和）。

CLB 返回的 503 状态码	在统计粒度内，负载均衡返回 503 状态码的个数（负载均衡和网关节点返回码之和）。
CLB 返回的 504 状态码	在统计粒度内，负载均衡返回 504 状态码的个数（负载均衡和网关节点返回码之和）。
2xx 状态码	在统计粒度内，后端服务返回 2xx 状态码的个数。
3xx 状态码	在统计粒度内，后端服务返回 3xx 状态码的个数。
4xx 状态码	在统计粒度内，后端服务返回 4xx 状态码的个数。
5xx 状态码	在统计粒度内，后端服务返回 5xx 状态码的个数。
404 状态码	在统计粒度内，后端服务返回 404 状态码的个数。
499 状态码	在统计粒度内，后端服务返回 499 状态码的个数。
502 状态码	在统计粒度内，后端服务返回 502 状态码的个数。
503 状态码	在统计粒度内，后端服务返回 503 状态码的个数。
504 状态码	在统计粒度内，后端服务返回 504 状态码的个数。
最大请求时间	在统计粒度内，负载均衡的最大请求时间。
平均响应时间	在统计粒度内，负载均衡的平均响应时间。
最大响应时间	在统计粒度内，负载均衡的最大响应时间。
响应超时个数	在统计粒度内，负载均衡响应超时的个数。
每分钟成功请求数	在统计粒度内，负载均衡每分钟的成功请求数。
每秒请求数	在统计粒度内，负载均衡每秒钟的请求数。

5. 健康检查监控

指标名	指标含义
健康检查异常数	在统计周期内，负载均衡的健康检查异常个数

业务监控

实例

指标名	指标含义
-----	------

服务来源健康检查失败总请求数	云原生网关集群服务来源健康检查失败总请求数。
服务来源健康检查成功总请求数	云原生网关集群服务来源健康检查成功总请求数。
服务来源健康检查总请求数	云原生网关集群服务来源健康检查总请求数。

服务来源

指标名	指标含义
云原生网关注册中心探测失败请求数	云原生网关注册中心探测失败请求数
云原生网关注册中心探测请求成功数	云原生网关注册中心探测请求成功数
云原生网关注册中心探测总请求数	云原生网关注册中心探测总请求数

服务

指标名	指标含义
云原生网关服务上游节点总数	云原生网关服务上游节点总数
云原生网关服务上游健康节点总数	云原生网关服务上游健康节点总数
网关服务上游异常节点数	网关服务上游异常节点数
云原生网关服务上游异常节点百分比	云原生网关服务上游异常节点百分比
云原生网关服务上游权重非0节点总数	云原生网关服务上游权重非0的节点百分比
云原生网关服务上游权重非0健康节点总数	云原生网关服务上游权重非0的健康节点总数
云原生网关服务上游权重非0异常节点总数	云原生网关服务上游权重非0的异常节点总数
云原生网关服务上游权重非0异常节点百分比	云原生网关服务上游权重非0的异常节点百分比

查看监控入口

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关 > 实例详情页 > 数据观测，在页面上方选择好网关实例。

3. 设置要查看的视图和时间范围，即可查看以下监控指标信息。

日志

查看默认日志

最近更新时间：2026-07-23 17:07:34

操作场景

云原生智能网关默认为您提供网关实时日志服务和简单搜索能力，免费使用。

默认日志主要分为用户访问日志和网关错误日志。您可以通过查看云原生智能网关的用户访问日志了解用户的请求相关信息，便于进行数据分析、审计、业务排障或者查看，也可以查看云原生智能网关的网关错误日志，以便排查问题。

- 用户访问日志（accessLog）记录了用户的请求相关信息，可用于进行数据分析、审计、业务排障等。
- 网关错误日志（errorLog）是网关内部错误日志，用于网关排障。

本文为您介绍云原生智能网关默认日志功能的使用说明。

前提条件

已创建云原生智能网关实例，具体操作请参见 [新建网关](#)。

查看默认日志

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关 > 实例列表，选择需要查看默认日志的网关实例，单击实例 ID/名称进入实例详情页后，选择数据观测页签，单击默认日志页面。
3. 设置好您要查看的日志，页面即可展示相关日志内容。可以通过关键字查询相关日志。输入关键词查询，例如：“info”，注意日志检索区分大小写。

编辑默认日志规则

在默认日志页面，单击右上角的编辑日志规则，即可修改默认日志规则。您可以选择继续使用默认规则，也可以根据您的业务需求自定义日志规则。

编辑默认日志规则时，选择自定义规则后，支持开启和关闭 HTTP(S) 日志和 TCP/UDP 日志输出。修改默认日志规则后，投递到 CLS 的日志规则也会同步修改，请谨慎操作。

编辑默认日志规则



⚠ 修改默认日志规则后，投递到 CLS 的日志规则也会同步修改，请谨慎操作。

日志规则类型

默认规则

自定义规则

日志规则名称

请输入日志规则名称

访问日志

日志格式

JSON

单行文本

日志以 JSON 格式写入

HTTP(S)日志

一键填充HTTP(S)日志配置

HTTP(S)日志字段

字段名称	字段值	字段说明

新增字段

TCP/UDP日志

一键填充TCP/UDP日志配置

TCP/UDP日志字段

字段名称	字段值	字段说明

新增字段

错误日志

日志级别

notice

确定

取消

日志字段

下表列出了云原生智能网关支持的用户访问日志字段，您可以根据需要进行配置：

HTTP/HTTPS 日志字段

字段	说明
\$remote_addr	客户端地址。
\$status	HTTP 状态码。
\$remote_user	Basic Authentication 提供的用户名。
\$time_local	请求时间。
\$request	完整的请求行。

\$body_bytes_sent	发送给客户端的文件主体内容的大小。
\$request_method	请求方法。
\$host	请求携带 Host 请求头时为 “Host” 字段的值，未携带时为主机虚拟域名。
\$upstream_addr	后端服务的 IP 地址。
\$upstream_status	上游服务返回响应中的 HTTP 响应码。
\$upstream_response_time	上游服务响应耗时（毫秒精度），包括网关向后端服务开始建立连接、接收数据、关闭连接的时间。
\$scheme	HTTP 或 HTTPS 协议。
\$url	请求 URL。
\$request_length	请求数据大小 bytes，包含请求行、请求头、请求体。
\$bytes_sent	响应字节数。
\$http_referer	页面来源，header Referer 引用页面 URL。
\$http_user_agent	客户端代理信息。
\$request_time	请求耗时，从接收请求开始到发送完响应数据的时间，包含接收请求数据、处理请求、返回响应数据的时间。

TCP/UDP 日志字段

字段	说明
\$remote_addr	客户端 IP 地址。
\$time_local	请求时间，处理完请求打印日志的时间。
\$protocol	四层协议 TCP 或 UDP。
\$status	连接状态，可选值200, 400, 403, 500, 502, 503。
\$bytes_sent	响应字节数，网关发往客户端的字节数。

\$bytes_received	接收字节数，网关从客户端接收到的字节数，包括响应头和响应体。
\$session_time	会话持续时间（毫秒精度）。

Nginx 变量

当前不支持以下 Nginx 变量：

1. 以下具体变量：

- \$connection_time
- \$http3
- \$jwt_claim_
- \$jwt_header_
- \$jwt_payload
- \$memcached_key
- \$mqtt_preread_clientid
- \$mqtt_preread_username
- \$otel_parent_id
- \$otel_parent_sampled
- \$otel_span_id
- \$otel_trace_id
- \$proxy_protocol_tlv_
- \$proxy_protocol_tlv_aws_vpce_id
- \$proxy_protocol_tlv_azure_pel_id
- \$proxy_protocol_tlv_gcp_conn_id
- \$secure_link
- \$secure_link_expires
- \$session_log_binary_id
- \$session_log_id
- \$slice_range
- \$ssl_alpn_protocol
- \$ssl_curve
- \$upstream_queue_time

2. geo 开头的变量。

日志投递至 CLS

最近更新时间：2026-07-23 17:07:35

操作场景

如您需要日志持久化存储，用于排障、审计等场景，建议开启 CLS 日志服务，将网关日志投递到 CLS。开启前请先确认您已开通 CLS 日志服务。日志服务由 [CLS 日志服务](#) 提供，会产生费用，具体计费项查看 [费用详情](#)。

前提条件

已创建云原生网关实例，具体操作请参见 [新建云原生网关](#)。

开启日志投递

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击[云原生网关](#) > [数据观测](#)，在页面上方选择目标网关实例，单击[日志投递](#)页签。
3. 在 CLS 日志投递页面，单击[开启](#)，在弹窗中开启[投递日志](#)按钮，勾选要投递的日志项，支持自动创建和选择已有日志集和日志主题，单击[确定](#)后，即可开始投递日志。

⚠ 注意：

关闭日志投递不会删除日志集和日志主题，如需删除，请前往 CLS 控制台操作。

CLS 日志投递



- i** 1. 日志服务 CLS 独立计费，具体费用按照您的使用情况收取。计费标准请参考 [CLS计费概述](#) [🔗](#)。
2. 关闭日志投递不会删除日志集和日志主题，如需删除，请前往 [CLS控制台](#) [🔗](#) 进行操作。

投递日志 ☒

投递日志项 ☒ 访问日志 ☒ 错误日志

访问日志 ☒ 自动创建 ☐ 选择已有

错误日志 ☒ 自动创建 ☐ 选择已有

确定

取消

4. 配置完成后您可以单击[关联的日志主题](#)链接地址，自动跳转至网关投递的日志库查看日志。

编辑 CLS 日志规则

在日志投递页面，单击右上角的编辑日志规则，即可修改 CLS 日志规则。您可以选择继续使用默认规则，也可以根据您的业务需求自定义日志规则。

⚠ 注意：

投递到 CLS 的日志与默认日志使用相同的日志规则。修改 CLS 日志规则后，默认日志规则也会同步修改，请谨慎操作。

编辑 CLS 日志投递

在日志投递页面，单击右上角的**编辑日志投递**，在弹窗中即可修改日志投递规则。

CLS 日志投递

1. 日志服务 CLS 独立计费，具体费用按照您的使用情况收取。计费标准请参考 [CLS计费概述](#)。

2. 关闭日志投递不会删除日志集和日志主题，如需删除，请前往 [CLS控制台](#) 进行操作。

投递日志

投递日志项

访问日志

错误日志

访问日志

自动创建

选择已有

访问日志配置

tem-pk | aa5c6f51-c75b-401c-b2e0-c8932c0d4314

tem-pk | 1d15f0e9-91e7-4db0-9c9d-3dfe91c26e20

错误日志

自动创建

选择已有

错误日志配置

tse_logset | 88b11ca6-c9b5-4d55-bdbb-3ed94e9f0eda

gateway-02dd985c_KongErrorLog_topic_1695796380 | b0b23ac0-3444

确定

取消

版权所有：腾讯云计算（北京）有限责任公司

第119 共261页

上报日志到 Kafka

最近更新时间：2026-07-23 17:07:34

本文介绍云原生网关如何将请求日志上报到 Ckafka 或自建 Kafka 。

操作场景

提供自建的 Kafka 以及对应的 topic 或使用腾讯云 Ckafka 产品，根据下述操作，就可以将云原生网关的流量日志落盘到该 Kafka 实例上。

操作步骤

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关，选择需要操作的网关实例，进入实例详情页后，选择插件管理 > 系统插件。
3. 安装最新版本 tse-kafka-log 插件。

The screenshot displays the 'System Plugins' management interface. At the top, there are tabs for 'System Plugins', 'Native Plugins', and 'Custom Plugins'. Below these, a horizontal filter bar includes categories like 'All', 'Traffic Control', 'Request Conversion', 'No Server', 'Global Configuration', 'Route Forwarding', 'Security', and 'Observable'. The main area shows a grid of plugin cards. The 'tse-kafka-log' card, which is currently 'Not Installed' (未安装), is highlighted with a green border. It describes the function as 'Kafka log forwarding' (Kafka日志投递) and provides a link to 'Install Latest Version' (安装最新版本). Other cards shown include 'tse-breaker', 'tse-traffic-mirror', 'tse-global-configuration', 'tse-route', and 'tse-rate-limiting', each with their respective versions and descriptions.

4. 进入云原生网关实例，切换页面到 基本信息 > konga 控制台，开启公网访问地址，获取账号和密码后，进入 konga 管理控制台

启用后，您的客户端可以使用系统分配的地址和端口通过外网访问实例，生效时间大约需要 1 分钟。

- IP 白名单可以提供一定保护，请确保白名单设置的安全性。
- 初次开通公网绑定策略，存在延时，请您耐心等待。
- IP 组合中包含0.0.0.0，将允许所有外网 IP 访问，请谨慎操作

实例信息

部署架构

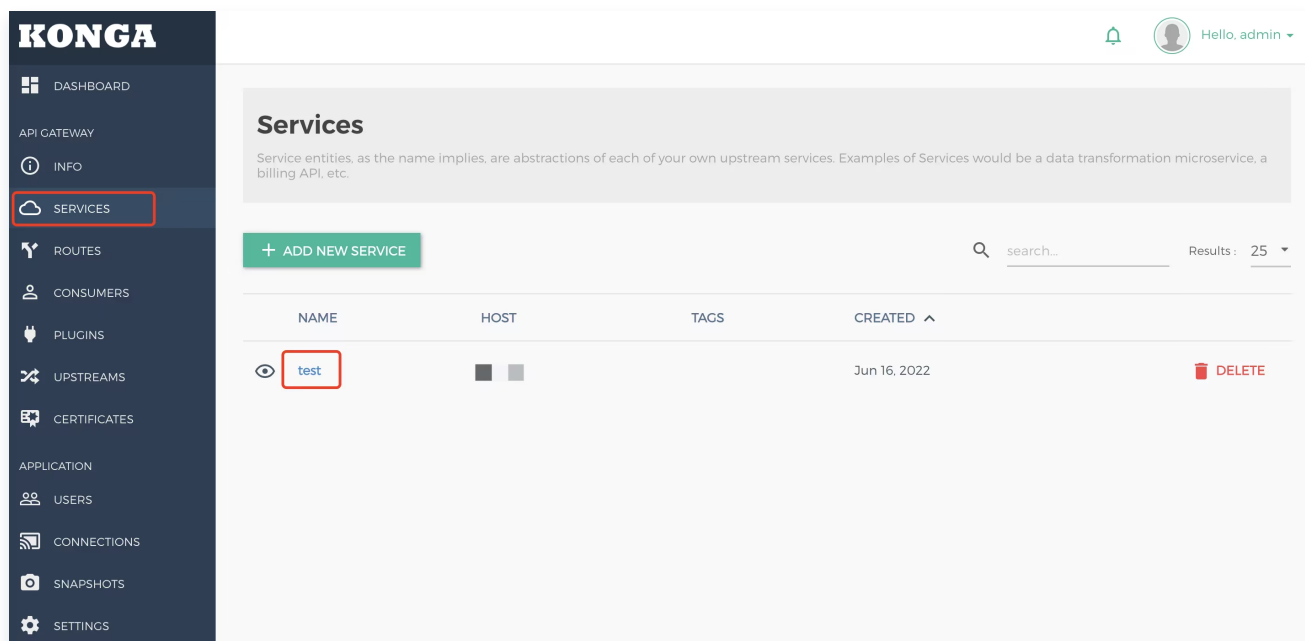
参数配置

弹性伸缩

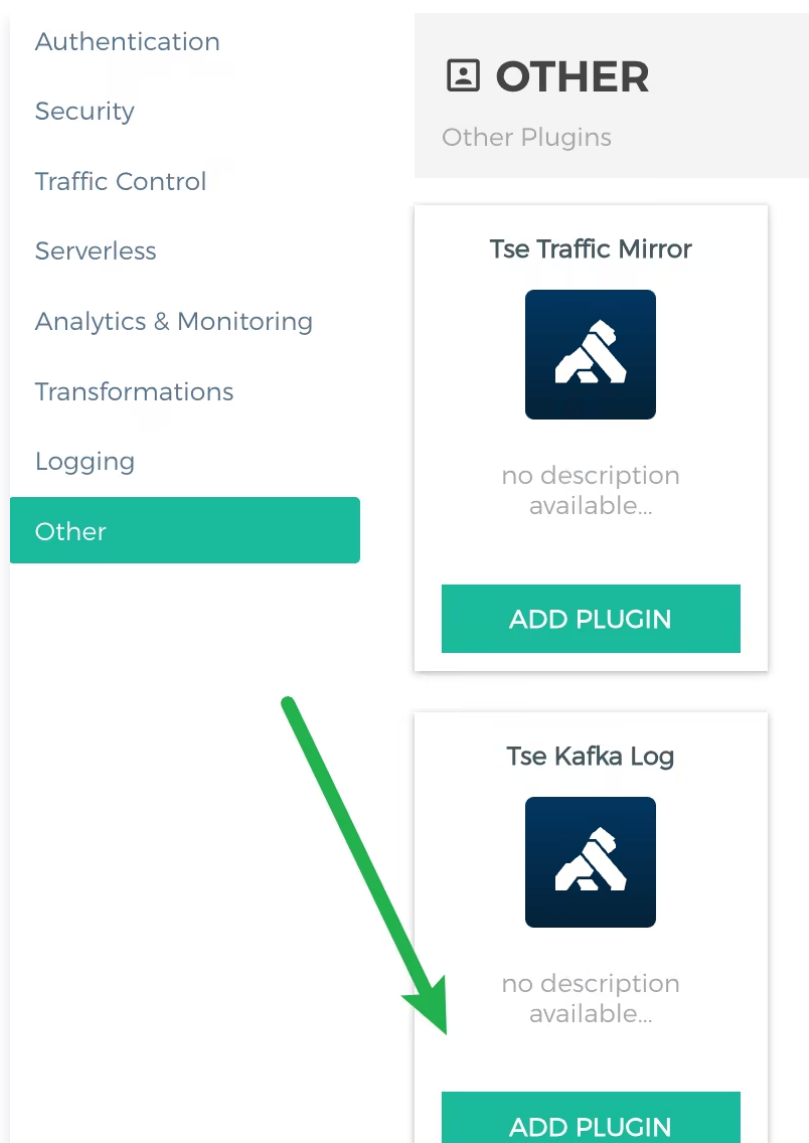
Konga 控制台公网访问地址 -konga.gz.apigw.tencentcs.com:1337/">https://gateway--konga.gz.apigw.tencentcs.com:1337/  关闭公网访问策略 白名单: () [修改](#)内网访问地址 -konga-in.gz.apigw.tencentcs.com:1337/">https://gateway-a-konga-in.gz.apigw.tencentcs.com:1337/  关闭管理员账号 admin 管理员密码 *****[显示](#) 

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

5. 进入 konga 管理控制台页面后，单击侧边导航栏 SERVICES，选择需要的 service 并进入详情页。



6. 在 TSF Kafka Log 下方点击 ADD PLUGIN 出现填写信息的弹窗



7. 填写 Kafka server 地址以及 topic，其中 bootstrap servers 填写 kafka 的 server 地址，topic 是 kafka 的 topic 名字。

⚠ 注意：

1. **topic** 是填写 topic 的名字，而不是 ID。
2. 插件配置の詳細字段说明，在 [步骤9](#) 中。

can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

bootstrap servers

Tip: Press **Enter** to accept a value.

topic

timeout 3000

max body size 8192

log req body ☐ NO

log res body ☐ NO

base64 req body encoding

Tip: Press **Enter** to accept a value.

base64 res body encoding

Tip: Press **Enter** to accept a value.

8. 该插件绑定 service 后，在已创建的服务器请求 service，消费 Kafka 后可以查看消费到当前请求的日志内容（JSON 格式）。

```
root@VM_0_12_centos ~# curl www.mytest.com -H'myheader:test'
headers begin<br/>
host : www.f[REDACTED].net.cn<br/>
x-forwarded-host : www.mytest.com<br/>
connection : keep-alive<br/>
x-forwarded-proto : http<br/>
myheader : test<br/>
x-forwarded-for : [REDACTED].195.[REDACTED].84<br/>
user-agent : curl/7.29.0<br/>
accept : /*<br/>
x-real-ip : [REDACTED].195.[REDACTED].84<br/>
x-forwarded-path : /<br/>
x-forwarded-port : 8000<br/>
GET / HTTP/1.1
nil
{"ret":0,"data":{"miniAppStatus":3},"code":"new code"}
```

```
[leonardjin@VM-203-246-centos ~/kafka/kafka_2.11-1.1.1/bin]$ sh kafka-console-consumer.sh --topic cms --bootstrap-server 11.[REDACTED]:16286 --from-beginning
one
{"request_time":"0.269","status":200,"date_time":"2022-06-09 18:23:03","method":"GET","http_host":"www.mytest.com","upstream_addr":"203.[REDACTED]:8000","upstream_response_time":"0.013","user_agent":"curl/7.29.0","machine":"kong-0","in_bytes":"93","out_bytes":"678","name":"mytest","uri":"/","remote_addr":"203.[REDACTED]","forwarded_ip":"203.[REDACTED]","query_string":""}
```

9. tse-kafka-log 插件配置字段详细说明

⚠ 注意:

在 konga 的页面中，配置项键的_会被隐藏掉，例如：bootstrap_servers 会显示为 bootstrap servers。

配置项键	是否必填	默认值	说明
bootstrap_servers	是	无	数组，一般的 bootstrap_servers 地址是9092端口，例如：10.0.90.16:9092
topic	是	无	topic 的名字，而不是 ID
max_body_size	否	8192	请求和响应的最大 body 大小，超过此大小的 body 会被截断
log_req_body	否	false	是否打印请求 body
log_res_body	否	false	是否打印响应 body
base64_req_body_encoding	否	无	数组。当请求 Header: content-encoding 符合其中一个，会把请求的 body 进行 base64 encode
base64_res_body_encoding	否	无	数组。当响应 Header: content-encoding 符合其中一个，会把响应的 body 进行 base64 encode
log_req_headers	否	true	是否打印请求 headers
log_res_headers	否	true	是否打印响应 headers
exclude_req_headers	否	无	数组。打印请求 headers 中，删除指定的 header
exclude_res_headers	否	无	数组。打印响应 headers 中，删除指定的 header
skip_expr_in_access	否	无	高级，在 access 阶段运行一个 Lua 代码，判断是否跳过打印日志。return true 就是跳过
skip_expr_in_body_filter	否	无	高级，在 body filter 阶段运行一个 Lua 代码，判断是否跳过打印日志。return true 就是跳过
http_log_mode	否	default	default/close/custom。default 为内置的日志字段，close 为关闭日志，custom 为自定义日志
http_log_custom_format	否	{}	七层自定义的日志格式。例如{"cip": "\$remote_addr", "host": "\$host"}，跟 CLS 日志

			保持一致
stream_log_mode	否	default	default/close/custom。default 为内置的日志字段，close 为关闭日志，custom 为自定义日志
stream_log_custom_format	否	{}	四层自定义的日志格式。例如{"cip": "\$remote_addr", "protocol": "\$protocol"}，跟 CLS 日志保持一致
batch_size	否	100	消息缓存队列长度，默认100
retry_count	否	3	消息发送重试次数，失败时候会重试
flush_timeout	否	2000	发送的超时时间，单位 ms
compressor	否	Lz4	消息压缩算法，默认用 Lz4

参考

更多相关说明请参见 [konga 插件文档](#)。

插件上报日志到 CLS

最近更新时间：2026-07-23 17:07:34

操作场景

本文介绍如何通过 CLS Log 插件，将包含详细请求、响应 Body 报文的日志上报到 CLS Topic 中。

⚠ 注意：

目前云原生网关已经支持 Access Log 投递到 CLS，并且支持修改 Access Log 的日志格式，可以先评估投递到 CLS 的功能是否满足需求，详细可阅读 [日志投递至 CLS](#)。再考虑是否用插件方式上报日志到 CLS，使用插件上报到 CLS，可以定制自己的日志上报格式和上报更多的字段。

前置条件

1. 已创建云原生网关实例，具体操作请参见 [新建云原生网关](#)。
2. 进入 [TSF 控制台](#)，在云原生智能网关页面，找到需要的实例。
3. 进入实例详情页后，找到[插件管理](#) > [自定义插件](#)。

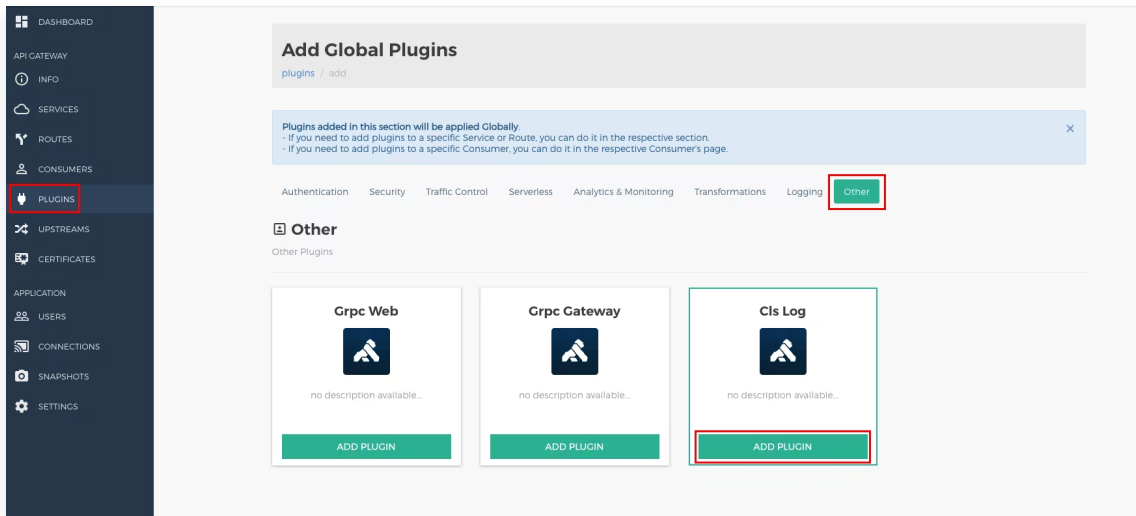
操作步骤

1. 下载 CLS Log 插件安装包 [cls_log.zip](#)。
2. 参考 [插件管理](#)，在网关实例插件配置中，上传安装 CLS Log 插件后，点击使用此版本启动插件。

The screenshot displays the 'Custom Plugins' (自定义插件) section of the Cloud Native Gateway console. On the left, a list of plugins shows 'cls-log 1.0' with a green checkmark indicating it is installed. On the right, the 'Basic Information' (基本信息) for the 'cls-log' plugin is shown, confirming its status as 'Installed' (已安装). Below this, the 'Version Management' (版本管理) table lists the current version 1.0 as 'Running' (正在运行) with a timestamp of 2023-03-21 15:02:58. The table includes columns for Version, Status, Version Description, Update Time, and Actions (Use this version, Download code, Delete).

版本	状态	版本描述	更新时间	操作
1.0	正在运行		2023-03-21 15:02:58	使用此版本 下载代码 删除

3. 登录 Konga 控制台，切换页面到[基本信息](#) > [konga 控制台](#)，开启公网访问地址，获取账号和密码后，进入 konga 管理控制台，在 PLUGINS 菜单中添加 CLS Log 插件全局开启，参数配置请参见下文中插件参数部分。



4. 通过网关 IP 发起请求，在 CLS 控制台确认日志正确上报到对应的 CLS Topic 中，日志示例如下：

```
{
  "tries": "
[{"balancer_latency\":"0, \"port\":"80, \"balancer_start\":"1654569xxxxxx,
 \"ip\":"172.0.0.2\"}],
  "upstream_uri": "/10k",
  "querystring": "{}",
  "req_url": "http://172.0.0.3/test",
  "latency_kong": "1",
  "req_method": "GET",
  "resp_headers": "{\"x-kong-upstream-latency\":"2\", \"date\":"Tue,
07 Jun 2022 02:34:15 GMT\"}",
  "req_uri": "/test",
  "resp_status": "200",
  "request_time": "3",
  "req_body": "",
  "resp_body": "some response body",
  "latency_proxy": "2",
  "req_headers": "{\"host\":"172.16.0.37\"}",
  "started_at": "1654569xxxxxx",
  "client_ip": "127.0.0.1",
  "upstream_response_time": "2",
  "resp_size": "10476",
  "route_id": "cca80823-3189-47fb-8a1f-21ebe7xxxxxx",
  "route_name": "",
  "service_id": "17caaef5-dc84-4c93-8148-e63ce1xxxxxx",
  "service_name": "",
```

```
"consumer_id": "",
"consumer_name": ""
}
```

插件参数介绍

参数	是否必填	默认值	描述
cls_host	是	—	CLS API Host，根据需要选择域名地址，例如：ap-beijing.cls.tencentyun.com。详情请参见 CLS doc
cls_topic	是	—	日志上报的目标 CLS Topic
secret_id	是	—	腾讯云账户 SecretID
secret_key	是	—	腾讯云账户 SecretKey
log_req_body	是	false	是否记录请求 Body 日志字段，开启后对网关性能有一定影响
log_resp_body	是	false	是否记录响应 Body 日志字段，开启后对网关性能有一定影响
max_req_body_size	是	10240	最大请求 Body 日志长度（单位字节），超出会被截断
max_resp_body_size	是	10240	最大响应 Body 日志长度（单位字节），超出会被截断
batch_size	是	100	单次上报日志数量，一批日志大小不建议超过 8 MB。
retry_count	是	3	最大重试次数。
mode	否	memory	memory: 通过 CLS SDK 从内存队列中异步上报日志。 access_log: 通过 access_log 模块，配合 loglistener 采集上报，受磁盘性能影响，如需使用请工单联系开通。
fallback_to_access_log	否	false	当使用 memory 模式时，上报队列堆积时是否降级到 access_log 上报模式，如需使用请工单联系开通。

注意事项

1. CLS Log 插件中配置的密钥对需拥有 `cls:UploadLog` 策略权限，详情请参见 [CLS 自定义策略示例](#) 中使用 API 上传数据部分。
2. 开启 CLS Log 日志插件会对网关性能有少量影响，影响程度和报文日志大小成正比。
3. CLS Log 插件中实现了缓冲队列、分批上报，失败重试等功能，能够在偶发性上报失败时进行重试，但由于缓冲队列大小有限，长时间上报链路故障时仍然有丢失日志的风险。
4. CLS Log 插件包含通用的基本日志字段，如需自定义日志字段，可以基于该插件二次开发。

查看日志大盘

最近更新时间：2026-07-23 17:07:34

操作场景

云原生网关提供开箱即用的 CLS 日志大盘，如您开启了 CLS 日志投递，即可使用日志分析仪表盘对投递到 CLS 的访问日志和错误日志进行快速分析。（仅专业版支持）

日志大盘支持的场景包括：

- 状态码分析
- 请求量分析
- 请求耗时分析
- 错误日志分析
- 用户行为分析
- 业务分析

本文为您介绍云原生网关日志大盘功能的使用说明。

前提条件

1. 已创建云原生网关实例，具体操作请参见 [新建云原生网关](#)。
2. 已开通 CLS 日志服务。
3. 已开启 CLS 日志投递，具体操作请参见 [日志投递至 CLS](#)。
4. 日志格式为 JSON 格式，如当前格式为单行文本，请修改日志格式为 JSON 格式，具体操作请参见 [编辑 CLS 日志规则](#)。
5. 由于大盘依赖部分日志字段进行展示，请将以下字段添加至日志规则：

字段名	字段值
status	\$status
ups_status	\$upstream_status
client	\$remote_addr
req_time	\$request_time
ups_rsp_time	\$upstream_response_time
referer	\$http_referer
method	\$request_method

host	\$host
uri	\$uri
ups_ip	\$upstream_addr

操作步骤

1. 登录 [微服务平台控制台](#)。
2. 在左侧导航栏单击云原生网关，选择需要操作的网关实例，进入实例详情页后，选择数据观测页签，单击日志大盘页签。

支持指标及含义

状态码分析

指标名	指标含义
客户端状态码分布	网关返回到客户端的状态码分布。
2xx次数	客户端收到的状态码为 2xx 的请求次数。
5xx次数	客户端收到的状态码为 5xx 的请求次数。
请求失败率趋势	网关请求失败率的趋势。
2xx占比	客户端收到的状态码为 2xx 的请求占比。
5xx占比	客户端收到的状态码为 5xx 的请求占比。
后端服务状态码分布	后端服务返回网关的状态码分布。
后端服务失败数	后端服务失败（4xx和5xx）的请求次数。
后端服务失败率趋势	后端服务收到失败请求（4xx和5xx）次数的趋势。

请求量分析

指标名	指标含义
总请求数	总请求数，按照所选择的时间粒度统计求和。
请求次数趋势	请求次数趋势。
请求地域分布	客户端 IP 所在地域分布。

请求耗时分析

指标名	指标含义
请求总耗时趋势	请求总耗时趋势。
后端响应耗时趋势	后端服务响应的耗时趋势。

错误日志分析

指标名	指标含义
Error 日志数	在所选择的时间范围内出现的错误日志总数。
Error 日志趋势	在所选择的时间范围内出现的错误日志总数趋势。
Error 日志	在所选择的时间范围内出现的错误日志详情。

用户行为分析

指标名	指标含义
HTTP Referer 分布	HTTP Referer 分布。
请求方法分布	请求方法分布。
Host 分布	Host 分布。

业务分析

指标名	指标含义
TOP10 调用量 API	在所选择的时间范围内调用量前 10 的 API。
TOP10 失败率 API	在所选择的时间范围内失败率前 10 的 API。
TOP10 调用量服务	在所选择的时间范围内调用量前 10 的后端服务。
TOP10 失败率服务	在所选择的时间范围内失败率前 10 的后端服务。
TOP10 请求 Host	在所选择的时间范围内数量前 10 的请求 Host。

Terraform 管理

最近更新时间：2026-07-23 17:07:32

Terraform 是一个 IT 基础架构自动化编排工具，它的口号是 “Write, Plan, and Create Infrastructure as Code”，是一个 “基础设施即代码” 工具，通过 Terraform 您可以创建、更新和版本控制网关中的资源。了解更多关于 Terraform，请参见 [Terraform 指南](#)。

操作前提

需要已安装并配置好 Terraform、需要已获取腾讯云 API 密钥并配置 Provider、了解 Terraform 基本语法。

通过 Terraform 管理 TSF 云原生网关资源

资源类型	说明
tencentcloud_tse_cngw_canary_rule	云原生网关灰度规则。
tencentcloud_tse_cngw_certificate	云原生网关证书。
tencentcloud_tse_cngw_gateway	云原生网关实例。
tencentcloud_tse_cngw_group	云原生网关实例分组。
tencentcloud_tse_cngw_route	云原生网关路由。
tencentcloud_tse_cngw_route_rate_limit	云原生网关路由级限流。
tencentcloud_tse_cngw_service	云原生网关服务。
tencentcloud_tse_cngw_service_rate_limit	云原生网关服务级限流。
tencentcloud_tse_waf_domains	云原生网关域名。
tencentcloud_tse_waf_protection	云原生网关 WAF 防护。

参数配置

最近更新时间：2026-07-23 17:07:32

操作场景

创建云原生网关实例时，实例中的参数例如 `nginx_http_client_header_buffer_size`、`nginx_http_client_max_body_size` 等采用默认配置，如果您的业务有特殊要求可以在 TSF 控制台修改对应的参数，优化网关性能。

前提条件

已创建云原生网关实例。

操作步骤

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关 > **实例列表**，单击目标实例的“ID”，进入网关实例基本信息页面。
3. 在**基础信息**页面，选择**参数配置**，进入参数配置页面。左侧以 JSON 格式展示所有支持修改的系统参数，右侧展示参数说明列表，您可通过参数名搜索，查看该参数支持的默认值和可选值。
4. 单击左上角的**修改参数**，编辑 JSON 内容，修改参数值为合适的内容。单击**保存**，实例将进行 kong reload 操作，完成参数修改，修改后立即生效。

⚠ 注意：

- 系统参数值均为字符串，修改时需带英文双引号（" "）。
- 默认展示实例初始参数值，当前实例该参数未设置则展示为 null。
- Access Log 日志格式的修改，在控制台的 **数据观测 > 默认日志 > 编辑日志规则**中修改，请参见 [查看默认日志](#)。
- 参数修改会触发 reload 操作，该操作可能会导致长连接中断，请谨慎操作。

插件管理

最近更新时间：2026-07-23 17:07:31

操作场景

云原生网关提供插件管理功能，您可以在控制台上管理您的系统插件、原生插件、部署自定义插件等。本文介绍相关的插件定义和开发自定义插件并部署到云原生网关的操作步骤。

插件介绍

系统插件		
系统插件是指云原生网关在开源基础上提供的增强插件。		
名称	类型	说明
tse-breaker	流量控制	Kong 熔断插件 。
tse-traffic-mirror	流量控制	流量镜像插件 。
tse-rate-limiting	流量控制	控制台服务与路由的限流策略能力， 限流控制 。
tse-proxy-rewrite	请求转换	重写插件， 鉴权配置 。
tse-custom-auth	安全	自定义鉴权的能力， 自定义鉴权 。
tse-cloud-waf	安全	控制台安全防护能力， 控制台安全鉴权 。
tse-kafka-log	可观测	上报日志到 Kafka， 查看插件说明 。
tse-prometheus	可观测	控制台 Prometheus 监控能力。
tse-trace	可观测	控制台链路追踪能力， 链路追踪 。
tse-global-configuration	全局配置	控制台 全局配置 能力。
tse-scf-web	无服务器	控制台转发云函数类型服务能力， 服务转发能力 。

tse-scf-event	无服务器	
tse-route	路由转发	建议使用控制台服务与路由的灰度策略能力， 灰度发布 。

原生插件

原生插件指开源 Kong 网关自带的插件。

名称	类型	说明
acl	安全	通过使用任意 ACL 组名称将消费者列入白名单或将其列入黑名单来限制对 Service 或者 Route 的访问。此插件需要在服务或路由上启用身份验证插件。
acme	安全	允许 Kong 使用来自 Let' s Encrypt 或者其他 ACMEv2 服务的证书，并且支持自动续期。
aws-lambda	无服务器	从 Kong 调用 AWS Lambda 函数。它可以与其他请求插件结合使用以保护，管理或扩展功能。
azure-functions	无服务器	从 Kong 调用 Azure 上的无服务器函数。它可以与其他请求插件结合使用以保护，管理或扩展功能。
basic-auth	认证鉴权	使用用户名和密码保护将基础认证添加到 Service 或 Route 。
bot-detection	安全	保护服务或路由免受大多数常见机器人的攻击，并具有将自定义客户端列入白名单和列入黑名单的能力。
correlation-id	请求转换	使用通过 HTTP header 传输的唯一 ID 关联请求和响应。
cors	安全	通过启用此插件，轻松将跨源资源共享（CORS）添加到 Service, Route
datadog	分析和监控	记录 Service 的指标，并路由到本地 Datadog 代理。
file-log	日志	将请求和响应数据写入磁盘上的日志文件，不建议在生产中使用此插件。
grpc-gateway	其他	通过 HTTP REST 接口公开 gRPC 服务。它以 JSON 格式转换请求和响应，允许通过普通 HTTP 请求访问上游 gRPC 服务。
grpc-web	其他	通过 gRPC-web 协议访问 gRPC 服务。

hmac-auth	认证鉴权	将 HMAC 签名身份验证添加到服务或路由以确定传入请求的完整性。该插件将验证在 Proxy-Authorization 或 Authorization header 中发送的数字签名（按此顺序）。
http-log	日志	将请求和响应数据转发到 HTTP 服务器。
ip-restriction	安全	通过将 IP 地址列入白名单或列入黑名单来限制对 Service 或 Route 的访问，支持配置单个 IP，多个 IP 或 CIDR 范围，如 10.10.10.0/24。
jwt	认证鉴权	验证包含 HS256 或 RS256 签名 JSON Web 令牌请求。在请求中如果验证了令牌的签名，Kong 会将请求代理到您的上游服务，否则将丢弃该请求。
key-auth	认证鉴权	将密钥身份验证（有时也称为 API 密钥）添加到 Service 或 Route。
ldap-auth	认证鉴权	使用用户名和密码保护将 LDAP 绑定身份验证添加到路由。该插件将检查 Proxy-Authorization 和 Authorization header 中的有效凭据（按此顺序）。
loggly	日志	通过 UDP 将请求和响应数据记录到 Loggly。
oauth2	认证鉴权	使用授权码模式（Authorization Code Grant），客户端凭证模式（Client Credentials），简化授权模式（Implicit Grant）或密码模式（Resource Owner Password Credentials Grant）授权添加 OAuth 2.0 身份验证层。
post-function	无服务器	在访问阶段运行其他插件之后运行一段 Lua 代码。
pre-function	无服务器	在访问阶段运行其他插件之前运行一段 Lua 代码。
prometheus	分析和监控	以 Prometheus exposition 格式公开与 Kong 和代理上游服务相关的指标，Prometheus 服务器可以对这些指标进行抓取。
proxy-cache	流量控制	基于可配置的响应代码和内容类型以及请求方法来缓存响应实体。它可以缓存每个 Consumer 或每个 API。
request-size-limiting	流量控制	限制 body 大于特定大小（以兆字节为单位）的传入请求。
request-termination	流量控制	使用指定的状态代码和消息终止传入请求。

request-transformer	请求转换	在将请求转发给上游服务之前，转换客户端发送的请求。
response-ratelimiting	流量控制	根据上游服务返回的自定义响应头限制可以发出的请求数。
response-transformer	请求转换	在将响应返回给客户端之前，转换上游服务发送的响应。
session	认证鉴权	管理通过 Kong 代理的 API 的浏览器会话，它为会话数据存储、加密、续订、到期和浏览器 Cookie 提供配置和管理。
statsd	日志	为 Service Route 记录日志指标到一个 StatsD 服务器。
syslog	日志	将请求和响应数据转发到 syslog 服务器。
tcp-log	日志	将请求和响应数据转发到 TCP 服务器。
udp-log	日志	将请求和响应数据转发到 UDP 服务器。
zipkin	分析和监控	传播 Zipkin 分布式跟踪 span，并向 Zipkin 服务器报告 span。

自定义插件

云原生网关 Kong 提供自定义插件能力扩展，您可按照 Kong 规定的格式，开发自己的定制插件，以实现数据流的修改和深层次的集成。详细操作请参见使用自定义插件。

准备自定义插件代码

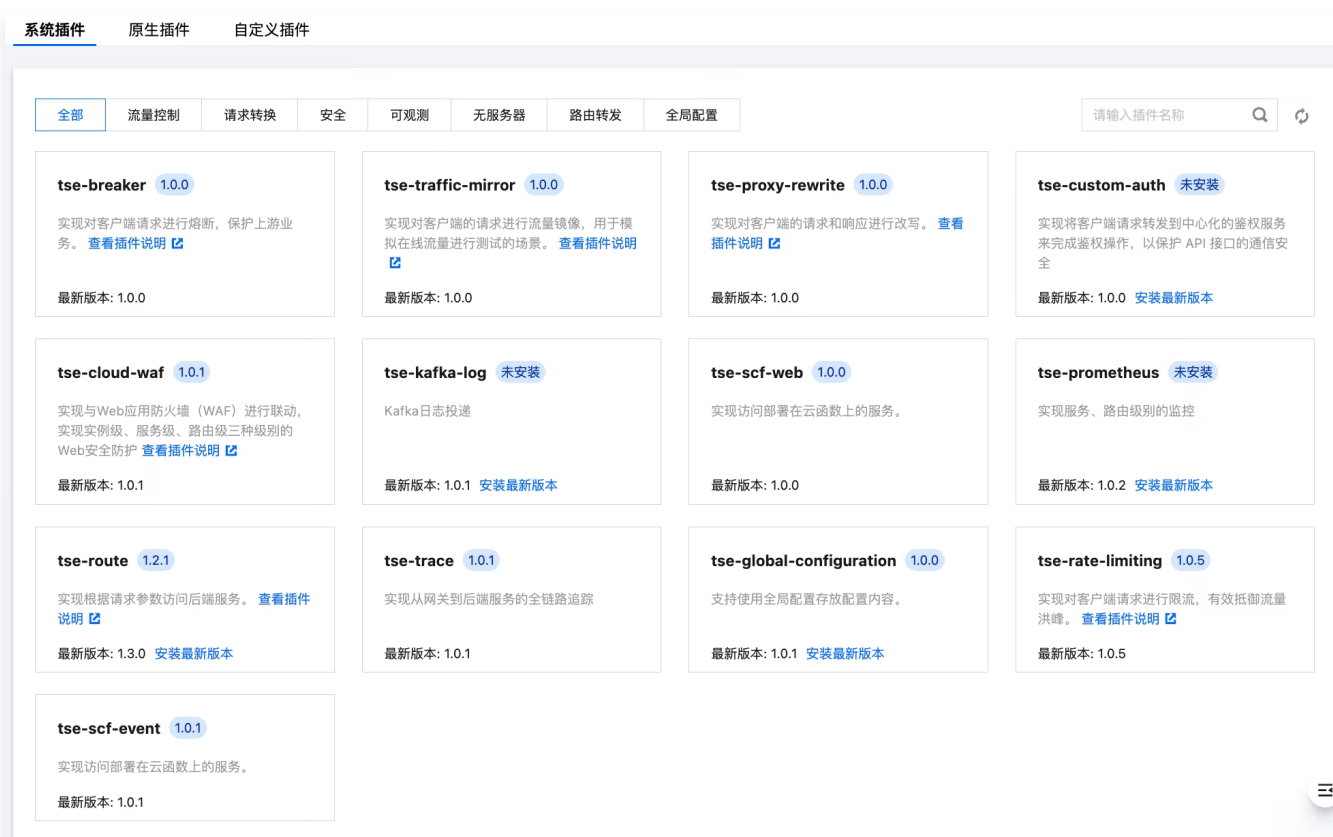
根据 kong lua 插件开发规范：[Plugin Development – File Structure – v2.7.x | Kong Docs](#)，自定义插件在保留 Kong 原生插件开发规范的情况下，同时还支持了 so 库的能力，建议 so 库在 CentOS7 的环境下编译。

插件目录结构可以参照如下：


```
[root@d7a0776b00a9 plugins]# tree demo
demo ── 插件名称
|-- handler.lua ── handler.lua 必须存在
|-- lua-lib
|   |-- demo.so ── 如插件依赖 so 库, 则 so 库放在 lua-lib 目录下
|   |-- schema.lua ── schema.lua 必须存在
1 directory, 3 files
[root@d7a0776b00a9 plugins]#
```

进入插件管理入口

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏单击云原生网关，选择需要操作的网关实例，进入实例详情页后，选择 **插件管理** 页签，可以查看系统插件、原生插件和自定义插件信息。



部署自定义插件

1. 参考 [插件介绍-自定义插件](#) 准备好插件代码。
2. 在插件管理页面，选择自定义插件页签，单击上传插件。

上传插件

上传 zip 包 *

cls-log.zip

删除

暂不支持上传包含 migrations 目录的代码包

版本号 *

1.0

支持数字、点“.”，格式：x.y.z，如：0.2.0

版本描述

请输入版本描述

确定

取消

3. 选择插件后，填写好插件版本号和版本描述信息，点击开始上传并完成上传后，单击**确定**，完成上传。

4. 上传完插件后，单击 **使用此版本**，插件会自动部署在所有的 Kong 节点中。

系统插件原生插件自定义插件

上传插件

请输入关键字

共 1 条

cls-log

未安装

20 条 / 页1 / 1 页

cls-log

卸载

基本信息

插件名称cls-log

插件状态未安装

当前运行版本-

版本管理

版本	状态	版本描述	更新时间	操作
1.0	未安装		2023-03-20 19:49:42	使用此版本 下载代码 删除

共 1 条20 条 / 页1 / 1 页

5. 当您不再需要该插件时，可以单击插件右上角的**卸载**，卸载该插件。

弹性伸缩

最近更新时间：2026-07-23 17:07:32

操作场景

云原生网关能够根据预先设定的弹性伸缩规则自动增加或减少节点数量，即扩容或缩容，从而提高资源利用率、降低资源成本。弹性伸缩支持指标规则和定时规则两种类型的弹性伸缩策略。（仅专业版支持）

- 指标规则：适用于突发流量和典型周期性流量的应用场景。由指标名称、阈值描述弹性扩缩容的触发条件、节点数量变化。
- 定时规则：适用于资源使用率有周期性规律的应用场景。达到设定的周期时间点后，TSF 云原生网关将依据设定的策略自动调整实例节点数量。

前提条件

已创建云原生网关实例。

操作步骤

步骤1：创建弹性伸缩策略

1. 前往 [云原生智能网关控制台](#)，单击需要配置的实例名称，进入实例详情页。
2. 进入实例详情页后，在基础信息页选择弹性伸缩页签，单击新建策略。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

新建策略

策略ID/名称	策略类型	描述	创建时间	更新时间	操作
strategy-cb88759d	仅定时伸缩	-	2024-10-14 15:14:31	2025-01-08 02:16:30	绑定分组 编辑 删除

共 1 条

10 条 / 页

1

1 / 1 页

3. 在新建弹性伸缩弹窗，填写策略信息：

仅指标伸缩

- 节点数范围：弹性伸缩的节点数范围，最少为分组当前节点数，最大为100个。
- 指标伸缩：满足一个条件即触发弹性伸缩。
 - 指标名称：弹性伸缩触发条件，支持为监控指标（CPU 使用率、内存使用率、QPS）配置阈值。
 - 阈值：触发弹性伸缩指标到达的阈值。
- 策略名称：输入弹性伸缩策略的名称。

- 描述：输入弹性伸缩策略的描述信息。

仅定时伸缩

- 节点数范围：弹性伸缩的节点数范围，最少为分组当前节点数，最大为20个。
- 定时伸缩：如果多个规则触发时间相同，将按序执行，您可拖动调整顺序；若触发时间早于当前时间，则从下一个时间周期开始生效。
 - 周期：执行定时弹性伸缩策略的周期。
 - 节点数：弹性扩缩容生效后的节点数量。
 - 触发时间：弹性伸缩具体触发的时间点。
- 策略名称：输入弹性伸缩策略的名称。
- 描述：输入弹性伸缩策略的描述信息。

指标伸缩与定时伸缩

- 节点数范围：弹性伸缩的节点数范围，最少为分组当前节点数，最大为20个。
- 定时伸缩：如果多个规则触发时间相同，将按序执行，您可拖动调整顺序；若触发时间早于当前时间，则从下一个时间周期开始生效。
 - 周期：执行定时弹性伸缩策略的周期。
 - 节点数：弹性扩缩容生效后的节点数量。
 - 触发时间：弹性伸缩具体触发的时间点。
- 指标伸缩：满足一个条件即触发弹性伸缩。
 - 指标名称：弹性伸缩触发条件，支持为监控指标（CPU 使用率、内存使用率）配置阈值。
 - 阈值：触发弹性伸缩指标到达的阈值，
- 策略名称：输入弹性伸缩策略的名称。
- 描述：输入弹性伸缩策略的描述信息。

4. 单击**确定**，弹窗提示需绑定集群分组后才能生效，您可直接前往绑定分组，或稍后绑定。

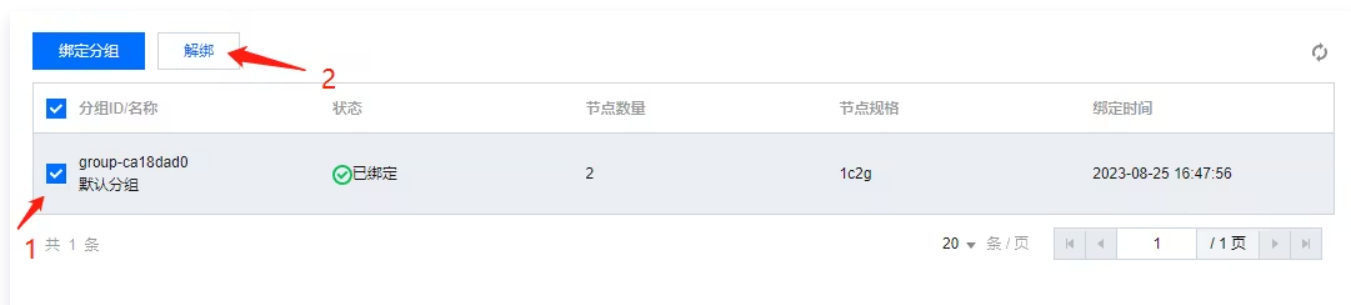
步骤2：绑定集群分组并生效

弹性伸缩策略需绑定集群分组后才能生效，支持一次绑定一个或多个集群分组。

1. 在弹性伸缩策略列表页，单击目标策略右侧**绑定分组**。
2. 在绑定集群分组弹窗中，选择一个或多个集群分组，单击**确定**，绑定后策略立即生效。

步骤3：查看策略已绑定的集群分组

1. 在弹性伸缩策略列表页，单击目标策略，进入策略详情页。
2. 单击已绑定分组，查看该策略已绑定的集群分组，您可在此绑定新的分组或解绑已有分组。



步骤4：编辑弹性伸缩策略

1. 在弹性伸缩策略列表页，单击目标策略右侧**编辑**。
2. 在 **编辑弹性伸缩策略** 页面，修改策略触发条件、弹性伸缩范围、策略名称或描述。
3. 单击 **确定**，完成编辑。

步骤5：删除弹性伸缩策略

1. 在弹性伸缩策略列表页，单击目标策略右侧**删除**。
2. 在 **删除策略** 弹窗，二次确认是否要删除该策略。
3. 单击**确定**，删除策略。

⚠ 注意：

删除后，该策略立即失效，自动与之前绑定的分组解绑，不再进行自动扩缩容，请谨慎操作。

同时配置指标伸缩与定时伸缩

定时伸缩通过设置定时的方式触发网关节点的弹性伸缩，常用在周期性任务中。指标伸缩通过配置关键指标的阈值进行节点弹性伸缩，常用于防止突发的流量冲击等状况。在实际使用中，可结合两种方式进行配置，当云原生网关感知到两种策略同时配置时，会结合两种策略内容进行节点数量调整。

当两种策略同时生效时，定时伸缩并不直接进行节点数调整，而是通过调整指标伸缩节点数范围的方式来进行伸缩判断与调整，调整原则为：

- 当前节点数 = 定时伸缩目标节点数，不进行调整。
- 当前节点数 < 定时伸缩目标节点数：
 - 定时伸缩目标节点数 > 指标伸缩最小节点数，调整指标伸缩最小节点数为定时弹性伸缩目标节点数。
 - 定时伸缩目标节点数 < 指标伸缩最大节点数，指标伸缩最大节点数不调整。
- 当前节点数 > 定时伸缩目标节点数：
 - 定时伸缩目标节点数 > 指标伸缩最小节点数，指标伸缩最小节点数不调整。
 - 定时伸缩目标节点数 < 指标伸缩最大节点数，指标伸缩最大节点数不调整。

节点数会根据指标伸缩节点数范围进行相应的调整，以实现节点弹性伸缩。

下表列出了当两种策略同时生效时，不同场景下节点扩缩容结果：

指标伸缩节点数 (调整前)	定时伸缩目标 节点数	当前节点 数	指标伸缩节点数 (调整 后)	节点数 (调整后)
最小： 1 最大： 10	5	5	最小： 1 最大： 10	5
最小： 1 最大： 10	7	5	最小： 7 最大： 10	7
最小： 1 最大： 10	3	5	最小： 1 最大： 10	5

事件中心

最近更新时间：2026-07-23 17:07:31

事件中心将云原生网关生成的事件信息进行统一管理、存储、分析和展示，您可以在事件中心查看事件详情，并为事件配置告警通知，以便及时发现问题并进行处理。

云原生网关当前支持的事件有：

事件类型	事件名称	事件详情
弹性伸缩	弹性扩容	触发的策略类型、策略 ID（名称）、变更节点 IP
弹性伸缩	弹性缩容	触发的策略类型、策略 ID（名称）、变更节点 IP
弹性伸缩	绑定弹性伸缩策略	触发的策略类型、策略 ID（名称）
弹性伸缩	解绑弹性伸缩策略	触发的策略类型、策略 ID（名称）
实例变更	修改访问策略	变更前：黑名单/白名单（变更 IP），变更后：黑名单/白名单（变更 IP）
实例维护	网关版本升级	网关版本升级，原始版本（版本号）、升级目标版本（版本号）
实例维护	撤销网关版本升级	撤销网关版本升级，原始版本（版本号）、升级目标版本（版本号）

查看事件详情

1. 登录 [微服务平台控制台](#) 。
2. 在左侧导航栏单击云原生网关 > 实例列表，找到目标网关，单击网关的“ID”，进入实例信息页面。
3. 进入实例详情页后，选择数据观测页签。
4. 在可观测性页面，进入事件中心 Tab。
5. 在页面上方设置时间范围（支持近 7 天、近 30 天和自定义时间范围）。

字段	描述
事件类型	支持弹性伸缩。
事件名称	事件信息的概要描述，如弹性扩容、弹性缩容、绑定弹性伸缩策略、解绑弹性伸缩策略。
状态	该事件是否执行成功，支持成功、失败两种。
影响对象	弹性伸缩事件类型：分组 ID。

事件详情	事件明细信息，例如事件摘要、涉及的对象和事件内容等。
触发时间	事件触发时间，精确到秒。

配置事件告警规则

- 在事件中心页面，单击左上角的前往事件总线配置告警按钮跳转至事件集页面。
- 在事件集页面，单击新建事件集，配置事件集信息。
 - 事件集名称：填写事件集名称。
 - 事件集描述：选填。
 - 追踪方式：
 - 投递配置：可选择自定义或默认投递。
 - 标签：根据需要启用并配置。
 - 事件存储：启用事件存储开关，配置事件存储天数（默认 1 天），事件存储功能当前处于公测期，用户可免费使用。后续收费计划和计费策略将另行通知。
- 事件集创建完成后，进入事件集详情页，单击基本信息 Tab，单击管理事件规则，进入事件规则页面，点击新建，配置事件模式信息。
 - 基础信息
 - 规则名称：填写事件规则名称，规则名称一旦创建无法修改。
 - 事件示例
 - 连接器类型：勾选您要配置告警规则的类型。
 - 事件实例选择：选择对应的事件实例。
 - 事件匹配
 - 事件模式：可选择表单模式和自定义模型。
 - 连接器类型：勾选您要配置告警规则的类型。
- 单击下一步，填写事件目标信息。
 - 触发方式：选择消息推送。
 - 消息模板：根据事件类型，提供两种不同类型推送模板，您可以根据实际情况选择，对于告警事件，建议统一选择通用通知模板。
 - 用户通知：根据您的需要自定义选择通知方式和告警接收对象。
- 单击完成，完成事件告警规则配置。

故障演练

最近更新时间：2026-07-23 17:07:32

背景

云原生网关是一个常用的 API 网关，用于管理和保护后端服务。为保证您的业务能够持续提供服务，云原生网关（Kong）分组（标准版、专业版）提供节点同城双可用区部署，可以在某些特殊场景下导致的可用区故障或节点临时宕机等非预期情况时，保护您的应用程序不受影响。为此混沌演练平台提供了针对云原生网关（Kong）分组节点重启的故障演练动作，您可使用混沌演练平台进行该故障演练，模拟可用区故障或节点临时宕机离线的故障场景，以此验证业务系统韧性，及时规避隐患，从而保证业务能够持续稳定地提供服务。

注意事项

- 使用故障演练前，需已购买智能顾问-混沌演练服务。
- 该故障动作支持按照默认分组以及其他分组（需指定分组 ID）进行故障注入。当选择其他分组指定分组时，如果需要对多个网关实例注入故障，建议拆分为一个动作组一个网关实例执行。
- 如果待注入的分组中的节点数量小于2，则会拒绝注入节点重启故障。
- 故障执行会从分组下的所有节点中随机选择一个节点进行重启，重启完成时间预计3~5分钟，在故障期间，可能会出现连接到重启节点的长连接断开，请业务做好重试连接机制。
- 环境检测会检查网关实例的分组状态、分组节点数量、注入分组类型等，若不通过，请按照提示进行调整或检查故障配置。您可登录混沌演练控制台在演练编排过程中查看环境检测结果，如下图所示：

环境检测 ⓘ

✔ 环境检测通过，发现1个提示项

动作组1: Kong测试 云原生网关

- 云原生网关节点重启分组检查 ✔
- 云原生网关节点重启操作提示 ⓘ

操作提示: 云原生网关节点重启时若选择其他分组注入时，需要手动输入分组ID，若此时动作组中包含多个云原生网关实例，建议拆分为多个动作组执行。

演练实施

步骤1：演练准备

准备一个云原生智能网关实例，且至少包含一个具有**2个节点**及以上的分组。如尚未创建网关实例及节点，可参考[新建网关](#)完成实例创建，并确保分组节点数不少于2个。

步骤2：演练编排

1. 登录 [混沌演练控制台](#)，进入[演练管理](#)页面，单击[新建云架构可视化演练](#)。
2. 填写演练基本信息，然后单击[下一步](#)。
3. 选择对象类型**云原生API网关**，然后单击[添加实例](#)，添加需要进行演练的实例。
4. 选择实例之后，单击演练动作模块中[立即添加](#)。
5. 添加**网关节点重启(分组)**故障动作之后单击[下一步](#)。
6. 配置动作参数。这里选择**默认分组**进行注入，然后单击[确认](#)。

⚠ 注意：

使用其他分组时，需要手动输入分组 ID。

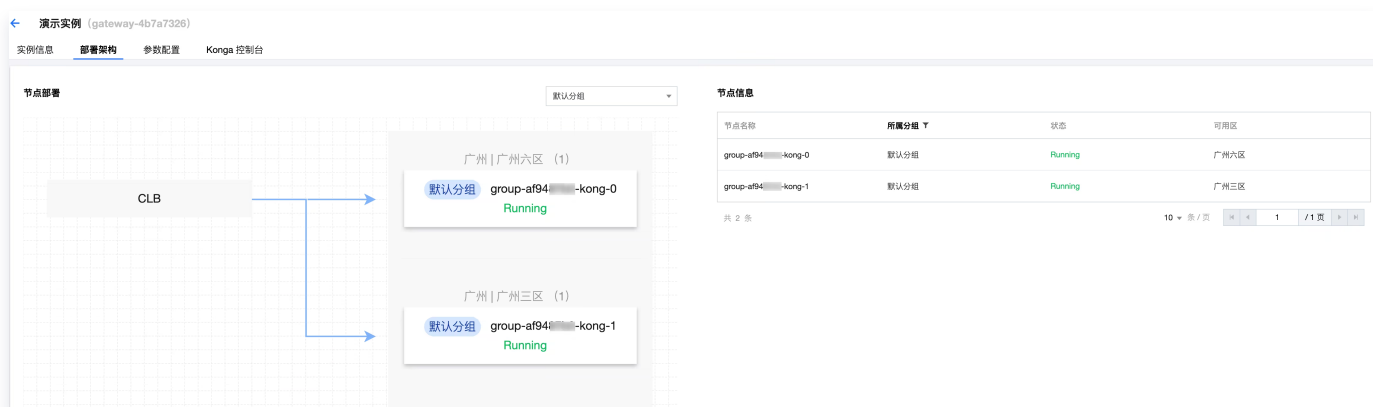
7. 单击[下一步](#)，进入全局配置，详情请参见 [快速入门](#)。
8. 确认无误后，单击[提交](#)。

步骤3：执行演练

❗ 说明：

本步骤涉及两个平台的操作切换：**执行演练动作**（如单击执行、查看日志）在 [混沌演练控制台](#) 完成；**查看演练前后网关实例的分组节点状态与结果**则在 [腾讯云微服务平台控制台](#) 完成。请根据操作需要在两个平台间切换查看。

1. 登录 [腾讯云微服务平台控制台](#)，单击需要配置的实例名称，进入实例详情页。
2. 在网关实例基本信息页面，单击[部署架构](#)。观察演练前实例分组节点信息数据。可知晓，演练前实例默认分组具有2个节点，分别位于广州六区和广州三区。



3. 由于演练为手动推进，所以需要手动执行故障动作。可单击**执行**启动故障注入。

演练动作组

护栏策略

监控指标

演练动作组

动作组1

动作组名称

云原生网关分组节点重启-文档

动作组描述

云原生网关分组节点重启-文档

实例类型

云资源

实例对象

云原生网关

环境检测结果

环境检测通过 (提示项1)

实例列表

故障注入模式: 全部注入 总实例数量: 1

实例ID	实例名称	产品版本	网关类型
gateway-4b7a7	演示实例	标准版	kong

演练动作

网关节点重启(分组)

待执行

执行

4. 故障注入过程中，可在 [腾讯云微服务平台控制台](#) 对应实例的**部署架构**模块中观察到实例的对应分组节点的状态。

- 故障中：默认分组下广州三区的节点因重启下线。

演示实例 (gateway-4b7a7)

实例信息

部署架构

参数配置

Konga 控制台

节点部署

默认分组

CLB

广州 | 广州六区 (1)

默认分组 group-af94b3-kong-0 Running

节点信息

节点名称	所属分组	状态	可用区
group-af94b3-kong-0	默认分组	Running	广州六区

共 1 条 10 条 / 页 1 / 1 页

- 故障完成：默认分组下广州三区的节点重新上线。

演示实例 (gateway-4b7a7)

实例信息

部署架构

参数配置

Konga 控制台

节点部署

默认分组

CLB

广州 | 广州六区 (1)

默认分组 group-af94b3-kong-0 Running

广州 | 广州三区 (1)

默认分组 group-af94b3-kong-1 Running

节点信息

节点名称	所属分组	状态	可用区
group-af94b3-kong-0	默认分组	Running	广州六区
group-af94b3-kong-1	默认分组	Running	广州三区

共 2 条 10 条 / 页 1 / 1 页

5. 单击**查看日志**，查看故障演练日志。

实例	状态	是否可重试 ⓘ	执行时长	操作
gateway-4b7a	成功	是	3s	查看日志 ▲ 执行记录 ▼
1 [[2023-09-25 17:37:25]] : 网关节点重启成功。 2 目标分组ID=group-af94 , 分组名=默认分组 3 可前往云原生网关实例详情页, 选择部署架构, 观察对应分组节点状态 4 重启任务列表=[RestartKongNode, CheckKongNode] 5 节点重启相关任务信息: 6 +-----+-----+-----+-----+ 7 TaskName Status StartTime EndTime 8 -----+-----+-----+----- 9 RestartKongNode success 2023-09-25 17:35:32 2023-09-25 17:35:52 10 CheckKongNode success 2023-09-25 17:35:53 2023-09-25 17:37:23 11 +-----+-----+-----+-----+				

迁移指南

与自建网关对比

最近更新时间：2026-07-23 17:07:31

本文从系统搭建及运维成本、高可用、易用性、安全性、可观测性等方面介绍云原生网关 与 自建网关的对比，辅助您进行网关技术架构的选择。

对比项	云原生网关	自建网关
系统搭建及运维成本	- 资源全托管免运维，支持节点规格变更、节点数变更、计费模式变更、产品版本升级。 - 支持按业务类型进行集群分组划分，支持配置多接入点（CLB）。 - 支持网关节点弹性伸缩（HPC、HPA）以应对流量动态变化（专业版支持）。 - 支持对接 CVM 弹性伸缩组以进行后端 CVM 节点动态扩缩。 - 支持使用 Terraform 或云 API 进行 CI/CD 流程构建。	需自行购买各种资源搭建系统，运维升级需投入精力，人工成本高。
高可用	- 接入点（CLB）、网关节点、网关配置均支持跨多可用区部署，关键链路均实现跨可用区容灾。 - 全面的流量防护能力，开源增强的限流、灰度、熔断、流量镜像等。 - 支持服务无损上线。 - 支持模拟故障进行混沌演练。	需自行探索和开发高可用保障体系，对团队技术能力要求高。
易用性	- 支持腾讯云控制台、Ingress、云 API、原生控制台（Konga）等多种方式，可按使用习惯进行选择。 - 控制台提供丰富的功能，可进行实例、路由、服务、证书、插件等相关操作，操作便捷。	需自行搭建并运维可观测体系的大量组件，对团队技术要求高。
安全性	- 支持对接腾讯云 Web 安全防火墙（WAF）。 - 支持对接腾讯云 DDoS 高防。 - 支持使用腾讯云证书管理平台（SSL）进行证书管理。 - 支持常用的 API 认证鉴权和自定义鉴权。 - 支持访问 IP 黑白名单控制。	需自行对接相关云产品。需自行配置相关插件。

可观测性	● 支持在控制台查看请求数、时延、CPU、内存、带宽、连接数等监控指标并配置自定义告警，支持对接 Prometheus。 ● 支持在控制台查看网关请求日志和错误日志，支持日志投递到腾讯云日志平台（CLS）进行分析。	支持基础监控指标，需自行搭建告警体系。
权限管理	● 集成腾讯云 CAM 平台进行网关资源权限管理。 ● 支持接入点配置 IP 黑白名单进行访问控制。	需自行实现资源级权限管控。
云产品集成	● 支持访问腾讯云微服务平台（TSF）、腾讯云容器平台（TKE）、腾讯云 Serverless 云函数（SCF）、腾讯云服务器（CVM）上的服务。 ● 支持对接腾讯云 Web 安全防火墙（WAF）、腾讯云 DDoS 平台进行安全防护。 ● 与腾讯云日志平台（CLS）、腾讯云可观测平台、腾讯云证书管理平台（SSL）、腾讯云混沌演练平台深度集成。	不支持

从自建 Kong 网关迁移到腾讯云

最近更新时间：2026-07-23 17:07:32

❗ 说明：

目前云原生网关仅对微服务场景支持开通白名单新购，其余场景**不支持新接入**。如有新购需求，请 [提交工单](#) 处理。

操作场景

本文介绍利用 Kong 官方的 **deck** 工具来完成自建 Kong 网关到腾讯云云原生 Kong 网关的配置迁移。

前置条件

- 已购买并新建了网关实例可参见 [操作文档](#)，并 [配置 admin-api](#)。
- 有自建的 Kong 网关，并配置了 **services**、**routes**、**plugins**。
- 终端机安装了 **deck** 工具，请前往 Kong 官方下载 [deck](#) 并安装到您的终端。

操作步骤

步骤1：利用 deck 工具导出自建的 Kong 网关的配置

1. 打开终端，运行如下命令导出自建 Kong 网关的配置。

```
# 注意将 kong-zj 替换为自建kong网关的ip地址或域名
deck --kong-addr http://kong-zj:8001/ dump
```

2. 检查生成的 `kong.yaml` 文件。

```
# 注意kong.yaml内容与实际情况会有所不同
cat kong.yaml
```

```
_format_version: "1.1"
plugins:
- config:
    per_consumer: false
    enabled: true
    name: prometheus
    protocols:
```

```
- grpc
- grpcs
- http
- https
services:
- connect_timeout: 60000
  host: 127.0.0.1
  name: admin-api
  path: /
  port: 8001
  protocol: http
  read_timeout: 60000
  retries: 5
  routes:
  - https_redirect_status_code: 426
    name: admin-api
    path_handling: v1
    paths:
    - /admin-api
    preserve_host: false
    protocols:
    - http
    - https
    regex_priority: 0
    request_buffering: true
    response_buffering: true
    strip_path: true
  write_timeout: 60000
- connect_timeout: 60000
  host: www.tencent.com
  name: tencent
  path: /
  port: 80
  protocol: http
  read_timeout: 60000
  retries: 5
  routes:
  - https_redirect_status_code: 426
    id: 040b0c62-4d31-4286-a595-1832e55bf568
    path_handling: v1
```



```
paths:
- /
preserve_host: false
protocols:
- http
- https
regex_priority: 0
request_buffering: true
response_buffering: true
strip_path: true
write_timeout: 60000
```

步骤2：利用 deck 工具导入配置到云原生网关 Kong

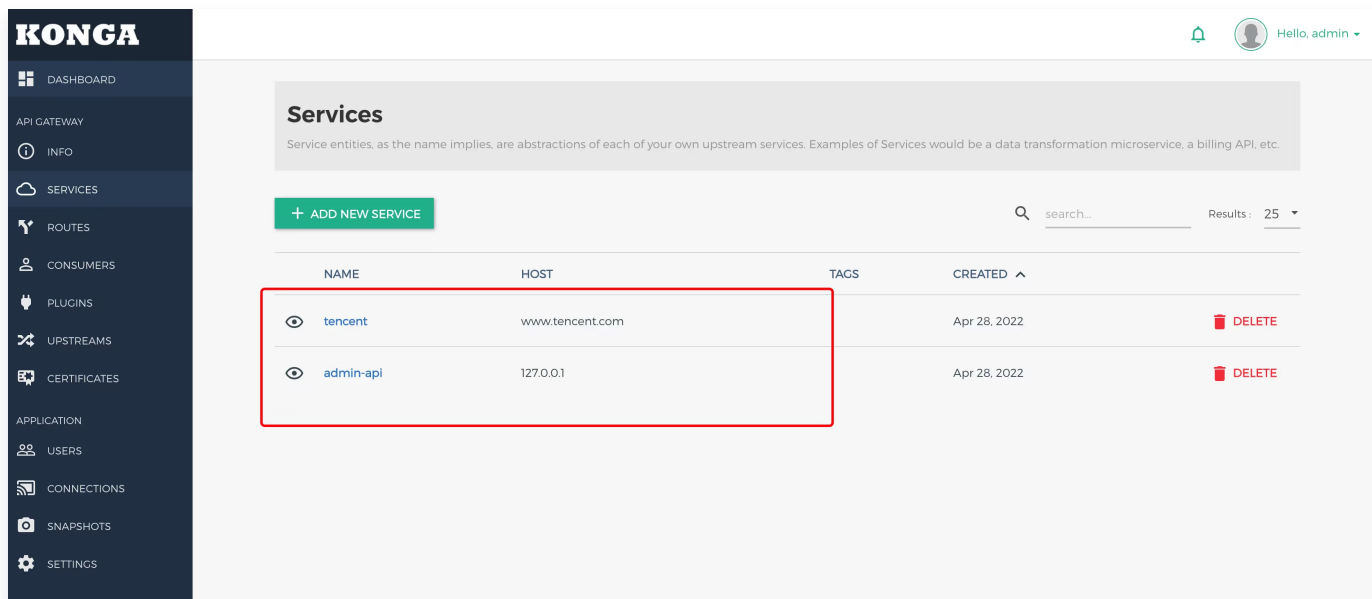
1. 打开终端，运行如下命令查看 `Kong.yaml` 中的配置与云原生 Kong 网关的配置差异。

```
# 注意将 kong-tencent 替换为腾讯云云原生网关Kong的代理地址
deck --kong-addr http://kong-tencent/admin-api diff
creating service tencent
creating route 040b0c62-4d31-4286-a595-1832e55bf568
creating plugin prometheus (global)
Summary:
  Created: 3
  Updated: 0
  Deleted: 0
```

2. 运行如下命令将 `Kong.yaml` 中的配置导入腾讯云云原生 Kong 网关中。

```
# 注意 kong-tencent 替换为腾讯云云原生网关Kong的代理地址
deck --kong-addr http://kong-tencent/admin-api sync
creating service tencent
creating route 040b0c62-4d31-4286-a595-1832e55bf568
creating plugin prometheus (global)
Summary:
  Created: 3
  Updated: 0
  Deleted: 0
```

3. 打开腾讯云云原生 Kong 网关的控制台查看配置是否导入成功。



KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

Routes

The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.

YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE

search... Results: 25

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED		
	040b0c62-4d31-4286-a595-		-	tencent	/	Apr 28, 2022	EDIT	DELETE
	admin-api		-	admin-api	/admin-api	Apr 28, 2022	EDIT	DELETE

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

Plugins

A Plugin entity represents a plugin configuration that will be executed during the HTTP request/response workflow, and it's how you can add functionalities to APIs that run behind Kong, like Authentication or Rate Limiting for example.

+ ADD GLOBAL PLUGINS

search... Results: 25

	NAME	SCOPE	APPLY TO	CONSUMER	CREATED	
	prometheus	global	All Entrypoints	All consumers	Apr 28, 2022	DELETE

注意事项

为了操作方便 **admin-api** 并未开启安全认证插件，强烈建议在配置同步完成后为 **admin-api** 开启安全认证插件。

参考

更多相关说明请参见 [deck 参考文档](#)。

从 Spring Cloud Gateway 迁移到云原生网关

最近更新时间：2026-07-23 17:07:32

本文介绍如何将服务从 Spring Cloud Gateway 迁移到云原生网关。

步骤1：明确服务来源

如果您当前的服务来源为以下几种，可以直接迁移 SCG 配置。

- K8S 服务（使用 TKE 或 EKS）。
- 注册中心服务（使用 TSF 或自建的 Nacos/Consul/Polaris）。
- 未依赖任何服务发现机制，使用域名/固定 IP/IP 列表。

如果您当前的服务来源为其他类型的注册中心，可以先执行相应的迁移操作将服务注册到 Nacos/Consul/Polaris，再进行 SCG 配置迁移。

步骤2：迁移 Spring Cloud Gateway 配置

SCG 相关功能配置可以在控制台进行相应操作完成：

1. 注册中心/K8S 集群

云原生网关通过服务来源关联用户的注册中心或 K8s 集群，添加对应类型的服务来源后，即可添加相应服务作为网关服务，可参考如下文档进行服务来源配置：[服务来源](#)。

2. 服务配置

Spring Cloud Gateway 的 service 对应云原生网关的服务，可参考如下文档进行服务配置：[服务](#)。

3. 路由配置

Spring Cloud Gateway 的 routes 对应云原生网关的路由，可参考如下文档进行路由配置：[路由](#)。

4. 常用的 Predicate 配置

- Host/Method/Path Route Predicate：云原生网关支持根据 Host，Method 或 Path 进行路由匹配，可在创建路由时设置匹配条件。
- Header/Cookie/Query Route Predicate：云原生网关支持根据上述参数进行路由，可通过灰度规则实现，可参考以下文档进行配置：[灰度发布](#)。

5. 默认 Filter

Spring Cloud Gateway 的 filter 对应云原生网关的配置如下：

GatewayFilter	对应的云原生网关能力	相关文档
AddRequestHeader, RemoveRequestHeader, AddRequestParameter, AddResponseHeader,	使用 tse-proxy-rewrite 插件	使用重写插件改写请求或响应

SetResponseHeader		
Hystrix, RequestRateLimiter	限流能力	限流策略
PrefixPath	默认根据前缀匹配方式进行路由匹配	—
PreserveHostHeader, StripPath, Retry	创建路由时支持保持 Host、StripPath 选项、支持设置重试次数	服务与路由
SetPath, RewritePath	使用 Request Transformer/Response Transformer 插件	—

6. 自定义 Filter

认证鉴权，云原生网关支持主流认证鉴权方式，可参考以下文档进行配置。

- [Basic Auth 安全认证](#)
- [IP 访问控制](#)
- [Key Auth 密钥认证](#)
- [使用 HMACAuth 认证访问](#)
- [使用 ACL 访问控制](#)
- [配置 JWT 认证鉴权](#)

7. 自定义 Filter

云原生网关支持使用自定义插件进行自定义逻辑处理，您需要编码将自定义 Filter 的逻辑通过自定义插件实现。详细可参见 [自定义插件](#)。

步骤3：配置网关监控告警

云原生网关支持查看监控告警，您可以配置相关监控信息掌握当前网关状况，请参考以下文档使用监控告警。

- [查看默认监控](#)
- [使用 Prometheus 监控](#)

步骤4：流量迁移

在完成上述准备过程后，可按照灰度方式逐步进行流量切换，最终完成流量迁移。

第一阶段：选取部分非核心业务场景配置服务和路由，修改访问地址，进行小规模验证。

第二阶段：逐步扩展至核心业务，对核心业务进行配置变更，并迁移部分核心场景流量。

第三阶段：待充分验证后，切换域名解析进行全量迁移。

实践教程

认证鉴权

开启 admin-api 并配置安全认证

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍 Kong 如何开启 admin-api 并为 admin-api 配置 Basic Auth 安全认证插件。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

操作步骤

步骤1：在 Kong 的管理控制台上配置 admin api 的服务和路由

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 单击公网访问地址，进入管理控制台页面，在左侧导航栏单击 **SERVICES**，进入服务列表。

- 单击 **ADD NEW SERVICE**，在弹出的对话框中填写内容，URL 的内容填写为 `http://127.0.0.1:2001/`，创建一个服务。

CREATE SERVICE

Name

admin-api

(optional)

The service name.

Description

(optional)

An optional service description.

Tags

(optional)

Optionally add tags to the service

Url

http://127.0.0.1:2001/

(shorthand-attribute)

Shorthand attribute to set `protocol`, `host`, `port` and `path` at once. This attribute is write-only (the Admin API never "returns" the url).

- 创建成功后，在服务列表中单击已创建好的服务，进入服务详情页面，在左侧导航栏单击 **Routes**，进入路由列表。

- 单击 **ADD ROUTE**，在弹出的对话框中填写内容，**Path** 的内容填写为 `/admin-api`，创建一个路由。

ADD ROUTE TO ADMIN-API

* For hosts, paths, methods and protocols, snis, sources, headers and destinations press enter to apply every value you type

Name

(optional)

The name of the Route.

Tags

(optional)

Optionally add tags to the route

Hosts

(semi-optional)

A list of domain names that match this Route. For example: example.com. At least one of hosts, paths, or methods must be set.

Paths

/admin-api

(semi-optional)

A list of paths that match this Route. For example: /my-path. At least one of `hosts`, `paths`, or `methods` must be set.

Headers

(semi-optional)

One or more lists of values indexed by header name that will cause this Route to match if present in the request. The `Host` header cannot be used with this attribute: hosts should be specified using the `hosts` attribute.
Field values format example: `x-some-header:foo,bar`

Path handling

v1

Controls how the Service path, Route path and requested path are combined when sending a request to the upstream. See above for a detailed description of each behavior. Accepted values are: `"v0"`, `"v1"`. Defaults to `"v1"`.

8. 打开浏览器通过公网地址访问 `admin api`，`uri` 的内容填写为 `/admin-api`。

```
⬅ ➡ 🔍 不安全 | 🌐 📄 /admin-api

{"tagline":"Welcome to kong","timers":{"pending":9,"running":1},"lua_version":"LuaJIT 2.1.0-beta3","version":"2.4.1","node_id":"7cd8e9e3-e4cb-4bde"
{"loggly":true,"datadog":true,"ldap-auth":true,"statsd":true,"bot-detection":true,"aws-lambda":true,"request-termination":true,"azure-functions":true,"t
cache":true,"session":true,"acme":true,"grpc-web":true,"jwt":true,"acl":true,"correlation-id":true,"cors":true,"oauth2":true,"
auth":true,"basic-auth":true,"ip-restriction":true,"request-transformer":true,"response-transformer":true,"request-size-limiting":true,"rate-limit
{"admin_listen":{"127.0.0.1:8001 reuseport backlog=16384","127.0.0.1:8444 http2 ssl reuseport backlog=16384"},"status_listen":{"off"},"stream_listen
["/usr/local/kong/ssl/admin-kong-default.crt","/usr/local/kong/ssl/admin-kong-default-ecdsa.crt"],"nginx_events_worker_connections":"auto","status
kong","nginx_main_user":"kong kong","nginx_daemon":"off","dns_stale_ttl":4,"nginx_main_daemon":"off","nginx_worker_processes":"auto","nginx_main_w
IP","nginx_proxy_real_ip_header":"X-Real-
IP","real_ip_recursive":"off","nginx_proxy_real_ip_recursive":"off","client_max_body_size":"0","nginx_http_client_max_body_size":"0","client_body_
*","pg_ssl":false,"pg_ssl_verify":false,"pg_max_concurrent_queries":0,"pg_semaphore_timeout":60000,"pg_ro_ssl":false,"pg_ro_ssl_verify":false,"cas
["127.0.0.1"},"cassandra_port":9042,"cassandra_ssl":false,"cassandra_ssl_verify":false,"cassandra_write_consistency":"ONE","db_update_frequency":5
l_enabled":false,"cassandra_lb_policy":"RequestRoundRobin","status_ssl_enabled":false,"cassandra_refresh_frequency":60,"cassandra_repl_strategy":
{"dc1":"dc2","dc2":"dc3"},"cassandra_schema_consensus_timeout":10000,"ssl_protocols":"TLSv1.1 TLSv1.2 TLSv1.3","nginx_http_ssl_protocols":"TLSv1.2 TLSv1.3
TLSv1.3","ssl_prefer_server_ciphers":"on","nginx_http_ssl_prefer_server_ciphers":"off","nginx_stream_ssl_prefer_server_ciphers":"off","ssl_dhparam
session_tickets":"on","nginx_http_ssl_session_tickets":"on","nginx_stream_ssl_session_tickets":"on","ssl_session_timeout":"1d","nginx_http_ssl_ses
,"proxy_error_log":"logs/error.log","proxy_stream_access_log":"logs/access.log
basic","proxy_stream_error_log":"logs/error.log","admin_access_log":"logs/admin_access.log","admin_error_log":"logs/error.log","status_access_log"
ue,"lua_ssl_trusted_certificate":{"},"lua_ssl_verify_depth":1,"lua_ssl_protocols":"TLSv1.1 TLSv1.2 TLSv1.3","nginx_http_lua_ssl_protocols":"TLSv1.1
TLSv1.3","worker_consistency":"strict","untrusted_lua":"sandbox","untrusted_lua_sandbox_requires":{"},"untrusted_lua_sandbox_environment":{"},"clust
olk25he2","upstream_keepalive_idle_timeout":60,"upstream_keepalive_max_requests":100,"pg_user":"database-olk25he2","cassandra_timeout":5000,"pg_ti
{"},"go_plugins_dir":"off","go_pluginserver_exe":"/usr/local/bin/go-pluginserver","cassandra_username":"kong","anonymous_reports":true,"nginx_main_
{"value":"auto","name":"worker_processes"},"{"value":"auto","name":"worker_rlimit_nofile"},"{"value":"auto","name":"worker_connections"},"{"value":"on","n
{"port":8001,"http2":false,"proxy_protocol":false,"deferred":false,"bind":false,"backlog=16384":true,"ip":"127.0.0.1","listener":{"127.0.0.1:8001
{"port":8444,"http2":true,"proxy_protocol":false,"deferred":false,"bind":false,"backlog=16384":true,"ip":"127.0.0.1","listener":{"127.0.0.1:8444 ss
{"value":"8k","name":"client_body_buffer_size"},"{"value":"0","name":"client_max_body_size"},"{"value":"prometheus_metrics 5m","name":"lua_shared_d
{"value":"/usr/local/kong/ssl/ffdhe2048.pem","name":"ssl_dhparam"},"{"value":"off","name":"ssl_prefer_server_ciphers"},"{"value":"TLSv1.2 TLSv1.3","
{"value":"1d","name":"ssl_session_timeout"},"{"port":8000,"http2":false,"proxy_protocol":false,"deferred":false,"bind":false,"b
backlog=16384","ssl":false,"reuseport":true},"{"port":8443,"http2":true,"proxy_protocol":false,"deferred":false,"bind":false,"backlog=16384":true,"
backlog=16384","ssl":true,"reuseport":true},"{"stream_listeners":{"},"nginx_upstream_directives":{"},"nginx_proxy_directives":{"value":"X-Real-IP","
"},"headers":{"server_tokens","latency_tokens"},"proxy_listen":{"0.0.0.0:8000 reuseport backlog=16384","0.0.0.0:8443 http2 ssl reuseport backlog=1
5m","name":"lua_shared_dict"},"{"value":"TLSv1.1 TLSv1.2 TLSv1.3","name":"lua_ssl_protocols"},"{"value":"/usr/local/kong/ssl/ffdhe2048.pem","name":"
TLSv1.3","name":"ssl_protocols"},"{"value":"on","name":"ssl_session_tickets"},"{"value":"1d","name":"ssl_session_timeout"},"{"db_cache_warmup_entiret
{"value":"ssl_cert_key_default":"/usr/local/kong/ssl/kong-default.key","nginx_sproxy_directives":{"},"nginx_err_logs":"/usr/local/kong/logs/error.log"},"d
{"value":"10m","name":"client_body_buffer_size"},"{"value":"10m","name":"client_max_body_size"},"{"lua_package_path":"./?.lua;./?.init.lua"},"nginx
{"},"dns_no_sync":false,"nginx_pid":"/usr/local/kong/pids/nginx.pid","host_ports":{"},"role":"traditional","error_default_type":"text/plain","admin_
```

步骤2：为 admin api 配置 basic-auth 安全认证插件

1. 打开 Kong 的管理控制台，在左侧导航栏单击 **CONSUMERS**，进入消费者列表。
2. 单击 **CREATE CONSUMER**，在弹出的对话框中填写内容。

CREATE CONSUMER

username
(semi-optional)

admin

The username of the consumer. You must send either this field or `custom_id` with the request.

custom_id
(semi-optional)

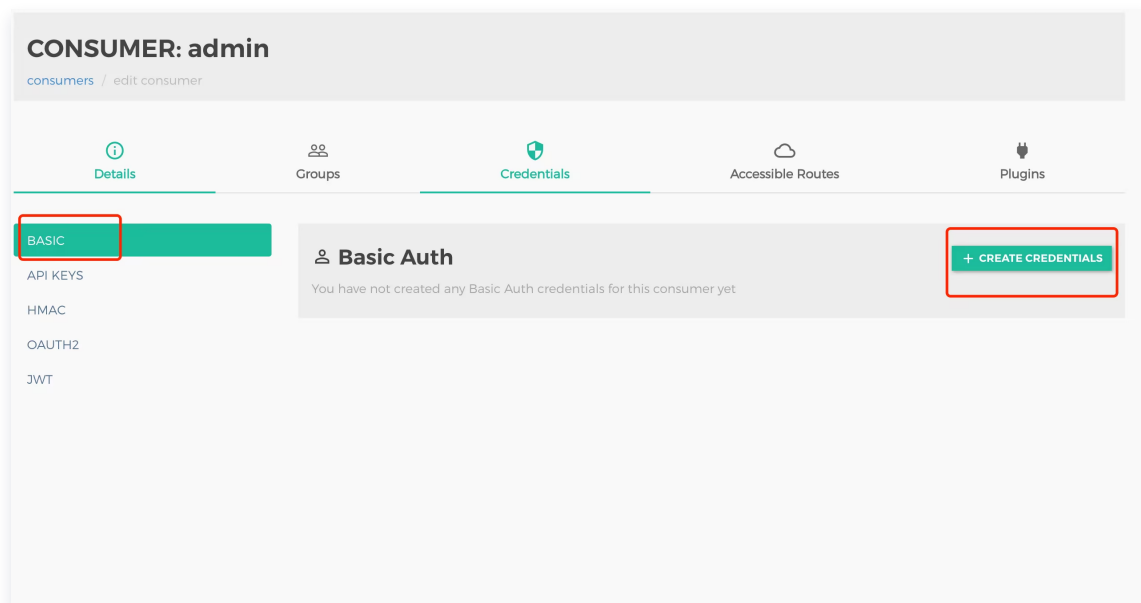
Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or `username` with the request.

Tags
(optional)

Optionally add tags to the consumer

SUBMIT CONSUMER

3. 单击 **Credentials** 左侧的选项选择 **BASIC**，并单击 **CREATE CREDENTIALS**。



在弹出的对话框中填写内容，创建一个消费者。

BASIC AUTH

Manage Basic Auth credentials for **admin**

username

(required)

admin

The username to use in the Basic Authentication

password

(optional)

admin

The password to use in the Basic Authentication

✓

SUBMIT

4. 在左侧导航栏中依次单击 **SERVICES** > **admin-api** > **Plugins** > **ADD Plugin**，选择 **Basic Auth** 插件并完成添加。

ADD PLUGIN

Authentication

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

AUTHENTICATION

Protect your services with an authentication layer

Basic Auth



Add Basic Authentication to your..

ADD PLUGIN

Key Auth



Add a key authentication to your..

ADD PLUGIN

Oauth2



Add an OAuth 2.0 authentication to your...

ADD PLUGIN

Hmac Auth



Add HMAC Authentication to your..

ADD PLUGIN

Jwt



Verify and authenticate JSON Web Tokens

ADD PLUGIN

Ldap Auth



Integrate Kong with a LDAP server

ADD PLUGIN

ADD BASIC AUTH

Add Basic Authentication to your APIs, with username and password protection. The plugin will check for valid credentials in the **Proxy-Authorization** and **Authorization** header (in this order).

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

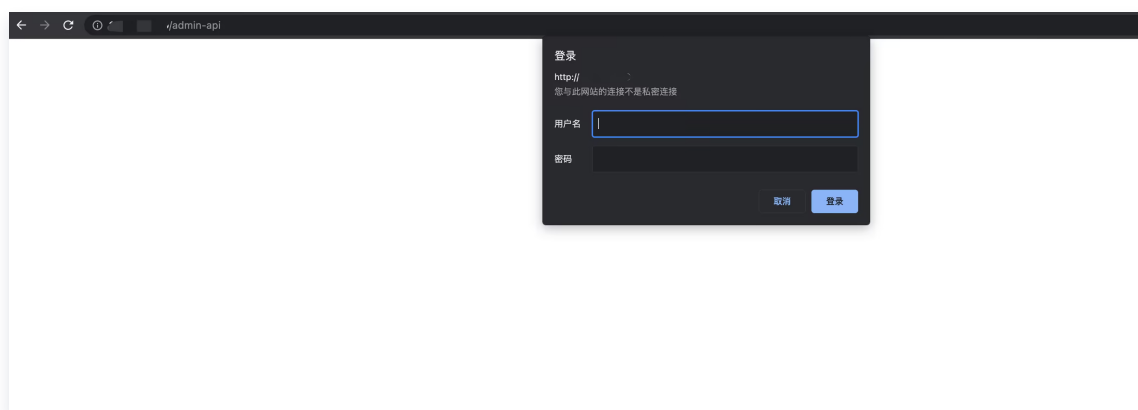
anonymous

hide credentials ☐ NO

An optional boolean value telling the plugin to hide the credential to the upstream API server. It will be removed by Kong before proxying the request

✓ ADD PLUGIN

5. 打开浏览器通过公网地址访问 **admin api**，**uri** 的内容填写为 `/admin-api`，输入之前配置好的用户名和密码即可访问。



参考

更多说明请参见 [Kong API Lookback](#)。

使用 IP 访问控制

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍如何在云原生网关上通过 Kong IP Restriction 插件实现 IP 访问控制，包括以下两种场景：

- 配置 IP 黑/白名单列表进行访问控制
- 配置 IP 黑/白名单 CIDR 段进行访问控制

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

操作步骤

场景一：配置 IP 黑/白名单列表进行访问控制

本场景操作步骤以黑名单 IP 列表为例，指导如何拒绝某个（些）IP 访问，白名单配置与之类似。

- 登录 [TSF 控制台](#)。
- 在左侧导航栏选择 [云原生网关](#)，单击目标实例进入实例详情。
- 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-1234567890-konga-gz-apigw.tencentcs.com:1337/>  关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-1234567890-konga-in-apigw.tencentcs.com:1337/>  关闭

管理员账号

admin 

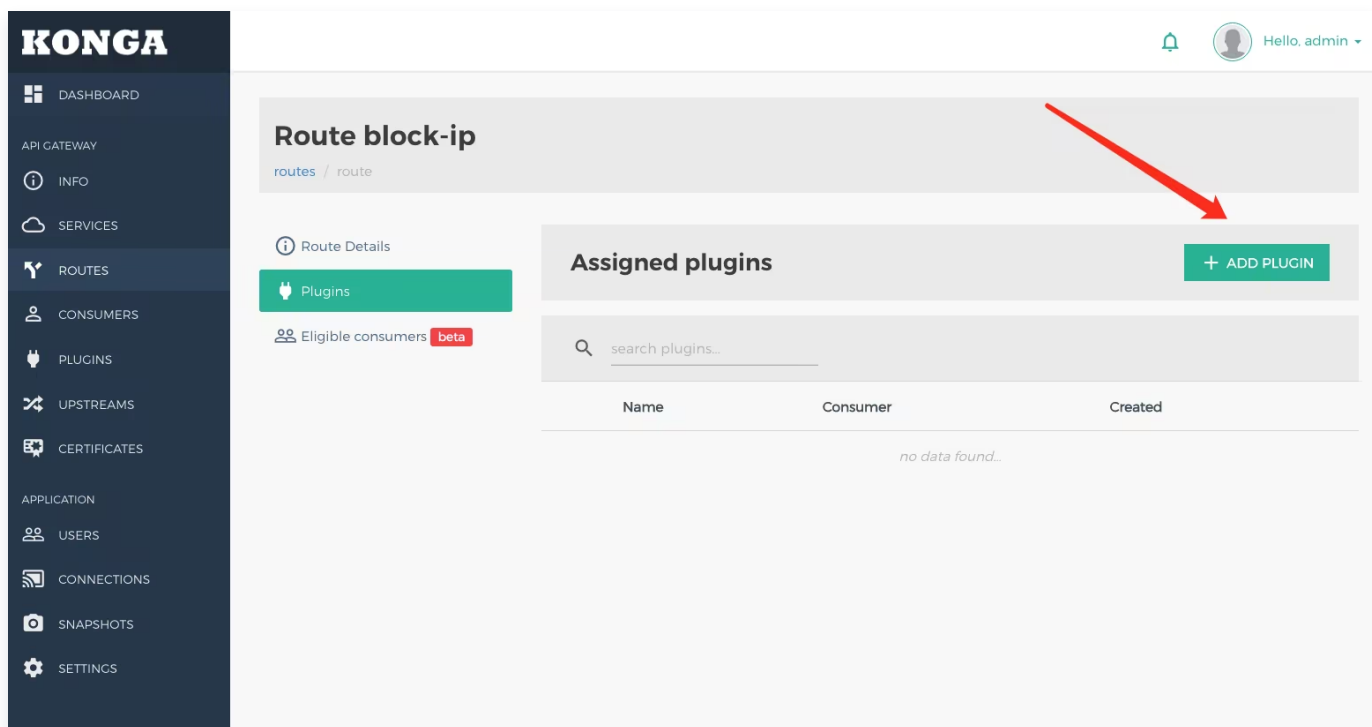
管理员密码

*****[显示](#) 

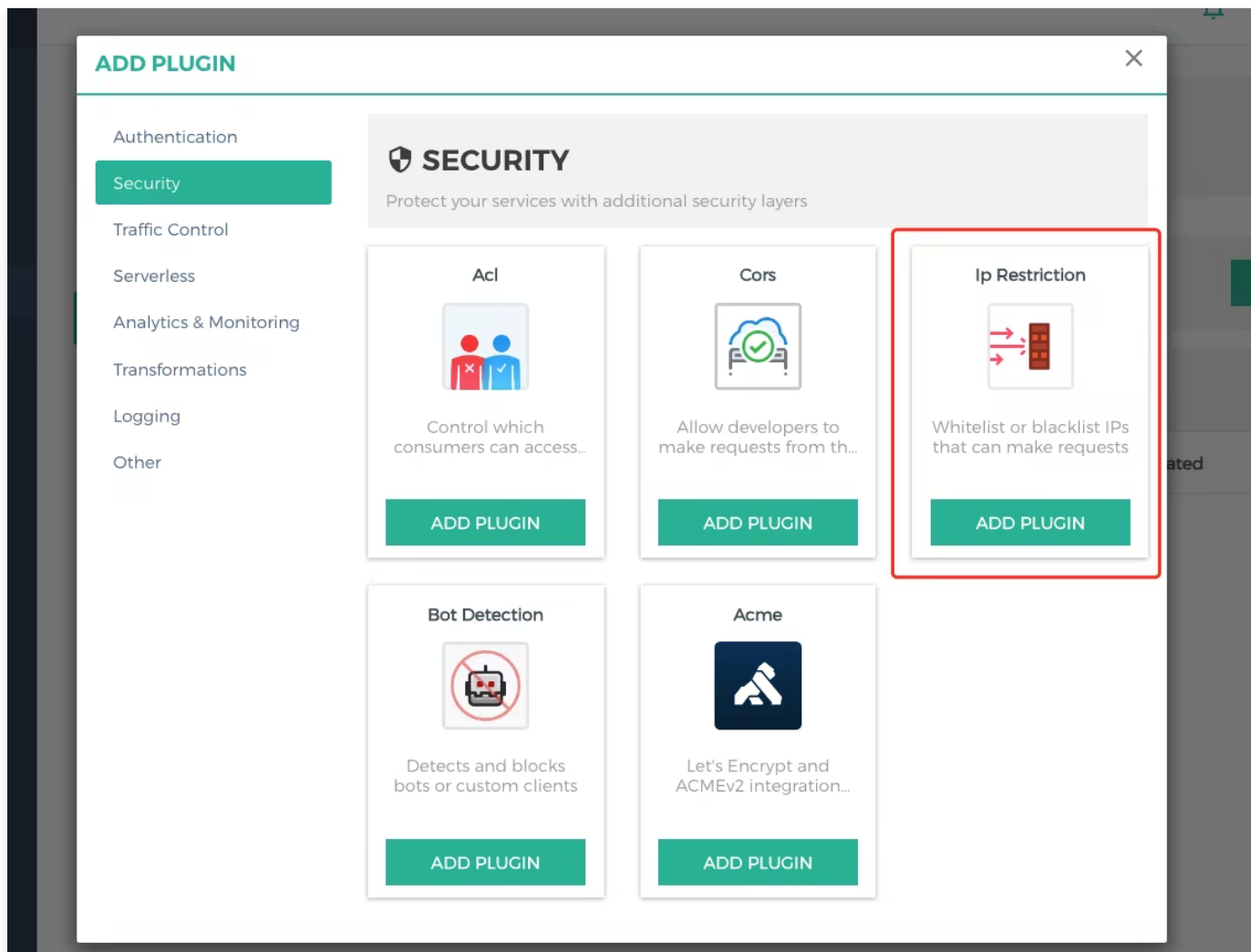
为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

- 单击公网访问地址进入 Konga 控制台

5. 登录 Konga 管理控制台，进入需要 IP 限制的 Route 详情页，单击 **Add Plugin** 按钮创建插件。



6. 在插件市场的 Security 分组中选择 IP Restriction 插件，单击 Add Plugin。



7. 在插件配置中输入需要配置限制的 IP，按 Enter，并保存。

- `allow`：填写允许访问的 IP，多个 IP 需要填写多个。
- `deny`：填写拒绝访问的 IP，多个 IP 需要填写多个。
- `consumer`：填写需要应用访问控制的 Consumer ID，如空缺，表示该 IP 访问控制应用于所有 consumer。

注意

allow 和 deny 两者至少配置一项。

Authentication

ADD IP RESTRICTION

Restrict access to an API by either whitelisting or blacklisting IP addresses. Single IPs, multiple IPs or ranges in CIDR notation like `10.10.10.0/24` can be used.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

allow

Tip: Press `Enter` to accept a value.

deny

113. .58 X 113. .68 X

Tip: Press `Enter` to accept a value.

✓ ADD PLUGIN

8. 返回 route 页面，确定该 route 已经绑定的和创建的插件。

Plugin added successfully!

Route block-ip

routes / route

Route Details

Plugins

Eligible consumers **beta**

Assigned plugins

+ ADD PLUGIN

search plugins...

Name	Consumer	Created	
ip-restriction	All consumers	Apr 25, 2022	DELETE

9. 发起 API 请求，该 IP 访问被限制。

```
HTTP/1.1 403 Forbidden
Connection: keep-alive
Content-Length: 48
```

```
Content-Type: application/json; charset=utf-8
Date: Mon, 25 Apr 2022 02:57:32 GMT
X-Kong-Response-Latency: 1

{
  "message": "Your IP address is not allowed"
}
```

场景二：配置 IP 黑/白名单 CIDR 段进行访问控制

本场景操作步骤以白名单 IP CIDR 段为例，指导如何允许某个 IP 段的请求访问，黑名单配置与之类似。

1. 进入需要 IP 限制的 Route 详情页，单击 Add Plugin 按钮创建插件。
2. 在插件配置中填写以下配置，按 Enter 键，并保存。
 - `allow`：填写允许访问的 IP CIDR 段。
 - `deny`：填写拒绝访问的 IP CIDR 段。
 - `consumer`：填写需要应用访问控制的 Consumer ID，如空缺，表示该 IP 访问控制应用于所有 consumer。

⚠ 注意

`allow` 和 `deny` 两者至少配置一项。

Route block-ip

EDIT IP RESTRICTION

Restrict access to an API by either whitelisting or blacklisting IP addresses. Single IPs, multiple IPs or ranges in CIDR notation like `10.10.10.0/24` can be used.

ENABLED

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

allow

Tip: Press **Enter** to accept a value.

deny

Tip: Press **Enter** to accept a value.

SUBMIT CHANGES

3. 发起 API 请求，使用不在该 CIDR 段的 IP 访问，请求被拒绝。

```
HTTP/1.1 403 Forbidden
Date: Mon, 25 Apr 2022 03:06:58 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Content-Length: 48
X-Kong-Response-Latency: 14

{
  "message": "Your IP address is not allowed"
}
```

注意事项

当对一个 IP 同时应用了 allow 和 deny，该 IP 会被限制访问。

参考

更多相关说明请参见 [Kong 插件文档](#)。

最近更新时间: 2026-07-23 17:07:35

本文介绍如何在云原生网关上通过 Key Authentication 密钥身份认证插件实现认证访问。

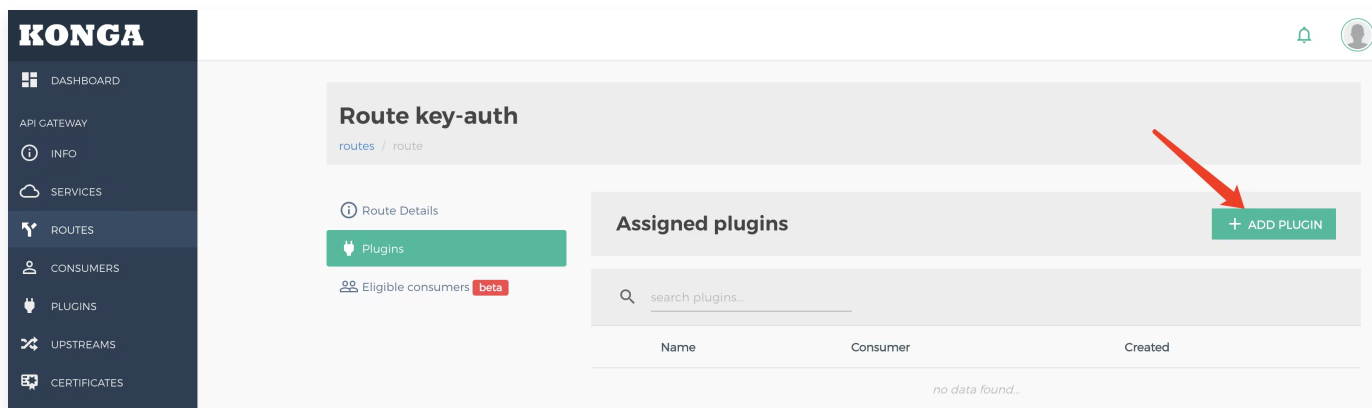
- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

本场景操作步骤以配置 Route 插件为例，指导如何实现密钥身份认证访问。

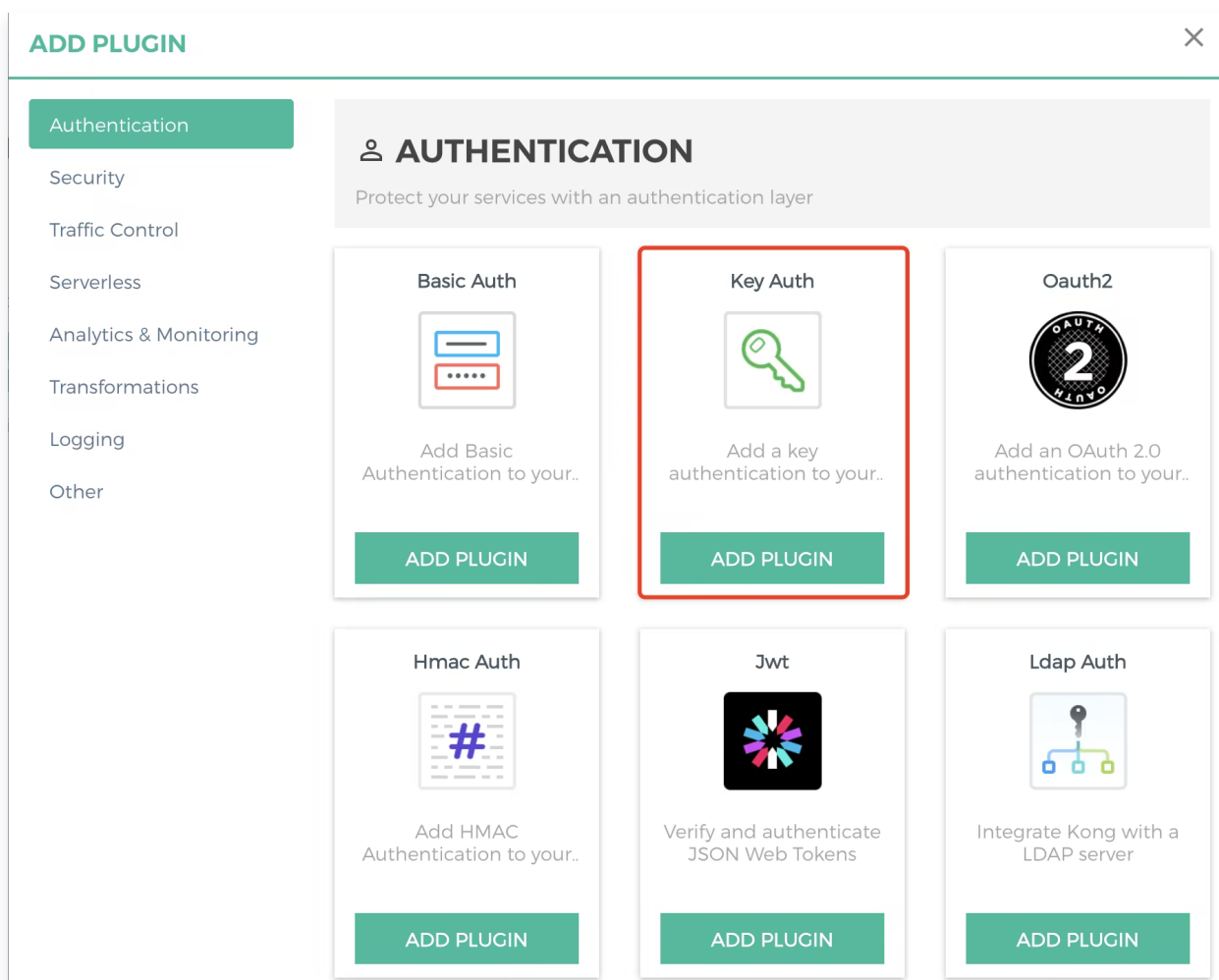
1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择**云原生网关**，单击目标实例进入实例详情。
3. 在基础信息页面，单击 **Konga 控制台** 标签查看管理控制台登录方式。

实例信息		部署架构	参数配置	Konga 控制台
公网访问地址	https://gateway-1122728-konga-jl.apigw.tencentcloud.com/1234567/  关闭			
公网访问策略	白名单: 0.0.0.0 修改			
内网访问地址	https://gateway-1122728-konga-jl.apigw.tencentcloud.com/1234567/  关闭			
管理员账号	admin 			
管理员密码	*****  显示			

4. 登录 Konga 管理控制台，进入需要密钥身份认证访问的 Route 详情页，单击 Add Plugin 添加插件。



5. 在插件市场的 Authentication 分组中选择 Key Auth 插件，单击 Add Plugin。



6. 设置密钥身份认证插件参数，并保存。

- `key names`：插件将要查找键的参数名称数组（默认为 `apikey`）。客户端必须以其中一个密钥名称发送身份验证密钥，插件将尝试从 `header` 或具有相同名称的查询参数读取凭据。

注意

键名只能包含[a-z]，[A-Z]，[0-9]，[_]和[-]。由于 NGINX 默认值中的附加限制，不允许在标题中使用下划线。

- `hide_credentials` : 是否隐藏来自上游服务的凭据。
- `anonymous` : 如果身份验证失败, 则用作“匿名”使用者的可选字符串 (使用者 `uuid`) 值。如果为空 (默认), 则请求将失败, 并且身份验证失败4xx。请注意, 此值必须引用 Kong 内部的 `Consumer id` 属性, 而不是其 `custom_id`。
- `key in header` : 如果启用, 插件将要查找键的参数放在 `header`。
- `key in query` : 如果启用, 插件将要查找键的参数放在 `query`。
- `key in body` : 如果启用, 插件将读取请求正文 (如果所述请求有一个并且支持其 MIME 类型) 并尝试在其中找到密钥。支持的 MIME 类型是 `application/x-www-form-urlencoded`, `application/json` 和 `multipart/form-data`。
- `run on preflight` : 插件是否应在 `OPTIONS` 预检请求上运行。

ADD KEY AUTH

×

Add Key Authentication (also referred to as an API key) to your APIs. Consumers then add their key either in a querystring parameter or a header to authenticate their requests.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

key names

Tip: Press **Enter** to accept a value.

Describes an array of comma separated parameter names where the plugin will look for a key. The client must send the authentication key in one of those key names, and the plugin will try to read the credential from a header or the querystring parameter with the same name.

hide_credentials

NO

An optional boolean value telling the plugin to hide the credential to the upstream API server. It will be removed by Kong before proxying the request.

anonymous

key in header

YES

key in query

YES

key in body

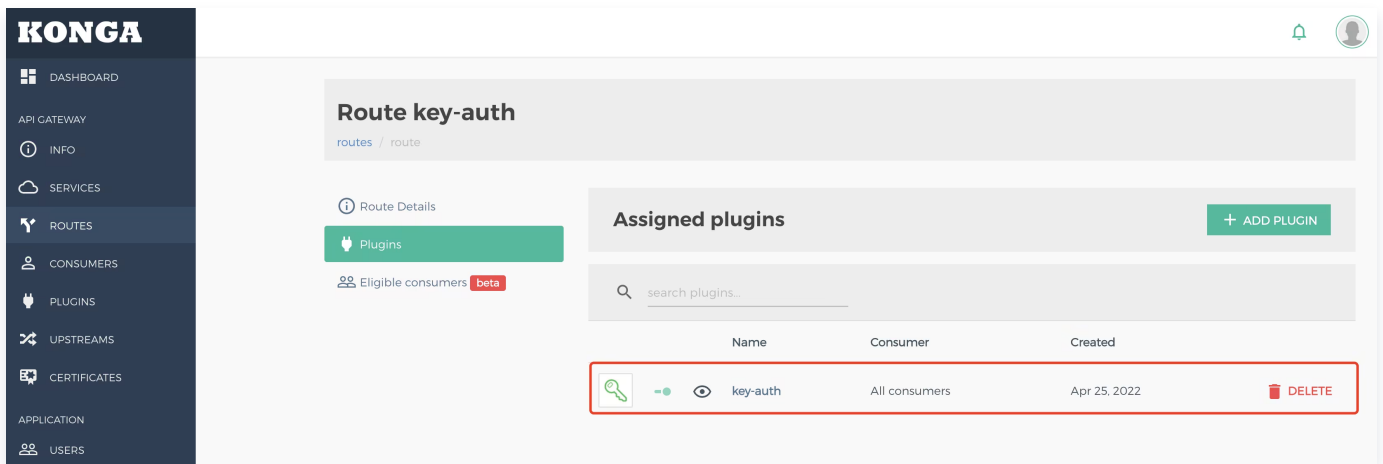
NO

run on preflight

YES

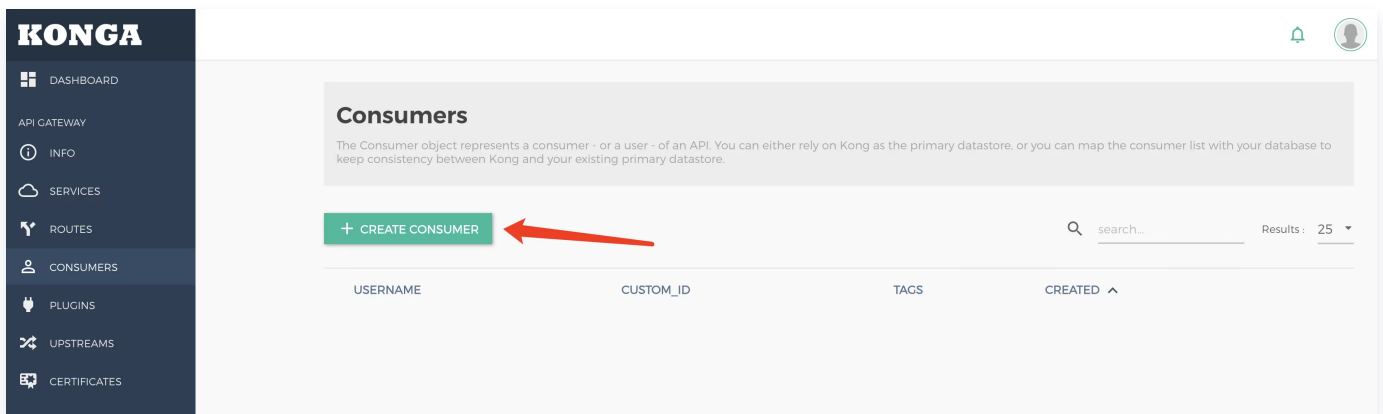
✓ ADD PLUGIN

7. 返回 route 页面，确定该 route 已经绑定的创建的插件。



步骤2：创建 Consumer

1. 进入 CONSUMERS 页面创建 Consumer。



- `username`：用户（应用）名，必须指定此字段或 `custom_id`。
- `custom_id`：用于将使用者映射到另一个数据库的自定义标识符。必须指定此字段或 `username`。
- `Tags`：标签。

CREATE CONSUMER

username
(semi-optional)

test

The username of the consumer. You must send either this field or `custom_id` with the request.

custom_id
(semi-optional)

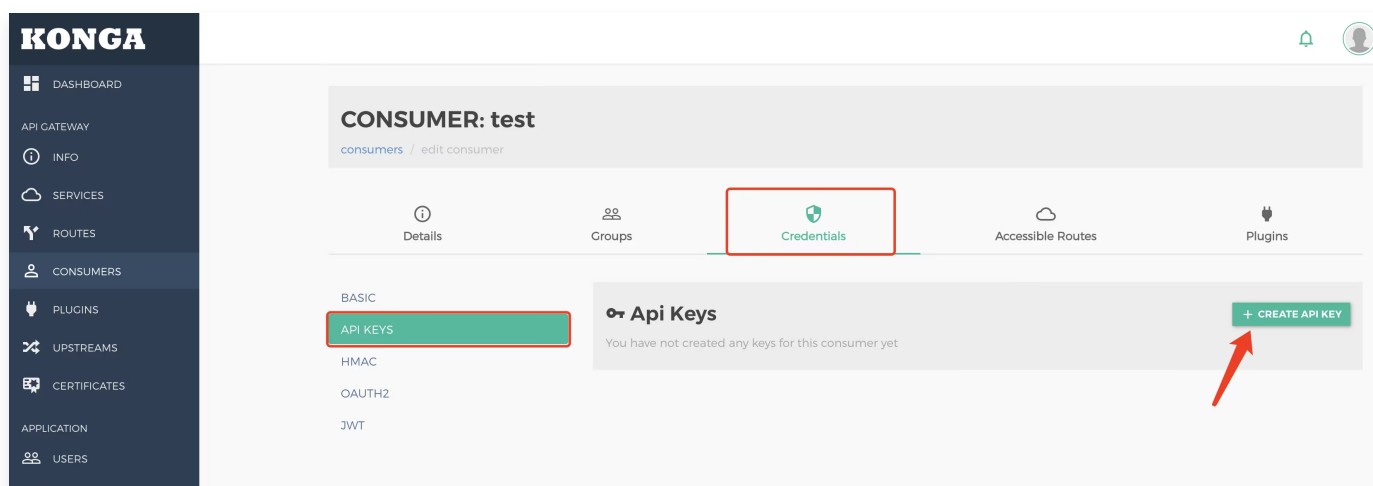
Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or `username` with the request.

Tags
(optional)

Optionally add tags to the consumer

✓ SUBMIT CONSUMER

2. 为 Consumer 创建 api key 凭证。



key : key 值，默认为空会自动生成 key。

CREATE API KEY

Create Api Key for **test**

key
(optional)

You can optionally set your own unique **key** to authenticate the client. If missing, Kong will generate one.

SUBMIT

3. 查看生成的 key。

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONSUMER: test

consumers / edit consumer

Details

Groups

Credentials

Accessible Routes

Plugins

BASIC

API KEYS

HMAC

OAuth2

JWT

Api Keys

+ CREATE API KEY

#	key	created
1.	iOu4k	Apr 25, 2022 DELETE

步骤3：发起 API 请求

1. 未携带 apikey 会返回报错：`curl http://kong:8000/{proxy path}/`。

说明：
proxy path 为请求地址。

```
HTTP/1.1 401 Unauthorized
Server:
Date: Mon, 25 Apr 2022 14:16:10 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 45
```

```
Connection: keep-alive
WWW-Authenticate: Key realm="kong"
X-Kong-Response-Latency: 1

{
  "message": "No API key found in request"
}
```

2. 正确携带 apikey 则放通请求: `curl http://kong:8000/{proxy path}?apikey=<some_key>`。

❗ 说明:

- proxy path 为请求地址。
- apikey 即为需要访问的使用的 key。

```
HTTP/1.1 200 OK
Server:
Date: Mon, 25 Apr 2022 14:19:19 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 6
Connection: keep-alive
X-Kong-Upstream-Latency: 5
X-Kong-Proxy-Latency: 0
Via: kong/2.4.1

ok
```

注意事项

Key Auth 不支持针对不同 consumer 使用不同的 Api Key，如需针对单个用户进行 Key Auth 认证，需配合 post function 插件一起使用，自定义增加识别用户 ID 的处理逻辑。

参考

更多相关说明请参见 [Kong 密钥认证插件官方文档](#)。

使用 HMAC Auth 认证访问

最近更新时间：2026-07-23 17:07:33

操作场景

本文介绍如何在 Kong 云原生网关上通过 HMAC Authentication 认证插件实现认证访问。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

操作步骤

本场景操作步骤以配置 Route 插件为例，指导如何实现 HMAC Auth 认证访问。

步骤1：配置 HMAC Authentication 认证插件

- 登录 [TSF 控制台](#)。
- 在左侧导航栏选择 **云原生网关**，单击目标实例进入实例详情。
- 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-41-11111-konga-gz.apigw.tencentcs.com:1337/>  关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-41-11111-konga-in.gz.apigw.tencentcs.com:1337/>  关闭

管理员账号

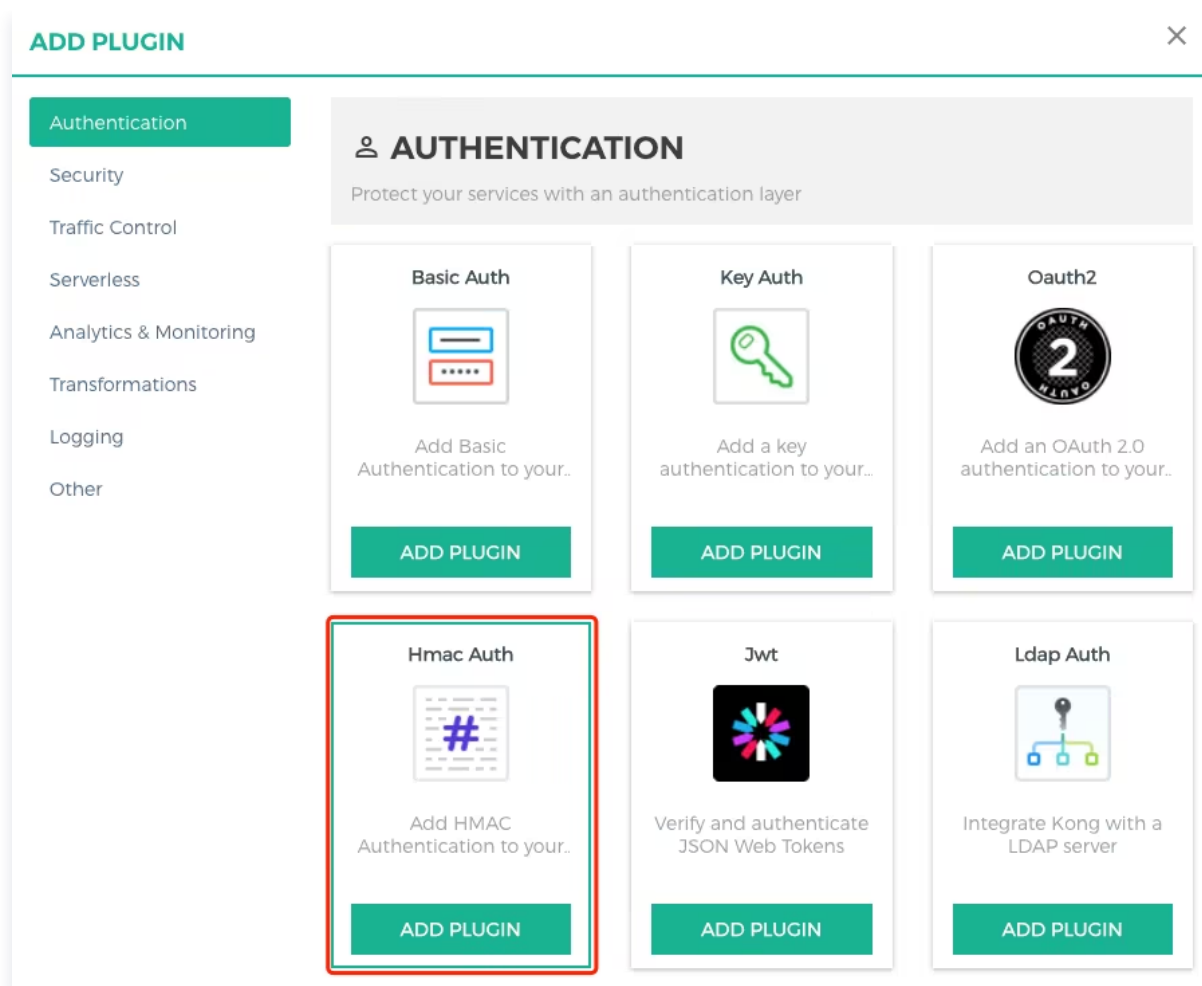
admin 

管理员密码

*****[显示](#) 

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

- 登录 Konga 管理控制台，进入需要限流的 **Route 详情页**，单击 **Add Plugin** 按钮创建插件，在 **Authentication** 分组下选择 **HMAC Auth** 插件。



5. 设置密钥身份认证插件参数，按 Enter 键，并保存。

- `consumer`：填写需要应用访问控制的 Consumer ID，如空缺，表示该 IP 访问控制应用于所有 consumer。
- `hide_credentials`：是否隐藏来自上游服务的凭据。
- `anonymous`：如果身份验证失败，则用作“匿名”使用者的可选字符串（使用者 uuid）值。如果为空（默认），则请求将失败，并且身份验证失败 4xx。请注意，此值必须引用 Kong 内部的 Consumer ID 属性，而不是其 `custom_id`。
- `clock_skew`：时钟偏移在几秒钟内以防止重放攻击。
- `validate_request_body`：是否启用 body 验证。
- `enforce_headers`：客户端至少应用于 HTTP 签名创建的 header 列表。
- `algorithms`：用户想要支持的 HMAC 摘要算法列表。允许的值为 `hmac-sha1`, `hmac-sha256`, `hmac-sha384` 和 `hmac-sha512`，默认全部支持。

EDIT HMAC AUTH

×

Add HMAC Signature Authentication to your APIs to establish the identity of the consumer. The plugin will check for valid signature in the **Proxy-Authorization** and **Authorization** header (in this order). This plugin implementation follows the [draft-cavage-http-signatures-00](#) draft with slightly changed signature scheme.

ENABLED

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

hide credentials

NO

An optional boolean value telling the plugin to hide the credential to the upstream API server. It will be removed by Kong before proxying the request

clock skew

300

Clock Skew in seconds to prevent replay attacks

anonymous

validate request body

YES

enforce headers

Tip: Press **Enter** to accept a value.

algorithms

hmac-sha1

 ×

hmac-sha256

 ×

hmac-sha384

 ×

hmac-sha512

 ×

Tip: Press **Enter** to accept a value.

✓ SUBMIT CHANGES

步骤2: 创建 Consumer

1. 进入 CONSUMERS 页面创建 Consumer。

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

Consumers

The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore.

+ CREATE CONSUMER

search...

Results: 25

USERNAME	CUSTOM_ID	TAGS	CREATED
----------	-----------	------	---------

版权所有：腾讯云计算（北京）有限责任公司

第181 共261页

- `username` : 用户（应用）名，必须指定此字段或 `custom_id`。
- `custom_id` : 用于将使用者映射到另一个数据库的自定义标识符。必须指定此字段或 `username`。
- `Tags` : 标签。

CREATE CONSUMER

`username`
(semi-optional)

test

The username of the consumer. You must send either this field or `custom_id` with the request.

`custom_id`
(semi-optional)

Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or `username` with the request.

`Tags`
(optional)

Optionally add tags to the consumer

✓ SUBMIT CONSUMER

2. 为 Consumer 创建 HMAC Auth 凭证。

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONSUMER: test

consumers / edit consumer

Details

Groups

Credentials

Accessible Routes

Plugins

BASIC

API KEYS

HMAC

OAuth2

JWT

<> HMAC Auth

+ CREATE CREDENTIALS

#	username	secret	created
1.	hmac-test	DO[REDACTED]	Apr 26, 2022 DELETE

- `username` : 在 HMAC 签名验证中使用的用户名。
- `secret` : secret 值，默认为空会自动生成 secret。

HMAC AUTH

Create HMAC credentials for **test**

username
(required)

test-hamd

The username to use in the HMAC Signature verification

secret
(optional)

The secret to use in the HMAC Signature verification

✓ SUBMIT

步骤3：签名认证方案

期望客户端使用以下参数化发送 Authorization 或 Proxy-Authorization header：

例如：

```
Authorization: hmac username="username", algorithm="hmac-sha1", headers="x-date digest", signature="Base64(HMAC-SHA1(signing_str, secret))"
```

签名参数

- `username`：凭证的 username。
- `algorithm`：用于创建签名的数字签名算法。
- `headers`：用于对请求进行签名的 HTTP 标头名称列表，由单个空格字符分隔。
- `signature`：客户端生成的 Base64 编码数字签名。

签名字符串构造

为了生成使用密钥签名的字符串，客户端必须按照它们出现的顺序获取 headers 指定的每个 HTTP 标头的值。

- 如果标题名称不是 request-line，则附加小写标题名称，后跟 ASCII 冒号：和 ASCII 空格。
- 如果标题名称是 request-line，则附加 HTTP 请求行（ASCII 格式），否则追加标题值。
- 如果 value 不是最后一个值，则附加 ASCII 换行符\n。字符串绝不能包含尾随的 ASCII 换行符。
- 服务器和请求客户端应与 NTP 同步，并且应使用 X-Date 或 Date header 发送有效日期（使用 GMT 格式，例如 Mon, 19 Mar 2018 12:08:40 GMT）。

正文验证 Body Validation

用户可以将 `config.validate_request_body` 设置为 `true` 以验证请求正文。如果启用，插件将计算请求正文的 SHA-256 HMAC 摘要，并将其与 Digest header 的值进行匹配。摘要 header 需要采用以下格式：

```
Digest: SHA-256=base64(sha256(<body>))
```

如果没有请求主体，则应将 Digest 设置为 0 长度的主体的摘要。

⚠ 注意：

为了创建请求主体的摘要，插件需要将其保留在内存中，这可能会在处理大型主体（几个 MB）或高请求并发期间对工作者的 Lua VM 造成压力。

步骤4：API 请求 Demo

❗ 说明：

以下 Demo 仅供参考，针对 POST 请求和 GET 请求，请依据实际使用场景进行修改。

Python 版本

```
# -*- coding: utf-8 -*-
import base64
import datetime
import hashlib
import hmac
import json
import requests
from urllib.parse import urlparse, urlencode

#username
Username = 'xxx'
#secret
Secret = 'xxxx'

# 访问地址
Url = 'http://test.com/'
HTTPMethod = 'POST' # method
Accept = 'application/json'
ContentType = 'application/json'

urlInfo = urlparse(Url)
Host = urlInfo.hostname
Path = urlInfo.path
```

```
Digest = ''

GMT_FORMAT = '%a, %d %b %Y %H:%M:%S GMT'

xDate = datetime.datetime.utcnow().strftime(GMT_FORMAT)

body_json = ''

# 修改 body 内容
if HTTPMethod == 'POST' :
    # 填写实际请求body
    body = { "arg1": "a", "arg2": "中文" }
    body_json = json.dumps(body)

body_digest = hashlib.sha256(body_json.encode()).digest()
Digest = "SHA-256=" + base64.b64encode(body_digest).decode()

# 获取签名串
signing_str = 'x-date: %s\ndigest: %s' % (
    xDate, Digest)

# 计算签名
sign = hmac.new(Secret.encode(), msg=signing_str.encode(),
    digestmod=hashlib.sha1).digest()
sign = base64.b64encode(sign).decode()
auth = "hmac username=\"" + Username + "\", algorithm=\"hmac-sha1\",
headers=\"x-date digest\", signature=\""
sign = auth + sign + "\""

# 发送请求
headers = {
    'Host': Host,
    'Accept': Accept,
    'Content-Type': ContentType,
    'x-date': xDate,
    'Authorization': sign,
    'Digest': Digest
}

if HTTPMethod == 'GET' :
    ret = requests.get(Url, headers=headers)
```

```
if HTTPMethod == 'POST' :  
    ret = requests.post(Url, headers=headers, data=(body_json))  
  
print(ret.headers)  
print(ret.text)
```

Java 版本

```
package org.example;  
  
import org.apache.http.HttpEntity;  
import org.apache.http.HttpResponse;  
import org.apache.http.client.methods.HttpPost;  
import org.apache.http.client.methods.HttpGet;  
import org.apache.http.entity.StringEntity;  
import org.apache.http.impl.client.CloseableHttpClient;  
import org.apache.http.impl.client.HttpClients;  
import org.apache.http.util.EntityUtils;  
  
import javax.crypto.Mac;  
import javax.crypto.spec.SecretKeySpec;  
import java.net.URI;  
import java.nio.charset.StandardCharsets;  
import java.text.SimpleDateFormat;  
import java.util.Base64;  
import java.util.Date;  
import java.util.Locale;  
import java.util.TimeZone;  
  
public class Main {  
    public static void main(String[] args) throws Exception {  
        String username = "xxxx";  
        String secret = "xxxx";  
        String url = "http://www.test.com/";  
        String httpMethod = "POST";  
        String accept = "application/json";  
        String contentType = "application/json";  
  
        URI uri = new URI(url);
```



```
String host = uri.getHost();

String digest = "";
SimpleDateFormat format = new SimpleDateFormat("EEE, dd MMM yyyy
HH:mm:ss z", Locale.US);
format.setTimeZone(TimeZone.getTimeZone("GMT"));
String xDate = format.format(new Date());
String body = "";

if (httpMethod.equals("POST")) {
    body = "arg1=a&arg2=中文";
}

byte[] bodyDigest =
java.security.MessageDigest.getInstance("SHA-
256").digest(body.getBytes(StandardCharsets.UTF_8));
digest = "SHA-256=" +
Base64.getEncoder().encodeToString(bodyDigest);
System.out.println(digest);

String signingStr = "x-date: " + xDate + "\ndigest: " + digest;
Mac mac = Mac.getInstance("HmacSHA1");
mac.init(new
SecretKeySpec(secret.getBytes(StandardCharsets.UTF_8), "HmacSHA1"));
byte[] signBytes =
mac.doFinal(signingStr.getBytes(StandardCharsets.UTF_8));
String sign = Base64.getEncoder().encodeToString(signBytes);
String auth = "hmac username=\"" + username + "\",
algorithm=\"hmac-sha1\", headers=\"x-date digest\", signature=\"";
sign = auth + sign + "\"";
HttpResponse response = null;
CloseableHttpClient httpClient = HttpClients.createDefault();
if (httpMethod.equals("POST")) {
    HttpPost httpPost = new HttpPost(url);
    httpPost.setHeader("Host", host);
    httpPost.setHeader("Accept", accept);
    httpPost.setHeader("Content-Type", contentType);
    httpPost.setHeader("x-date", xDate);
    httpPost.setHeader("Authorization", sign);
    if (!digest.isEmpty()) {
        httpPost.setHeader("Digest", digest);
    }
}
```

```
    }  
    StringEntity entity = new StringEntity("arg1=a&arg2=中文",  
"UTF-8");  
    httpPost.setEntity(entity);  
    response = httpClient.execute(httpPost);  
} else if (httpMethod.equals("GET")) {  
    HttpGet httpGet = new HttpGet(url);  
    httpGet.setHeader("Host", host);  
    httpGet.setHeader("Accept", accept);  
    httpGet.setHeader("Content-Type", contentType);  
    httpGet.setHeader("x-date", xDate);  
    httpGet.setHeader("Authorization", sign);  
    if (!digest.isEmpty()) {  
        httpGet.setHeader("Digest", digest);  
    }  
    response = httpClient.execute(httpGet);  
}  
  
HttpEntity responseEntity = response.getEntity();  
System.out.println(EntityUtils.toString(responseEntity));  
}  
}
```

相关说明

更多相关说明请参见 [Kong 插件](#)。

使用 ACL 访问控制

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍如何在 Kong 云原生网关上通过 Kong ACL 插件实现访问控制，它主要包括以下两种场景：

- 配置允许指定用户访问。
- 配置拒绝指定用户访问。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

操作步骤

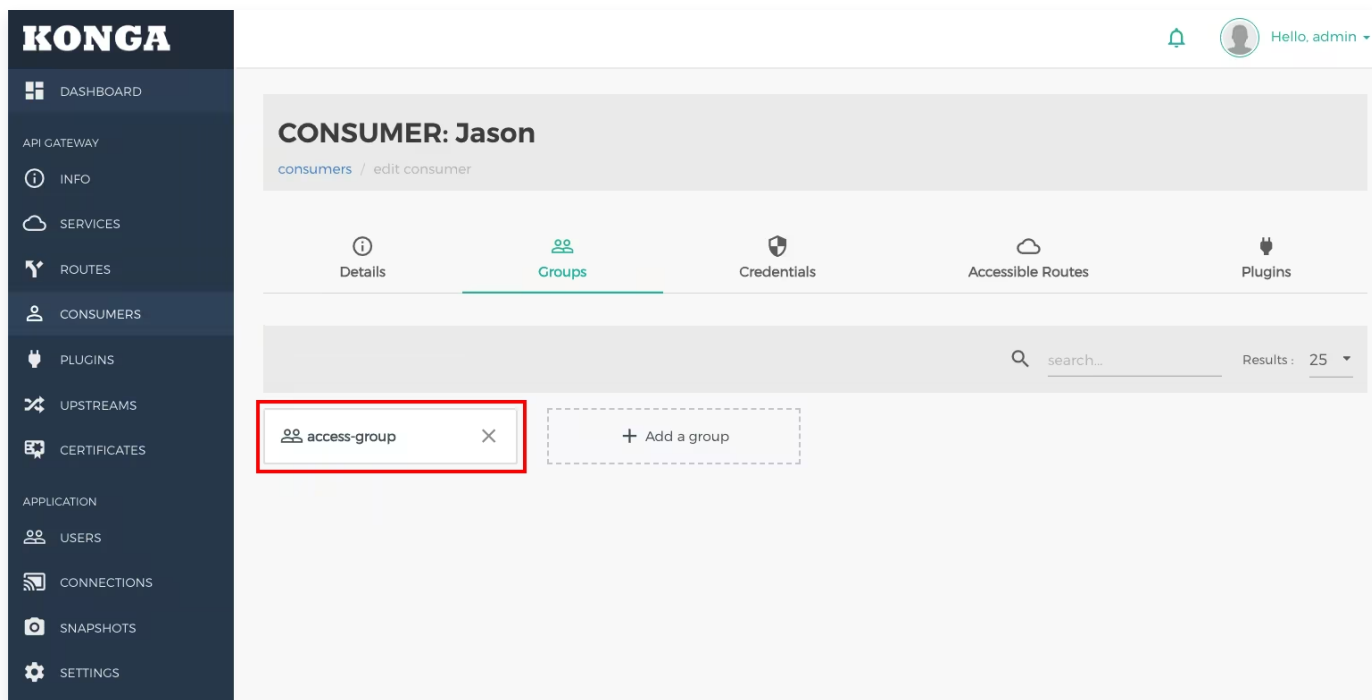
⚠ 注意

本操作步骤以允许指定用户访问的配置为例，拒绝指定用户访问的配置与之类似。

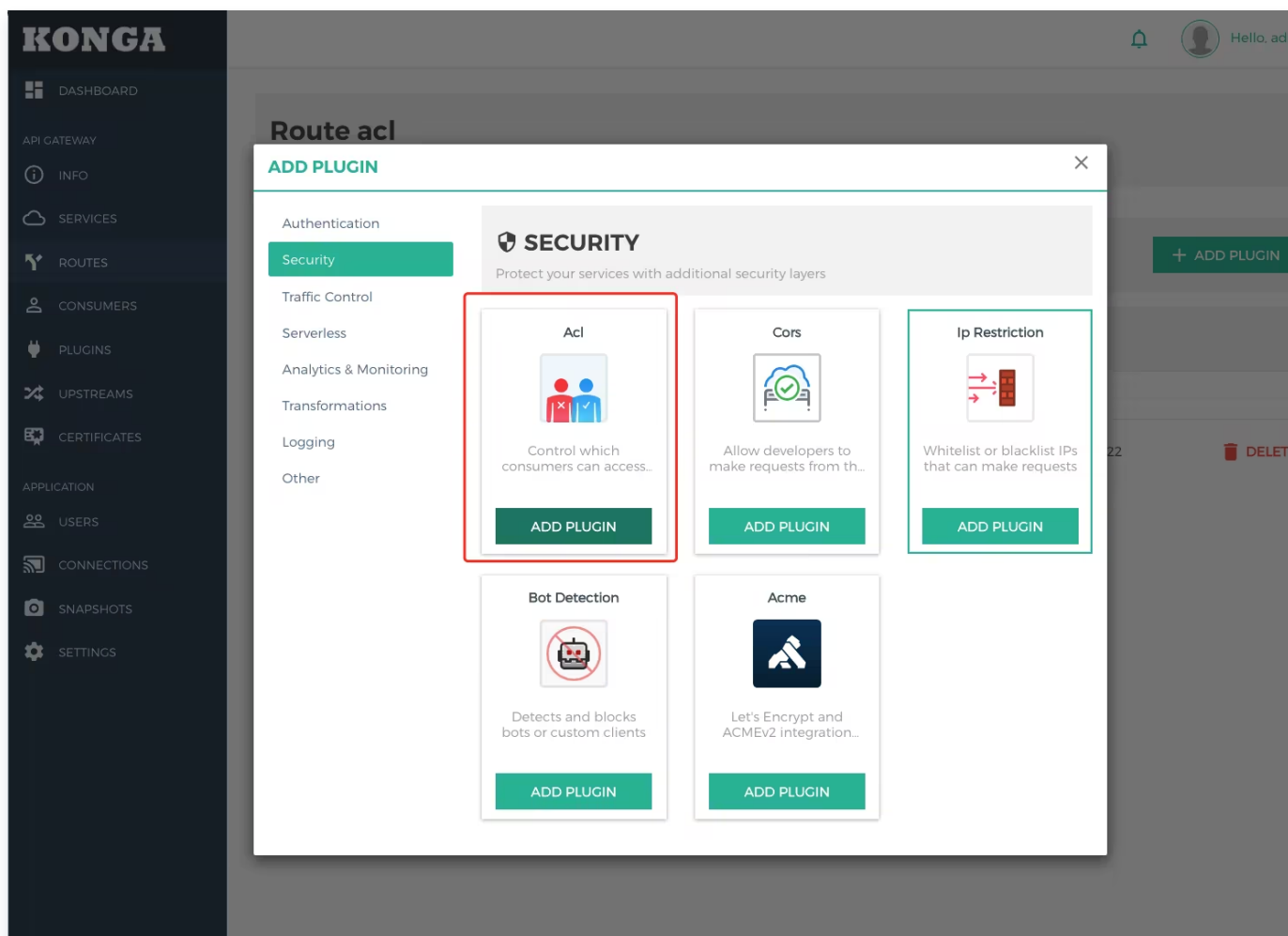
1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 登录到 Konga 控制台，进入 Consumer 详情页，选择需要配置访问控制的用户（例如 Jason），单击 Groups Tab，为该用户分配 Group（如 access-group）。



5. 进入选择的 Route，单击 **ADD PLUGIN**，在插件市场的 Security 分组中选择 ACL 插件，单击 **Add Plugin**。



6. 插件配置中填写需要配置允许访问的 Group 信息，按 Enter 键，并保存。

- `allow`：填写允许访问的 Group，多个 Group 需要填写多个。
- `deny`：填写拒绝访问的 Group，多个 Group 需要填写多个。
- `consumer`：填写需要应用访问控制的 Consumer ID，如空缺，表示该 IP 访问控制应用于所有 consumer。

⚠ 注意

allow 和 deny 两者至少配置一项。

ADD ACL ×

Restrict access to an API by whitelisting or blacklisting consumers using arbitrary ACL group names. This plugin requires an authentication plugin to have been already enabled on the API.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

allow

access-group ×

Tip: Press **Enter** to accept a value.

deny

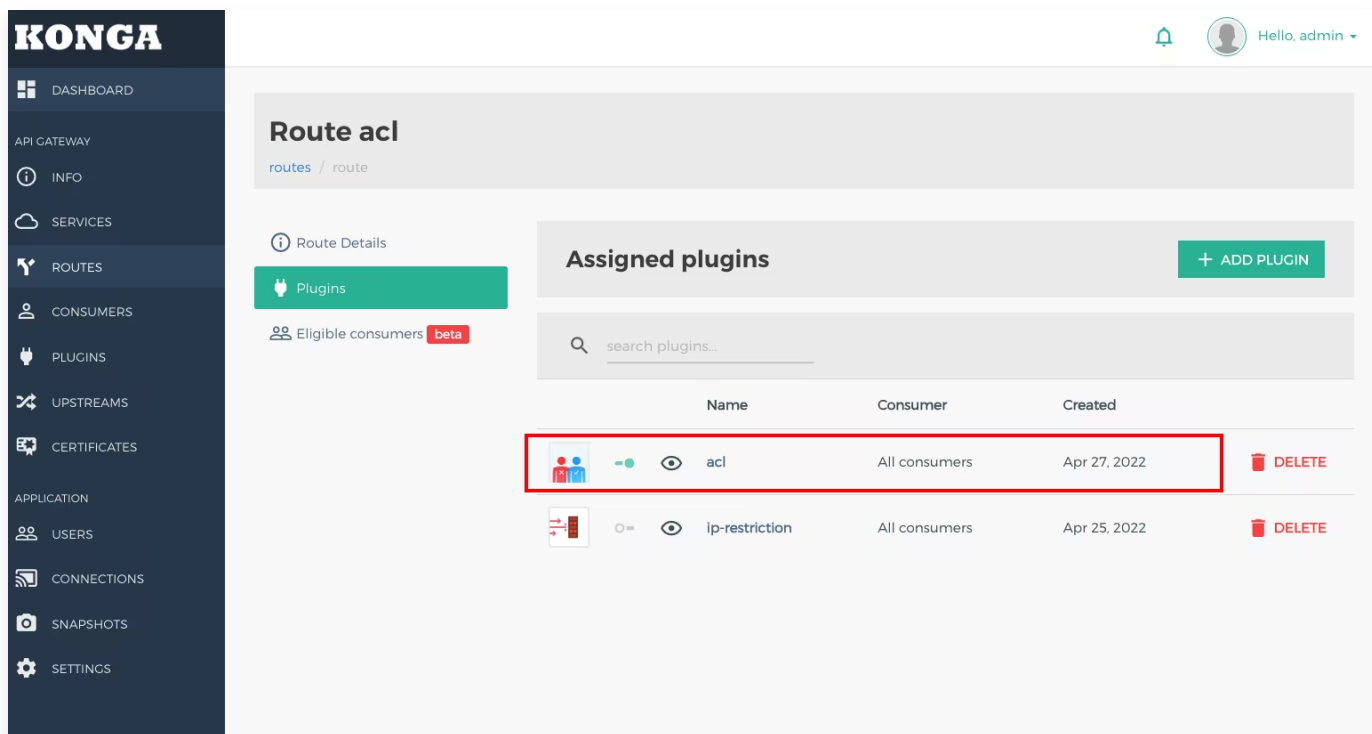
Tip: Press **Enter** to accept a value.

hide groups header

☐ NO

✓ ADD PLUGIN

7. 返回 route 页面，确定该 route 已经绑定的插件。



8. 使用允许访问用户组内的用户（如 Jason）的凭证发起 API 请求，该请求被允许。

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 13
Content-Type: text/plain; charset=UTF-8
Date: Wed, 27 Apr 2022 06:46:38 GMT
Server: Cowboy
Vary: Origin
X-Kong-Proxy-Latency: 10
X-Kong-Upstream-Latency: 1775

{"ok"}
```

9. 使用不在该用户组的用户（如 Tom）的凭证发起请求，该请求被拒绝。

```
HTTP/1.1 403 Forbidden
Connection: keep-alive
Content-Length: 49
Content-Type: application/json; charset=utf-8
Date: Wed, 27 Apr 2022 06:44:55 GMT
Server: kong/2.4.1
X-Kong-Response-Latency: 10
```

```
{  
  "message": "You cannot consume this service"  
}
```

相关说明

更多相关说明请参见 [Kong 插件](#)。

使用 Bot Detection 插件拒绝爬虫请求

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍如何在云原生网关上通过 Kong Bot Detection 插件屏蔽爬虫访问。

Bot Detection 插件通过检查 HTTP 请求中 User Agent 字段，识别发起请求的用户代理软件信息，拒绝爬虫请求。该插件内置了一些基础的校验规则用于校验请求，您可以参见 [爬虫示例](#) 定义爬虫请求。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

插件配置

字段名称	字段说明
allow	使用正则表达式定义允许放行的 User-Agent 列表。
deny	使用正则表达式定义拒绝放行的 User-Agent 列表。

操作步骤

- 登录 [TSF 控制台](#)。
- 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
- 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-1234567890-konga-gz.apigw.tencentcs.com:1337/>  关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-1234567890-konga-in.gz.apigw.tencentcs.com:1337/>  关闭

管理员账号

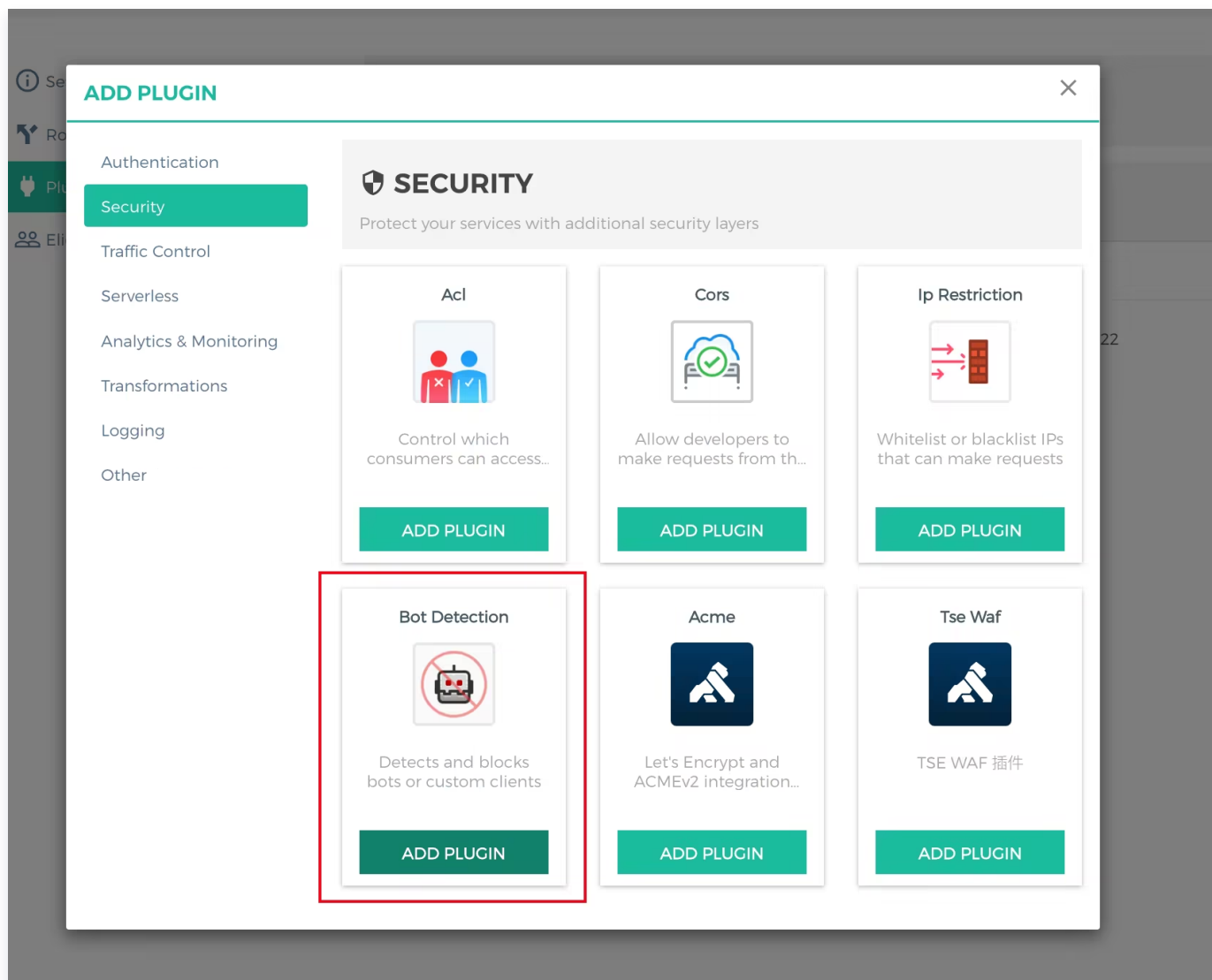
admin 

管理员密码

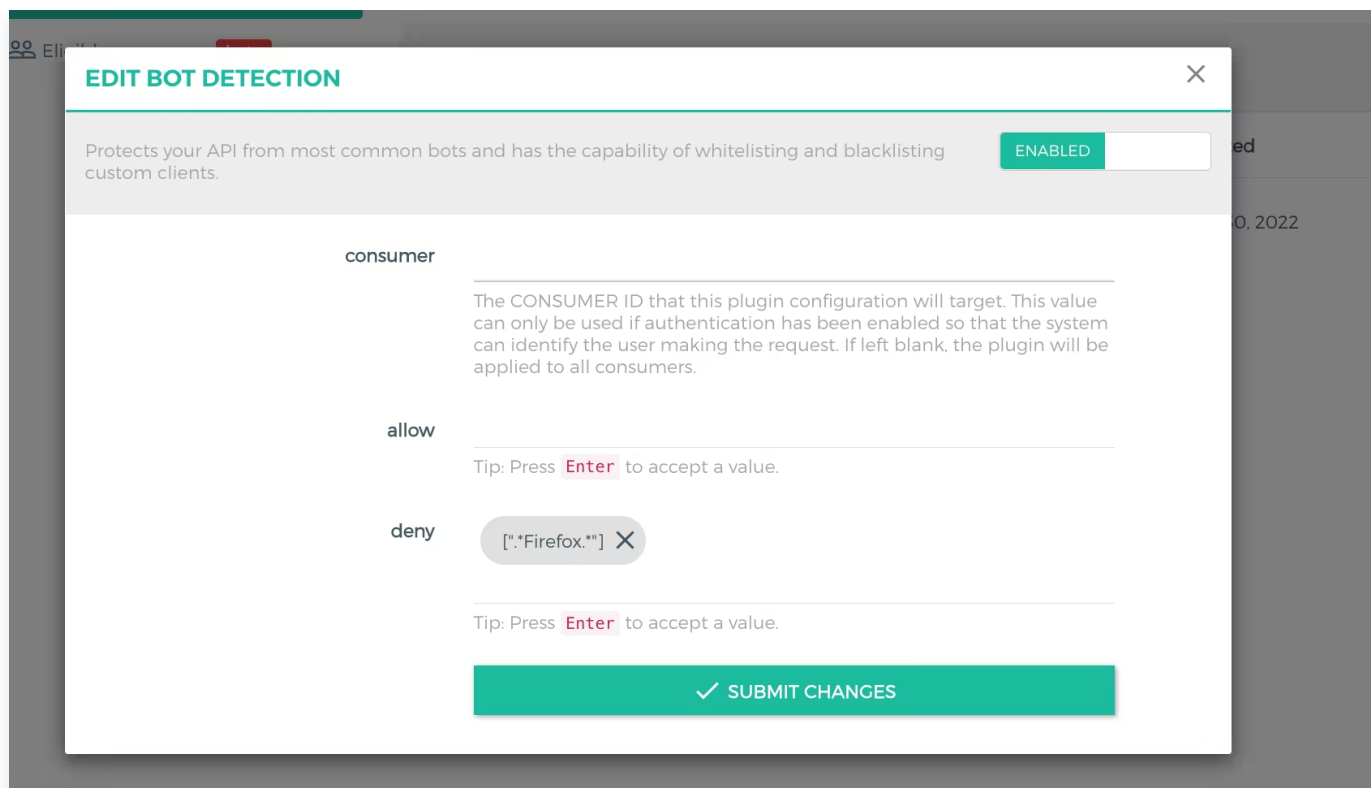
*****[显示](#) 

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

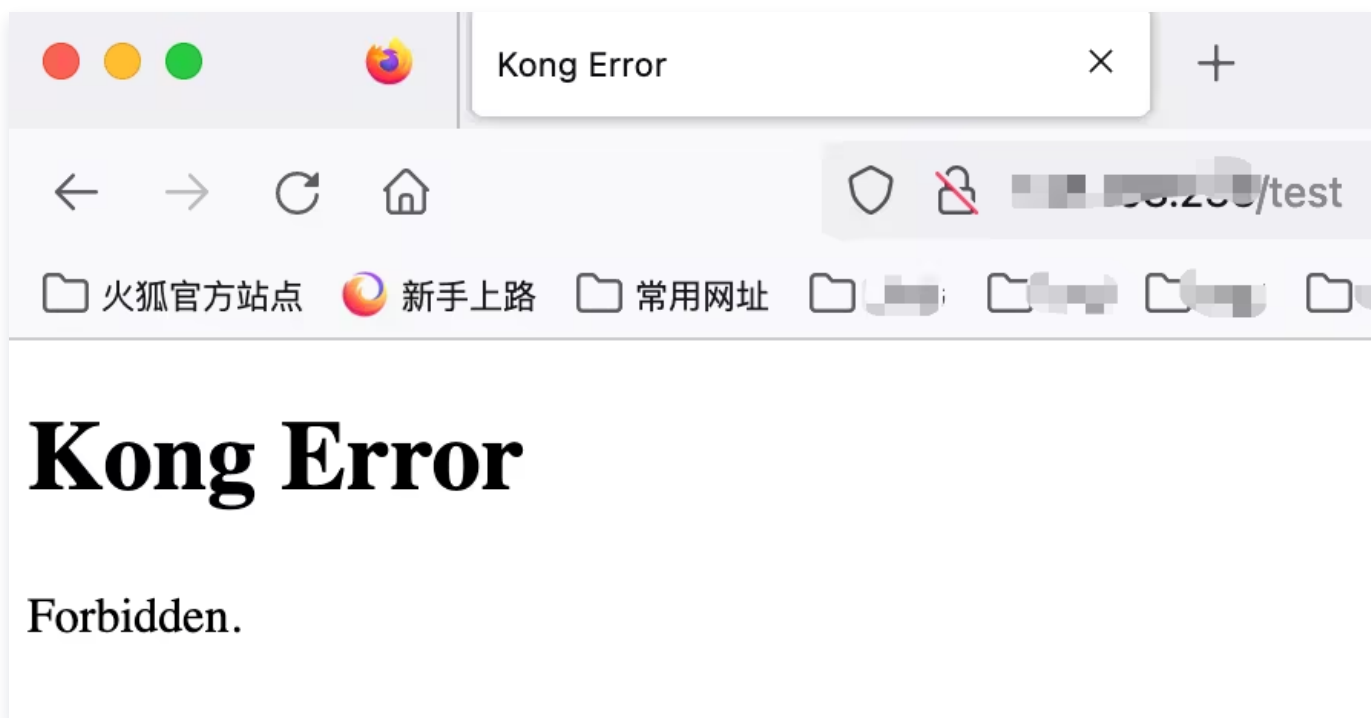
4. 登录 Konga 管理控制台，进入需要反爬虫的 Route 详情页，单击 **Add Plugin**，在插件市场的 Security 分组中选择 Bot Detection 插件，单击 **Add Plugin**。



5. 插件配置中使用正则表达式填写需要配置限制的代理信息，例如，拒绝来自 Firefox 的爬虫请求，在 deny 中配置 `[\".*Firefox.*\"]`，按 Enter 键。



6. 使用火狐浏览器发起 API 请求，由于请求头带有 `User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:106.0) Gecko/20100101 Firefox/106.0`，请求被拒绝。



参考

更多相关说明请参见 [Kong Bot Detection 插件官方文档](#)。

配置 JWT 认证鉴权

最近更新时间：2026-07-23 17:07:35

操作场景

JWT（JSON Web Token）是现代 Web 服务中经常用到的认证和授权方式。Kong 能够验证使用 HS256 或 RS256 签名的 JWT 的请求。您可以为消费者配置 JWT 凭据（公钥和私钥），并用这些凭据签署他们的 JWT。Kong 支持通过 Query、Cookie、Header 传递 JWT Token。如果 JWT 的签名通过验证，Kong 将把请求代理到上游服务，否则将丢弃请求。

JWT 常见的三种签名算法为 HS256、RS256 和 ES256，区别在于消息签名与签名验证需要的 key 不同。

- HS256 使用同一个 secret_key 进行签名和验证，适用于集中式认证场景，签名和验证都必须由可信方进行。
- RS256 是使用 RSA 私钥进行签名，使用 RSA 公钥进行验证，因此可以将验证委托给其他应用。
- ES256 使用私钥签名，公钥验证，但是拥有更短的签名长度。

本文介绍如何在 Kong 云原生网关上通过 JWT 插件实现下述常见 JWT 认证鉴权场景：

- 对使用 HS256 签名的请求进行认证鉴权。
- 对使用 RS256 签名的请求进行认证鉴权。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

插件配置

字段名称	字段说明
uri param names	用于获取 JWT Token 的 Query 参数列表，当 JWT Token 配置在 URI 时，设置此属性。
cookies names	用于获取 JWT Token 的 Cookie 参数列表，当 JWT Token 配置在 Cookie 时，设置此属性。
key claim name	Credential 的 key 值，对应的 JWT Claim 的内容，默认为 iss。
secret is base64	Credential 的 secret 是否进行 base64 编码。
claims to verify	Kong 进行校验的 Claims，支持 exp 或 nbf。

anonymous	认证失败时，判断是否作为匿名 Consumer。默认为空，表明请求失败时返回4xx。
run on preflight	插件是否应在 OPTIONS 预检请求上运行（并尝试进行身份验证）。如果设置为 false，则将始终允许 OPTIONS 请求。
maximum expiration	JWT Token 过期时间，支持0-31536000（365天）。当设置该值时，需要在 claims to verify 中指定 exp。默认值为0，表示无限期。当 Token 过期时请求将被拒绝，返回 HTTP 403。请注意，配置该值时应考虑潜在的时钟偏差。
header names	用于获取 JWT Token 的 Header 参数列表。当 JWT Token 配置在 URI 时，设置此属性。

操作步骤

场景一：对使用 HS256 签名的请求进行认证鉴权

步骤1：创建 Consumer

- 1. 登录 [TSF 控制台](#)。
- 2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
- 3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-4-1-1-konga-gz.apigw.tencentcs.com:1337/> 关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-4-1-1-konga-in.gz.apigw.tencentcs.com:1337/> 关闭

管理员账号

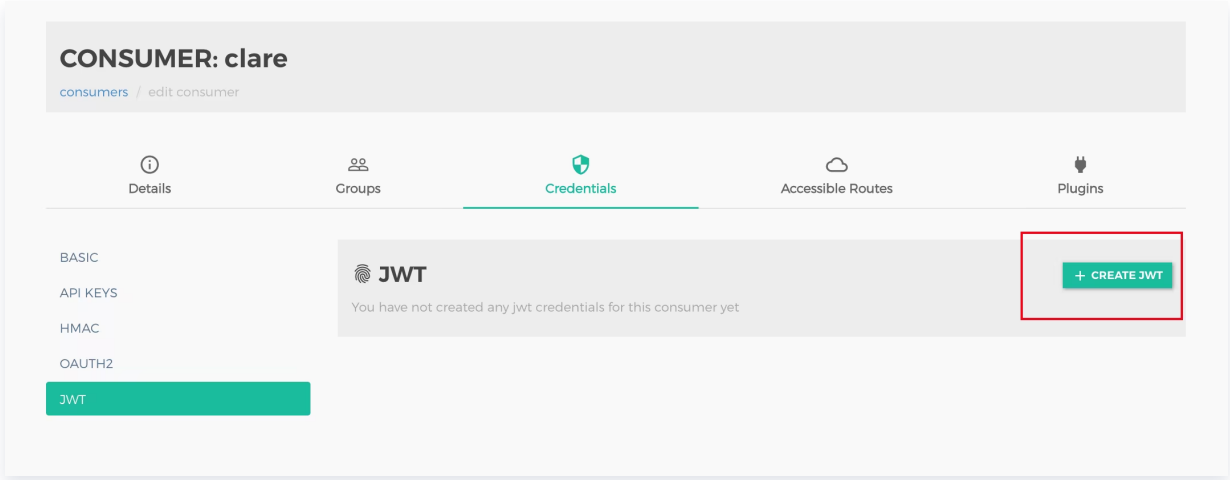
admin

管理员密码

*****[显示](#)

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

- 4. 登录到 Konga 控制台，进入 Consumer 详情页，选择需要配置访问控制的用户（如 clare），单击 **Credentials Tab**，选择 JWT，单击 **创建 JWT**，为该用户创建一个 JWT Token 作为访问凭证。



5. 配置 JWT 相关参数。

配置项	描述
key	对应 JWT Claims 的 issuer，即签发人。留空则自动生成。
algorithm	JWT Token 签名所使用的加密算法，支持 HS256 和 RS256。本场景设置为 HS256。
secret	当签名算法为 HS256 时，设置 secret，用于验证签名，留空则自动生成。

CREATE JWT

Create JWT for **clare****key***(optional)*

A unique string identifying the credential. If left out, it will be auto-generated. However, usage of this key is **mandatory** while crafting your token, as specified in the next section.

algorithm*(optional)*

HS256



The algorithm used to verify the token's signature. Can be **HS256** or **RS256**.

rsa_public_key*(optional)*

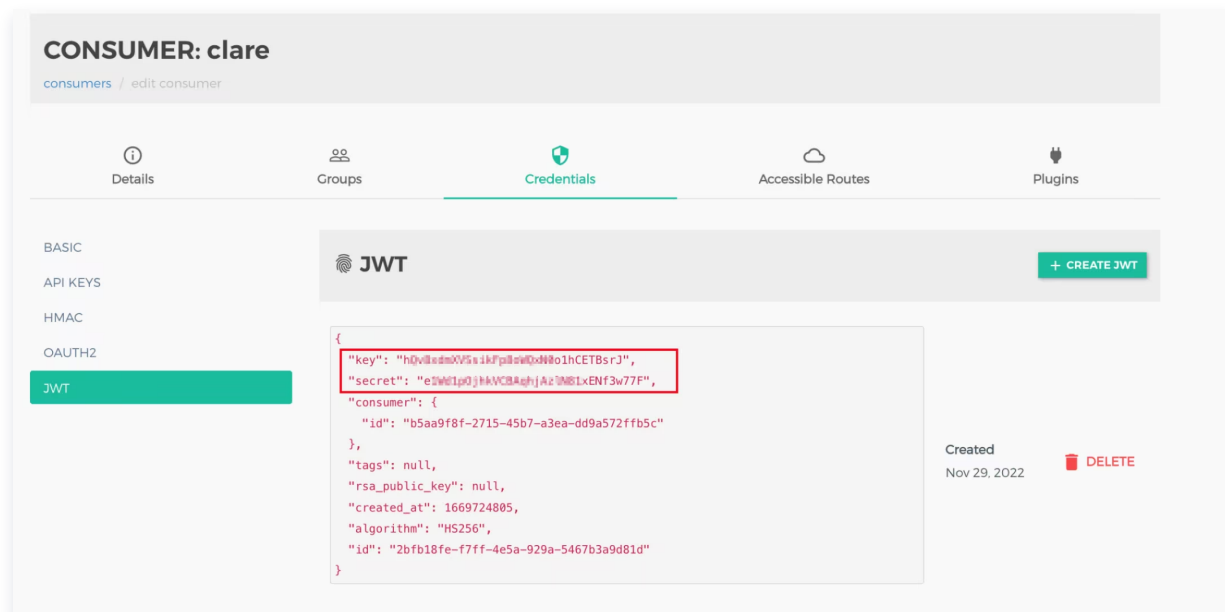
If **algorithm** is **RS256**, the public key (in PEM format) to use to verify the token's signature.

secret*(optional)*

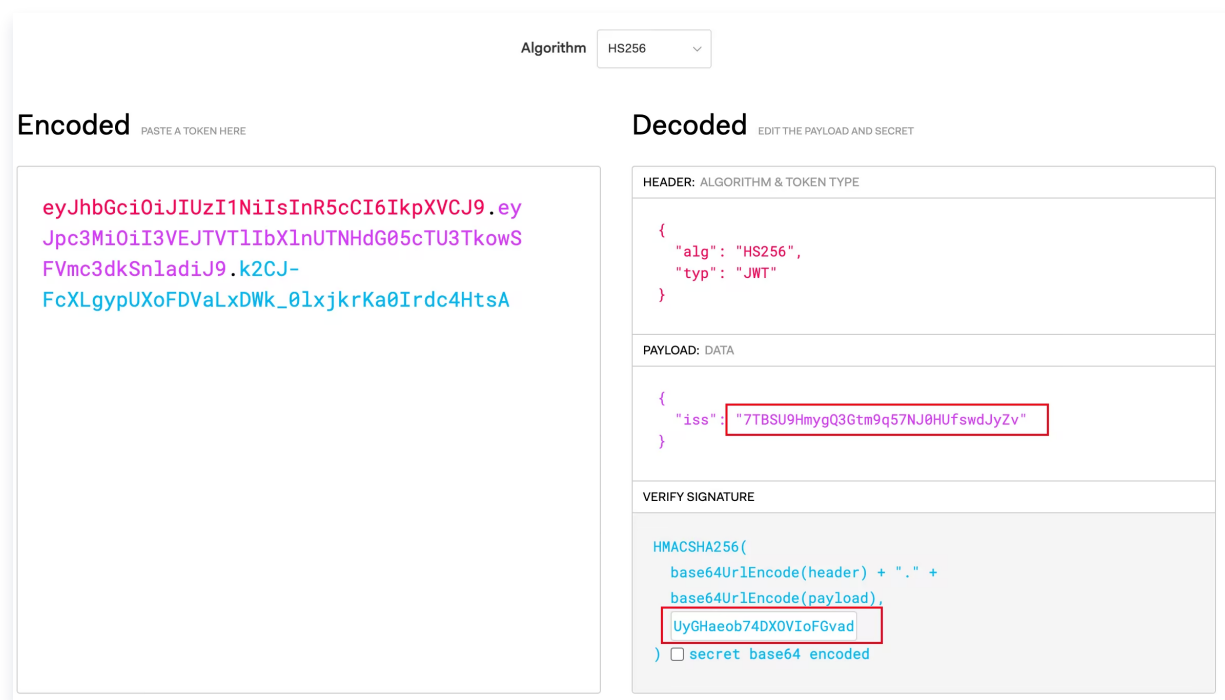
If **algorithm** is **HS256**, the secret used to sign JWTs for this credential. If left out, will be auto-generated. If **algorithm** is **RS256**, this is the private key (in PEM format) to use to verify the token's signature.

✓ SUBMIT

6. 查看生成的用户凭证。

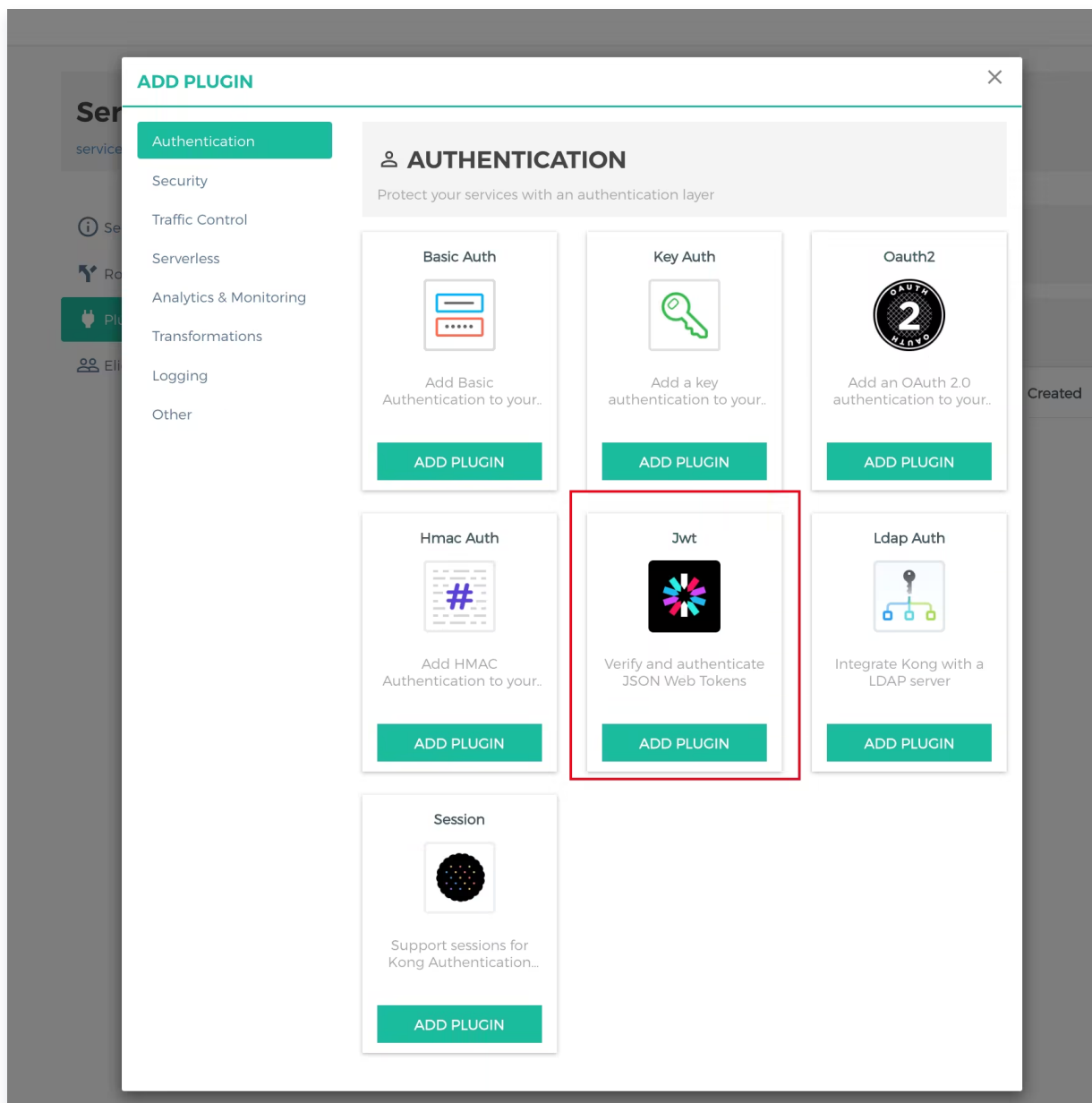


7. 使用程序生成对应的 JWT Token。



步骤2：绑定 JWT 插件

1. 进入需要认证的 Service，单击 **ADD PLUGIN**，在插件市场的 Authentication 分组中选择 JWT 插件，单击 **Add Plugin**。



2. 填写 JWT Token 校验的信息，单击 **ADD PLUGIN**。

ADD JWT

Verify requests containing HS256 or RS256 signed JSON Web Tokens (as specified in RFC 7519). Each of your Consumers will have JWT credentials (public and secret keys) which must be used to sign their JWTs. A token can then be passed through the Authorization header or in the request's URI and Kong will either proxy the request to your upstream services if the token's signature is verified, or discard the request if not. Kong can also perform verifications on some of the registered claims of [RFC 7519](#) (exp and nbf).

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

uri param names

Tip: Press **Enter** to accept a value.

A list of querystring parameters that Kong will inspect to retrieve JWTs.

cookie names

Tip: Press **Enter** to accept a value.

key claim name

iss

The name of the claim in which the key identifying the secret must be passed.

secret is base64

NO

If true, the plugin assumes the credential's secret to be base64 encoded. You will need to create a base64 encoded secret for your consumer, and sign your JWT with the original secret.

claims to verify

Tip: Press **Enter** to accept a value.

A list of registered claims (according to RFC 7519) that Kong can verify as well. Accepted values: exp, nbf.

anonymous

run on preflight

YES

maximum expiration

0

header names

Tip: Press **Enter** to accept a value.

✓ ADD PLUGIN

步骤3：测试请求

1. 不携带 JWT Token 请求服务，该请求被拒绝，返回401。

```
{ "message": "Unauthorized" }
```

2. 携带 JWT Token 请求服务，请求成功。

hello Kong

场景二：对使用 RS256 签名的请求进行认证鉴权

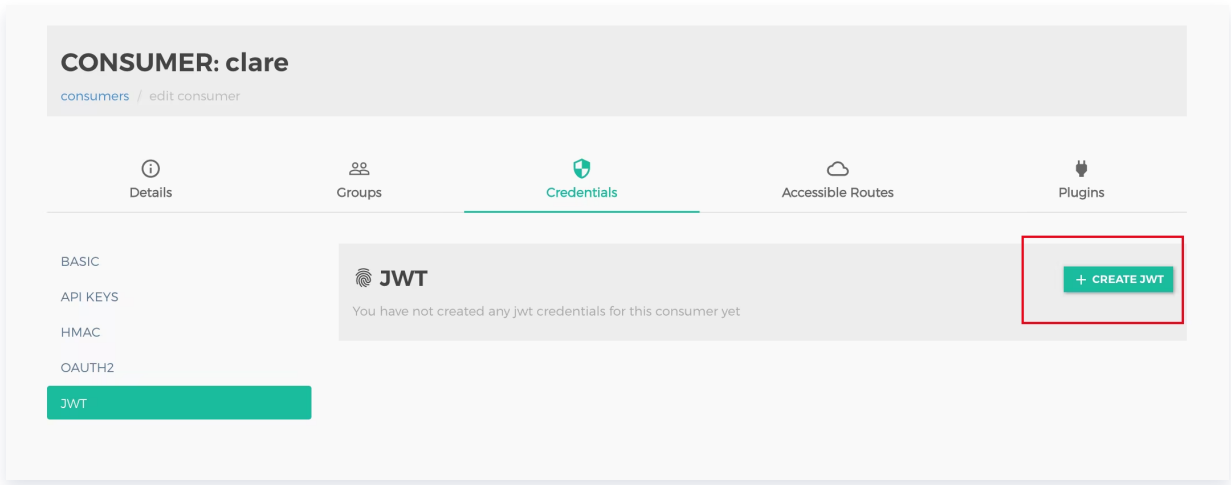
步骤1: 创建 Consumer

1. 登录 TSF 控制台。

2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 登录到 Konga 控制台，进入 Consumer 详情页，选择需要配置访问控制的用户（如 clare），单击 Credentials Tab，选择 JWT，单击 创建 JWT，为该用户创建一个 JWT Token 作为访问凭证。



5. 配置 JWT 相关参数。

配置项	描述
key	对应 JWT Claims 的 issuer，即签发人。留空则自动生成。

algorithm	JWT Token 签名所使用的加密算法，支持 HS256 和 RS256。本场景设置为 RS256。
rsa_public_key	当签名算法为 RS256 时，设置公钥（PEM 格式），用于验证签名。
secret	当签名算法为 RS256 时，设置私钥（PEM 格式），用于验证签名。

CREATE JWT

Create JWT for **clarersa**

key
(optional)
A unique string identifying the credential. If left out, it will be auto-generated. However, usage of this key is **mandatory** while crafting your token, as specified in the next section.

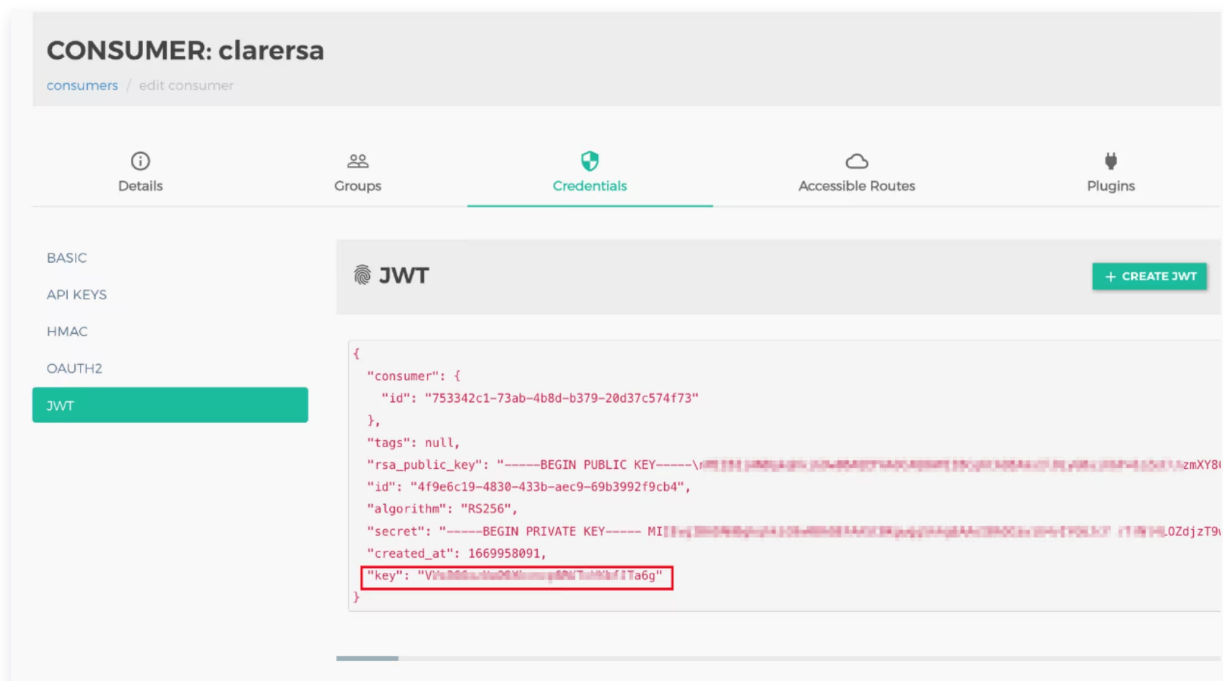
algorithm
(optional)
RS256
The algorithm used to verify the token's signature. Can be **HS256** or **RS256**.

rsa_public_key
(optional)
2QIDAQAB
-----END PUBLIC KEY-----
If **algorithm** is **RS256**, the public key (in PEM format) to use to verify the token's signature.

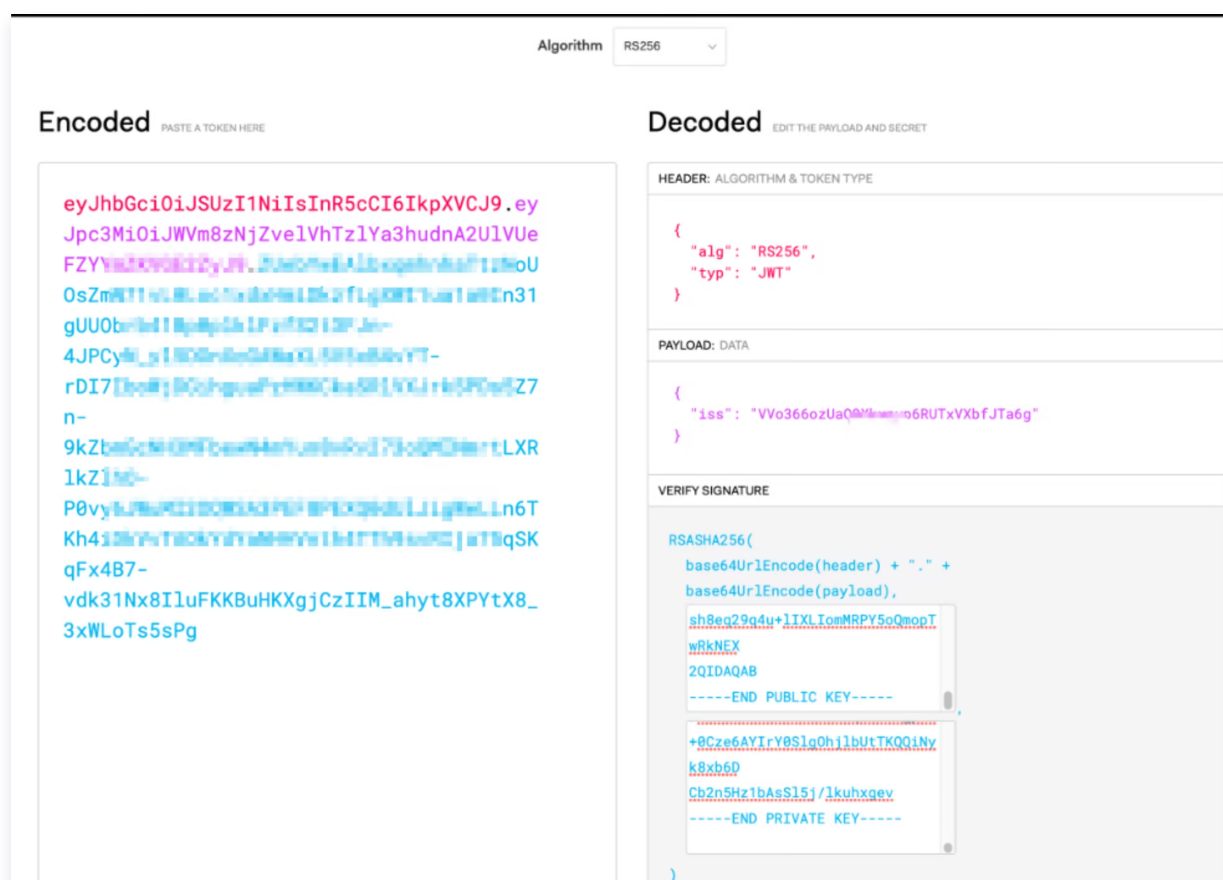
secret
(optional)
sSl5j/lkuhxgev -----END PRIVATE KEY-----
If **algorithm** is **HS256**, the secret used to sign JWTs for this credential. If left out, will be auto-generated. If **algorithm** is **RS256**, this is the private key (in PEM format) to use to verify the token's signature.

✓ SUBMIT

6. 查看生成的用户凭证。

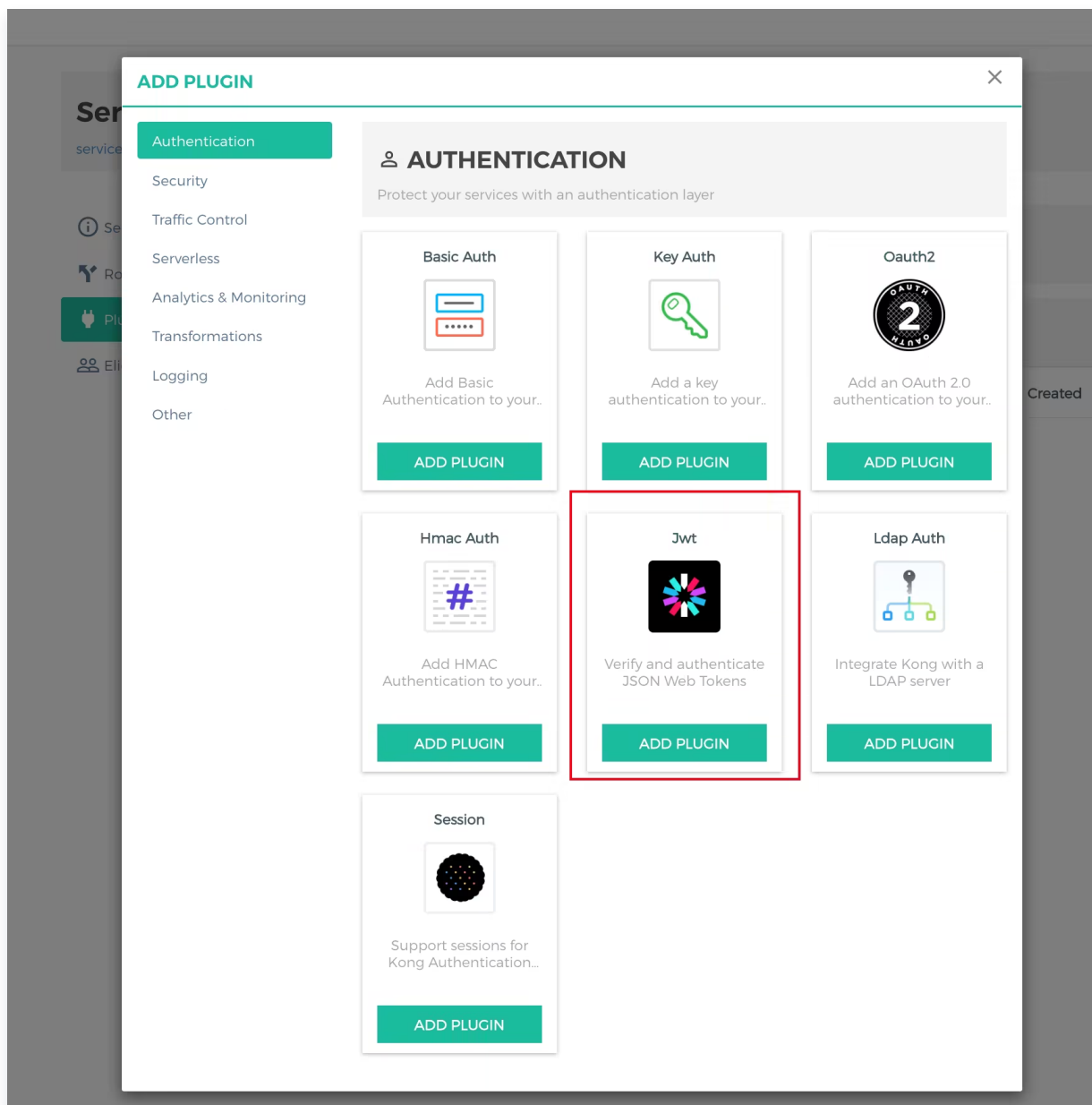


7. 生成对应的 JWT Token。



步骤2：绑定 JWT 插件

1. 进入需要认证的 Service，单击 **ADD PLUGIN**，在插件市场的 Authentication 分组中选择 JWT 插件，单击 **Add Plugin**。



2. 填写 JWT Token 校验的信息，单击 **ADD PLUGIN**。

ADD JWT

Verify requests containing HS256 or RS256 signed JSON Web Tokens (as specified in RFC 7519). Each of your Consumers will have JWT credentials (public and secret keys) which must be used to sign their JWTs. A token can then be passed through the Authorization header or in the request's URI and Kong will either proxy the request to your upstream services if the token's signature is verified, or discard the request if not. Kong can also perform verifications on some of the registered claims of [RFC 7519](#) (exp and nbf).

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

uri param names

Tip: Press **Enter** to accept a value.

A list of querystring parameters that Kong will inspect to retrieve JWTs.

cookie names

Tip: Press **Enter** to accept a value.

key claim name

iss

The name of the claim in which the key identifying the secret must be passed.

secret is base64

NO

If true, the plugin assumes the credential's secret to be base64 encoded. You will need to create a base64 encoded secret for your consumer, and sign your JWT with the original secret.

claims to verify

Tip: Press **Enter** to accept a value.

A list of registered claims (according to RFC 7519) that Kong can verify as well. Accepted values: exp, nbf.

anonymous

run on preflight

YES

maximum expiration

0

header names

Tip: Press **Enter** to accept a value.

✓ ADD PLUGIN

步骤3：测试请求

1. 不携带 JWT Token 请求服务，该请求被拒绝，返回401。

```
{ "message": "Unauthorized" }
```

2. 携带 JWT Token 请求服务，请求成功。

参考

更多相关说明请参见 [Kong JWT 插件官方文章](#)。

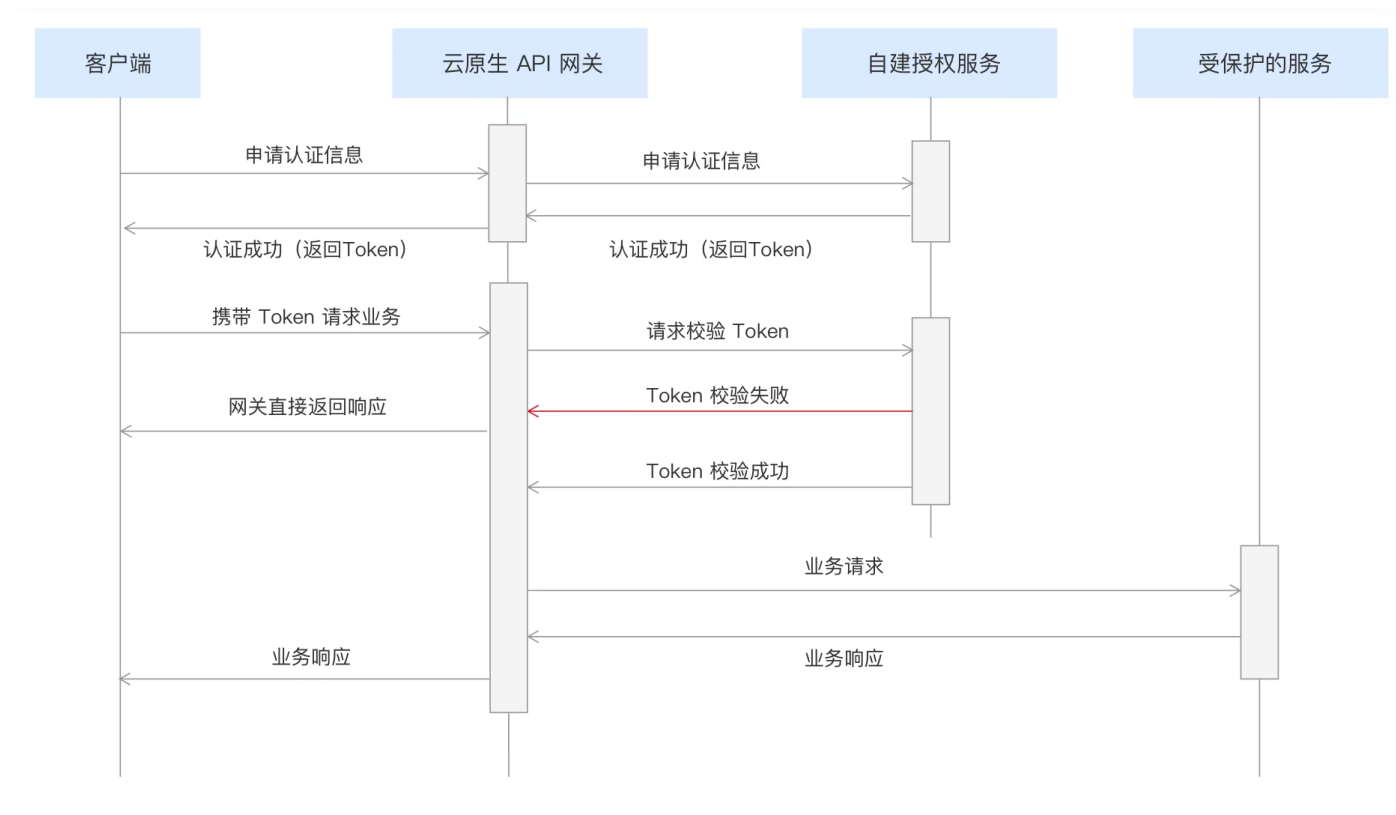
配置自定义鉴权

最近更新时间：2026-07-23 17:07:34

功能介绍

云原生网关提供自定义鉴权的能力，方便实现在网关入口进行统一鉴权。当客户端请求携带的凭证信息（Token）为自定义的格式时，网关可根据配置将请求转发到中心化的鉴权服务来完成鉴权操作，并根据鉴权结果转发/拒绝请求，以保护 API 接口的通信安全。

云原生网关自定义鉴权流程如下：



云原生网关自定义鉴权提供如下能力：

- 支持配置在网关服务或路由上，以针对特定路由或特定服务下所有路由进行自定义鉴权。
- 支持配置鉴权服务不可用策略，包括直接放通和直接拒绝。
- 支持配置路由白名单，简化使用。即服务下某些白名单路由将不进行鉴权，其余路由进行鉴权。
- 支持配置鉴权结果返回策略，包括通过 HTTP 状态码标识和通过响应头标识。
- 鉴权请求支持携带 Header 和 body。
- 鉴权响应支持保留 Header 和 body（通过响应头传递）。
- 支持未携带 Token 请求时跳转到重定向地址进行 token 获取。

插件配置说明

字段名称	字段含义	说明
description	插件描述	—
auth service id	鉴权服务 ID	进入 云原生网关控制台 ，单击目标实例 ID 进入实例详情页，单击左侧服务路由，单击服务，在服务列表中可直接复制服务 ID
auth service path	鉴权 API 路径	—
token key name	存放 Token 的请求头名称	—
route id whitelist	路由白名单，路由 ID 列表不进行鉴权。	—
timeout	超时时间：鉴权服务超时时间，默认 10 秒。	—
auth server unavailable policy	鉴权服务不可用策略，当鉴权服务不可用时，网关处理客户端请求的策略。支持直接放通（allow）和直接拒绝（deny）两种。	- 直接放通（allow）：当鉴权服务不可用时，网关直接放通客户端请求到后端服务。 - 直接拒绝（deny）：当鉴权服务不可用时，网关拒绝客户端请求，返回403状态码，返回 Header 标识鉴权服务不可用。
redirect path	请求未携带 Token 时的重定向路径	网关返回 302 进行重定向，重定向 URL 拼接规则为： <code>{请求scheme}://{请求Host}/{redirect_path}</code>
allow request body	鉴权请求中是否允许携带 Body	—
allow request headers	鉴权请求允许携带的 Header 列表	—
auth result identifier	鉴权结果返回策略，支持通过 HTTP 状态码标识（HTTP_status_code）或通过响应头（response_header）返回两种	- HTTP 状态码标识（HTTP_status_code）：鉴权服务返回 200 时，鉴权成功；其余均认为鉴权失败。 - 通过响应头返回（response_header）：鉴权服务器到网关返回 HTTP 200 ,通过自定义响应头标识鉴权结果，当指定响应对应值为 true 时，鉴权成功，响应头对应值为 false 或空时，鉴权失败。

auth result response header	鉴权结果自定义响应头	—
auth body header	鉴权响应 Body 所属响应头	支持将鉴权服务返回 Body 存放在自定义响应头中传回客户端。
allow response headers	鉴权响应中允许携带的 Header 列表	—

使用场景

前提步骤

1. 创建需要访问的后端服务 nginx 和路由。
2. 新建鉴权服务 auth_service，根据鉴权服务信息，填写服务类型、地址列表、请求协议。

新建服务

服务名称

auth_service

服务类型

K8S 服务

注册中心

IP 列表

域名/IP

私有域名

云函数

地址列表

IP	端口 ⓘ	权重
IP		

添加节点

CVM 弹性伸缩 ⓘ

☐ 开启

服务协议 ⓘ

☒ HTTP☐ HTTPS☐ TCP☐ UDP

服务路径

请输入服务路径

超时时间

-

60000

+

毫秒

重试次数

-

5

+

高级配置

负载均衡算法

轮询

慢启动

☐ 开启

开启后，服务对应的节点将在慢启动时间内，将权重从1逐步增加到目标值

确定

取消

3. 单击服务 ID，单击服务信息页签查看服务信息

路由管理 / 服务 / auth_service

路由管理灰度策略限流策略服务信息

基本信息

服务 ID

df-7bec

服务名称

auth-service

创建时间

2023-07-27 11:28:20

标签

TSE-Service-Type:IPListTSE-Upstream-SlowStart:0***

服务信息

服务类型

IP 列表

请求协议

HTTP

请求路径

/api/v1/auth/ident

弹性伸缩

未开启

超时时间

60000 毫秒

重试次数

5 次

负载均衡算法

轮询

慢启动

未开启

慢启动时间

0 秒

节点信息

节点地址	权重	健康状况	创建时间
42.80	100	健康	2023-07-27 11:28:20

4. 安装自定义鉴权插件

进入插件管理，选择系统插件 > tse-custom-auth，单击安装最新版本即可安装。

场景一：鉴权结果通过 HTTP 状态码标识

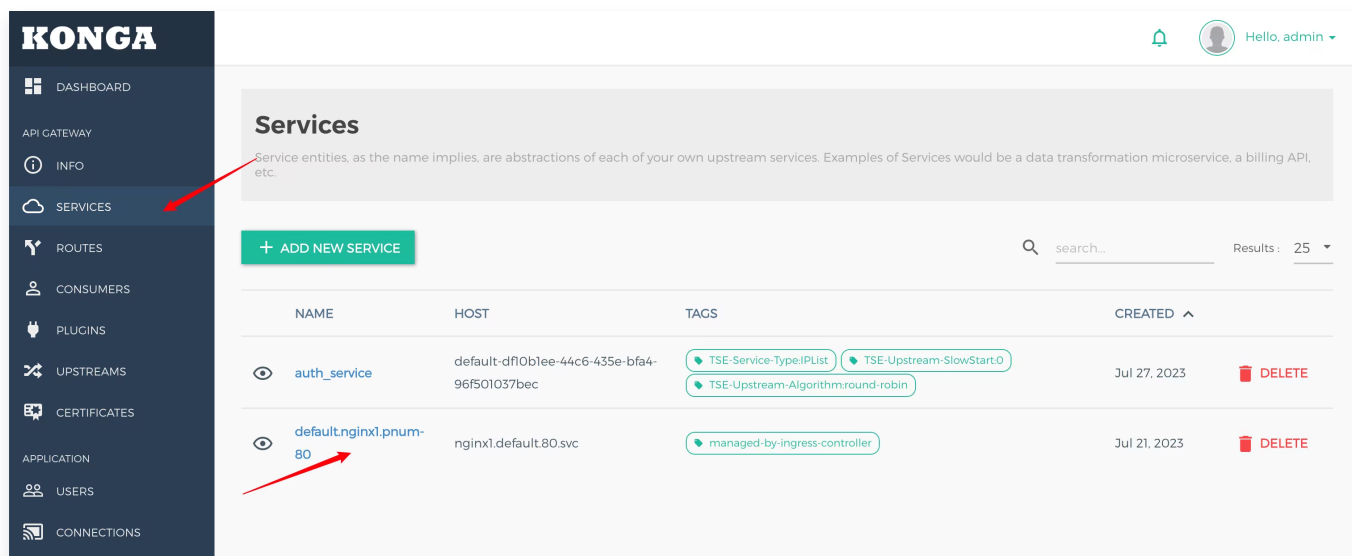
场景说明：当您的鉴权服务器使用 HTTP 状态码标识鉴权结果时，可配置 auth result identifier 为 HTTP status code，网关将根据鉴权服务器返回的状态码判定鉴权结果，当状态码为 200 时认为鉴权成功，网关将转发请求到后端服务；其余情况则认为鉴权失败，拒绝请求。

操作步骤：

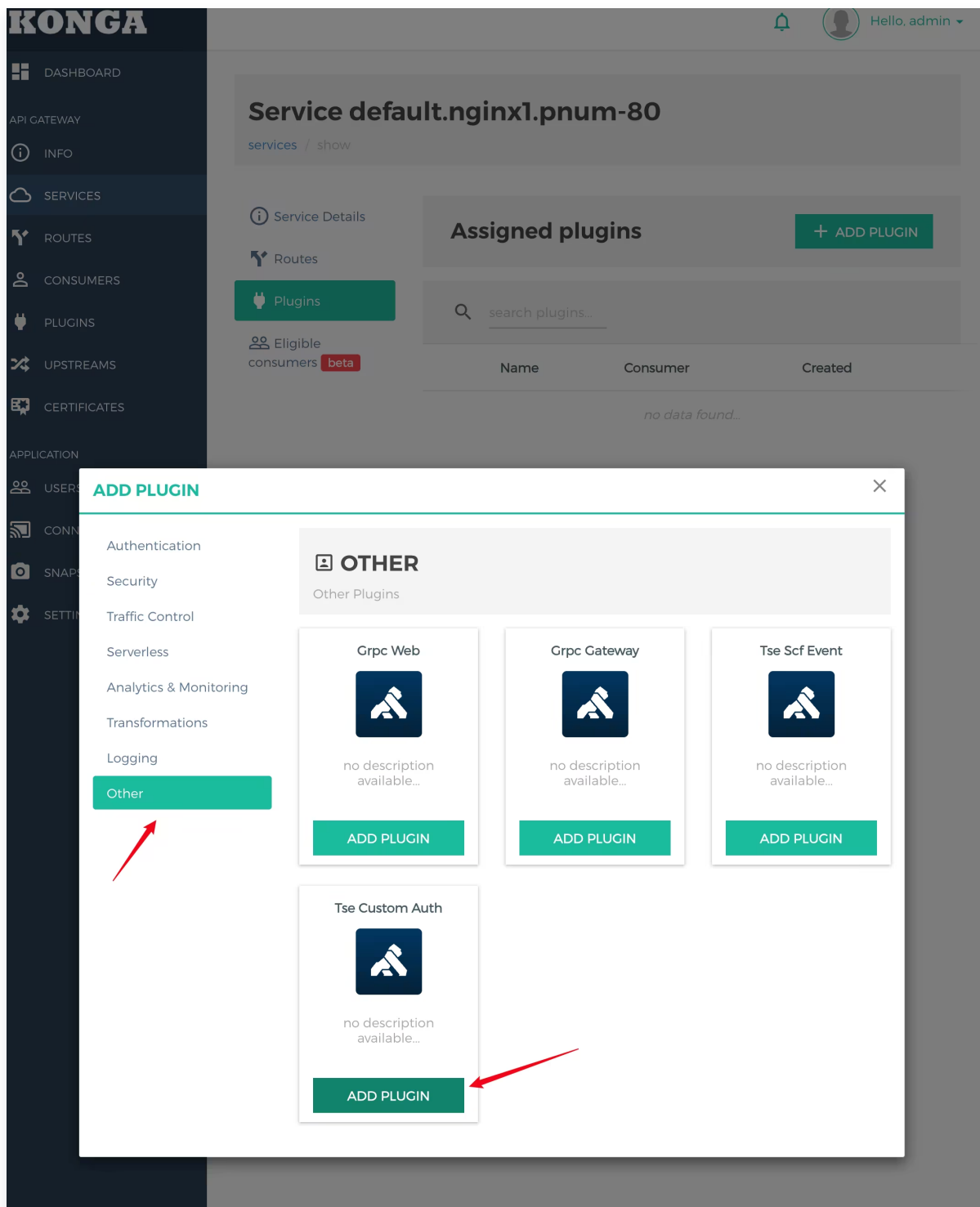
1. 登录 TSF 控制台。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 单击左侧 Service 页面，单击 nginx 服务进入详情。



5. 单击 Plugins > +ADD Plugin 添加自定义鉴权插件



6. 填写插件配置，单击 ADD PLUGIN，设置鉴权结果返回策略为 HTTP 状态码。

ADD TSE CUSTOM AUTH

×

Configure the Plugin.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

description

auth service id

7b833ed6-108e-4361-8470-38a5da15dbe0

auth service path

/oauth/currentUserForTencent

token key name

Authorization

route id whitelist

Tip: Press **Enter** to accept a value.

timeout

10000

auth server unavailable policy

allow

redirect path

allow request body

☐ NO

allow request headers

Tip: Press **Enter** to accept a value.

auth result identifier

HTTP_status_code

auth result response header

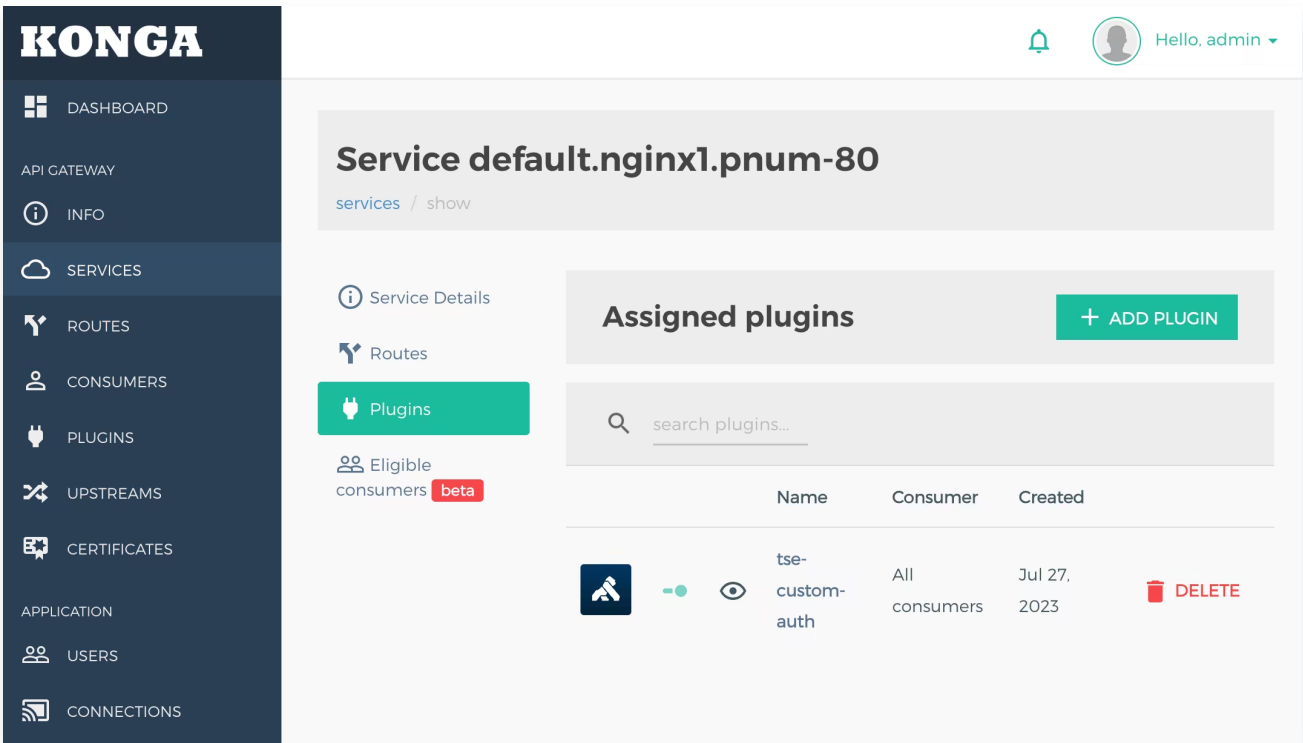
auth body header

allow response headers

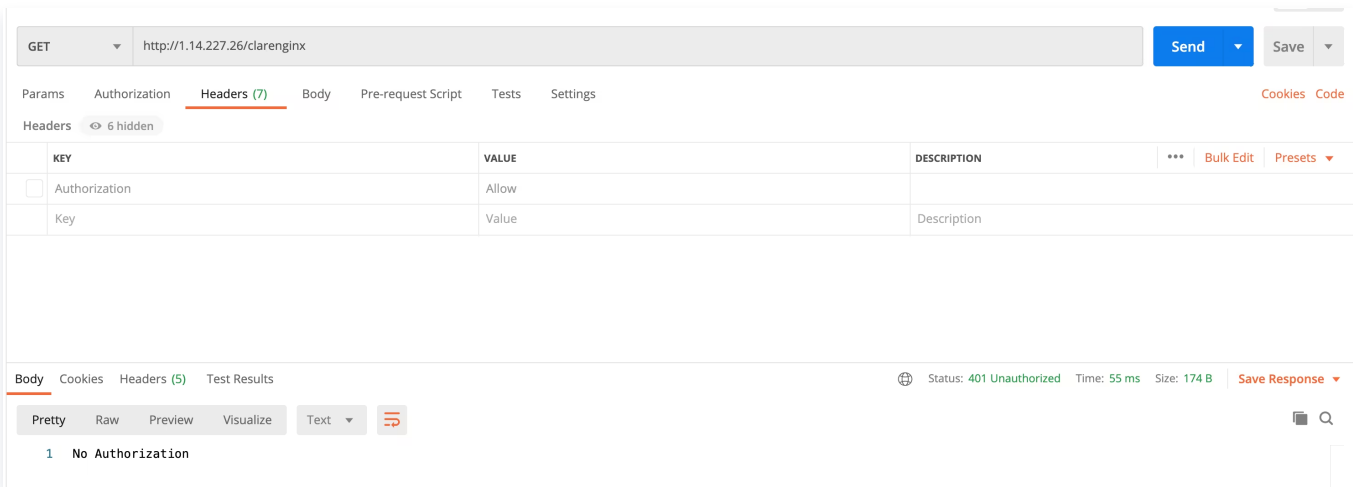
Tip: Press **Enter** to accept a value.

✓ ADD PLUGIN

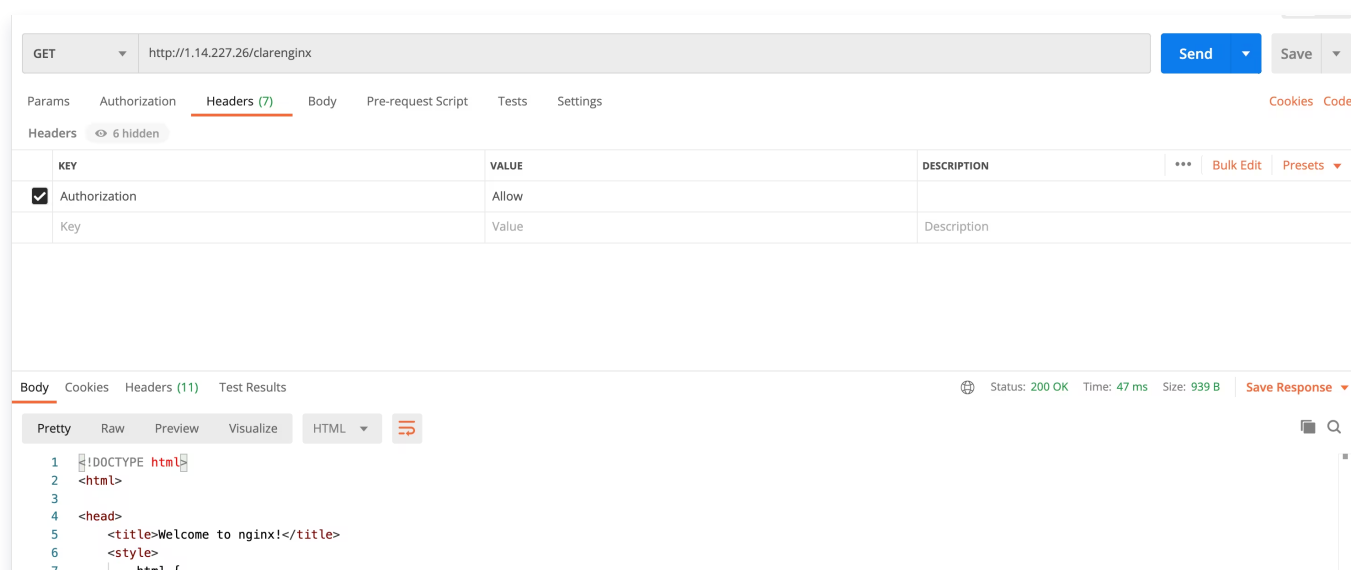
7. 确认插件已绑定后端服务。



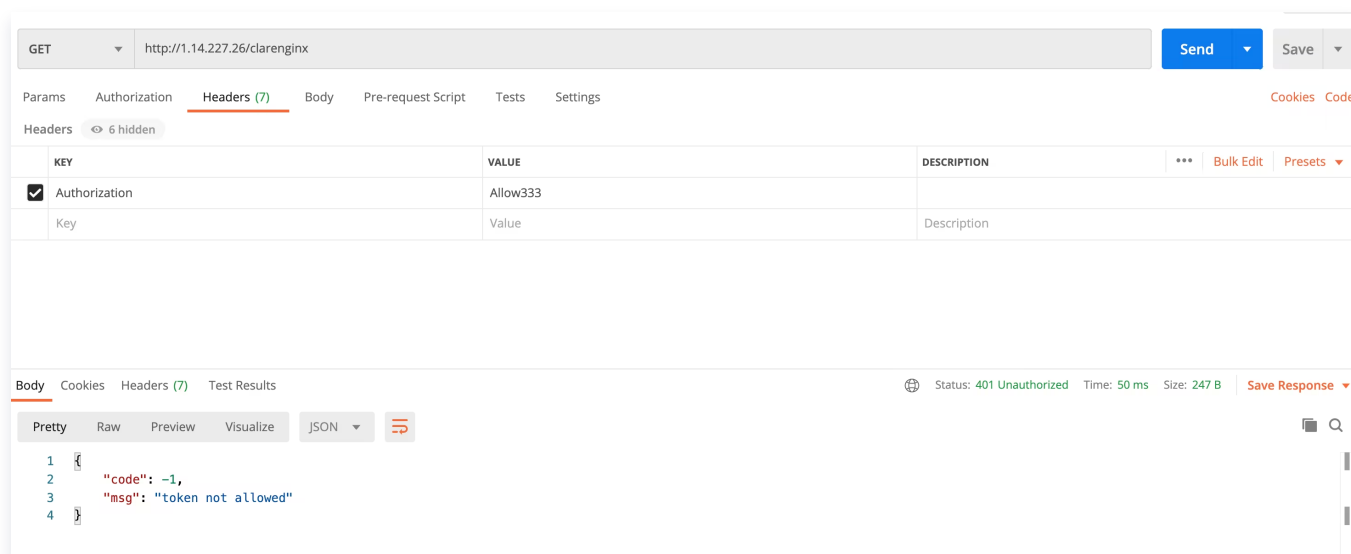
8. 未携带 Token 访问后端服务，访问被拒绝，返回 401。



9. 携带正确的 Token 访问后端服务，成功访问后端服务。



10. 携带错误的 Token 访问后端服务，访问被拒绝，返回 401。

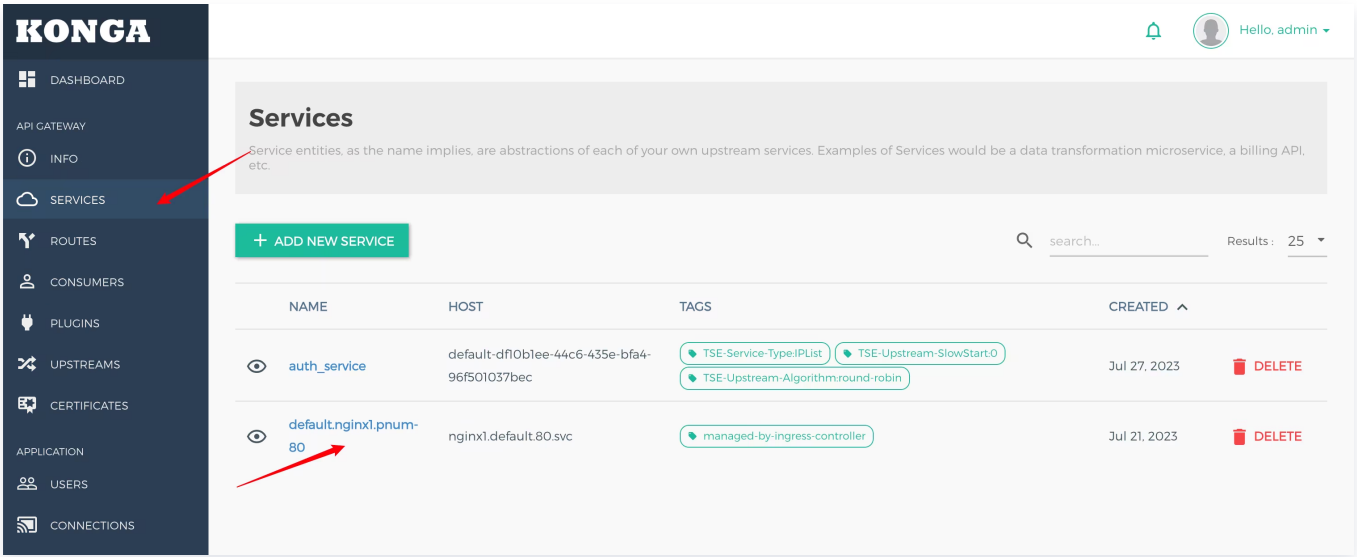


场景二：鉴权结果通过响应头返回失败信息

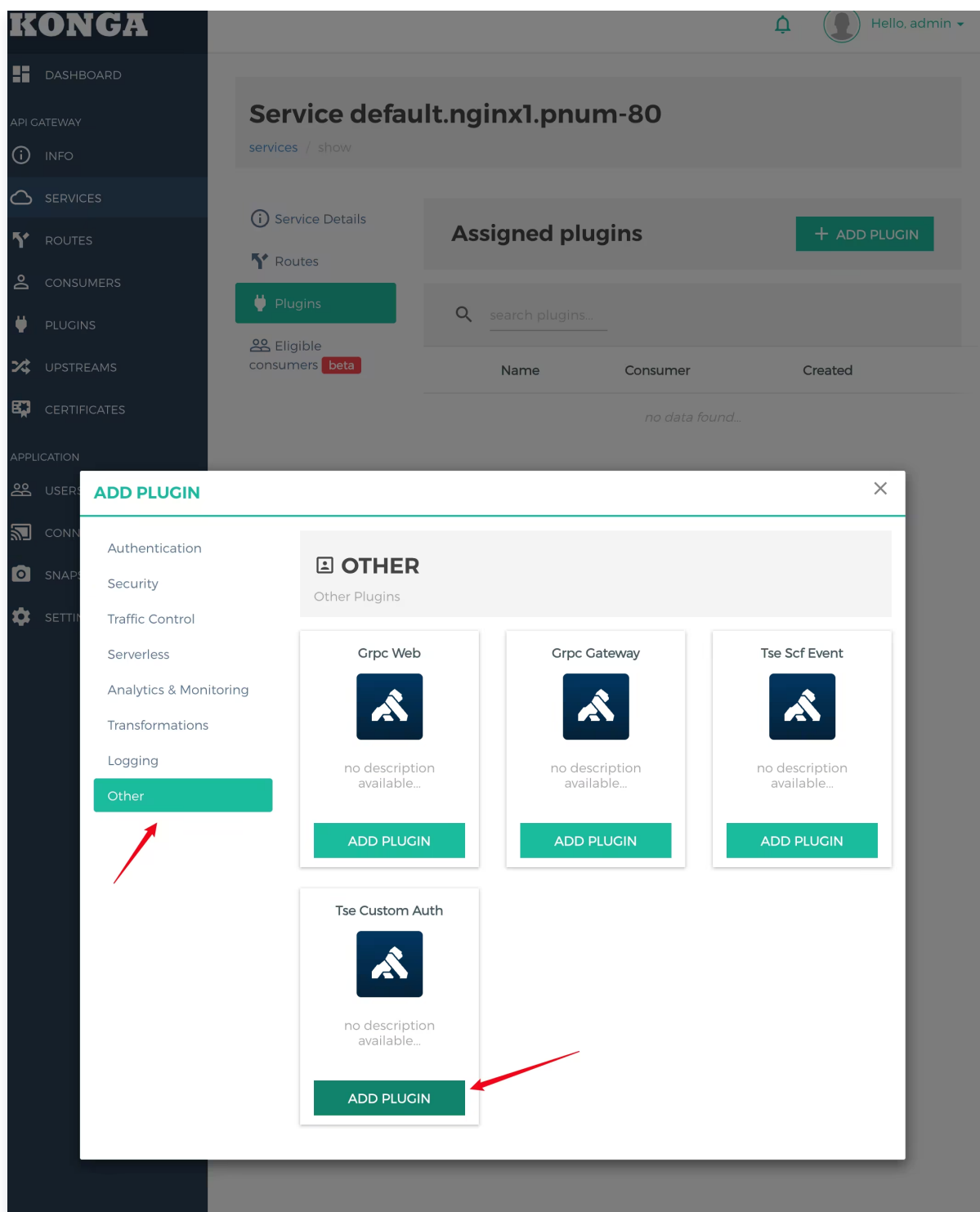
场景说明：当您的客户端始终接受 200 状态码，需要通过响应头来标识鉴权结果时，可配置 `auth result identifier` 为 `response header`，并定义一个存放鉴权结果的 Header 头。网关将尝试获取该响应头的值判定鉴权结果，当 Header 值为 `true` 时认为鉴权成功，网关将转发请求到后端服务；其余情况则认为鉴权失败，拒绝请求。

操作步骤：

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择 **云原生网关**，单击目标实例进入实例详情。
3. 在基础信息页面，单击 **Konga 控制台** 标签查看管理控制台登录方式。
4. 单击链接进入 Konga 控制台登录页面，输入用户名密码进入 Konga 控制台。



5. 单击 nginx 服务进入详情，单击 Plugins > +ADD Plugin 添加自定义鉴权插件。



6. 填写插件配置，单击 ADD PLUGIN，设置鉴权结果返回策略为响应头，并定义鉴权结果所在的响应头 Header 为 AuthRes。

ADD TSE CUSTOM AUTH

×

Configure the Plugin.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

description

auth service id

7b833ed6-108e-4361-8470-38a5da15dbe0

auth service path

/oauth/currentUserForTencent

token key name

Authorization

route id whitelist

Tip: Press **Enter** to accept a value.

timeout

9000

auth server unavailable policy

deny

redirect path

allow request body

NO

allow request headers

Tip: Press **Enter** to accept a value.

auth result identifier

response_header

auth result response header

AuthRes

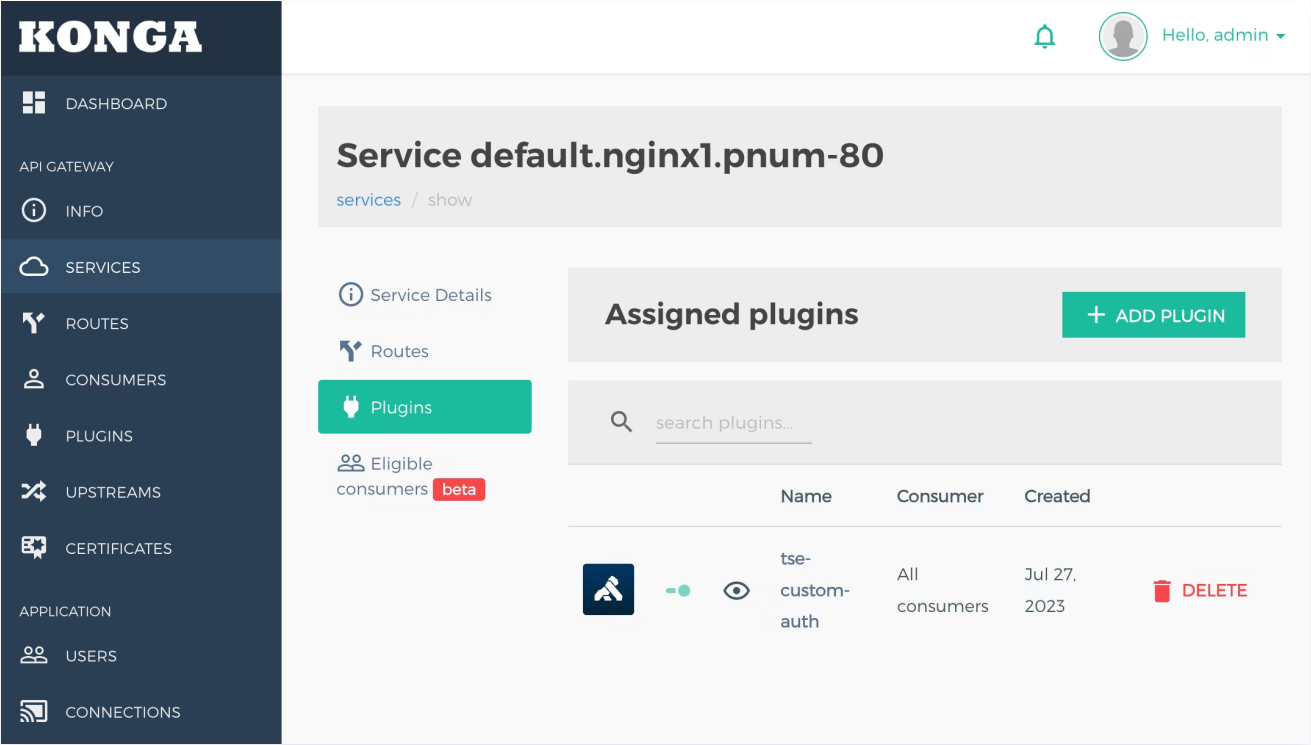
auth body header

allow response headers

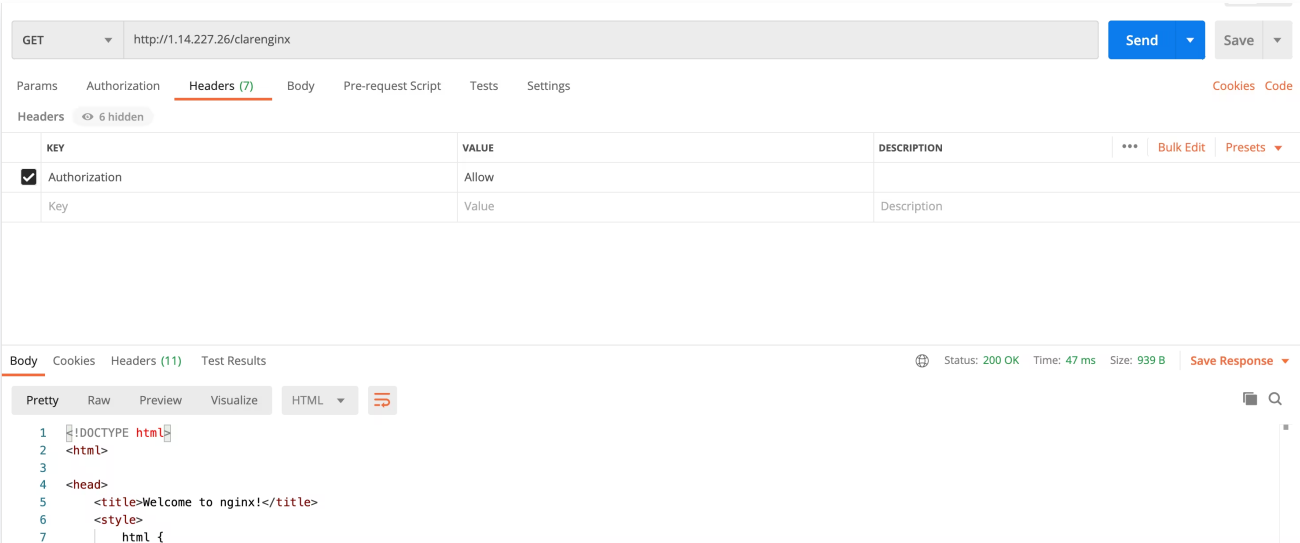
Tip: Press **Enter** to accept a value.

✓ ADD PLUGIN

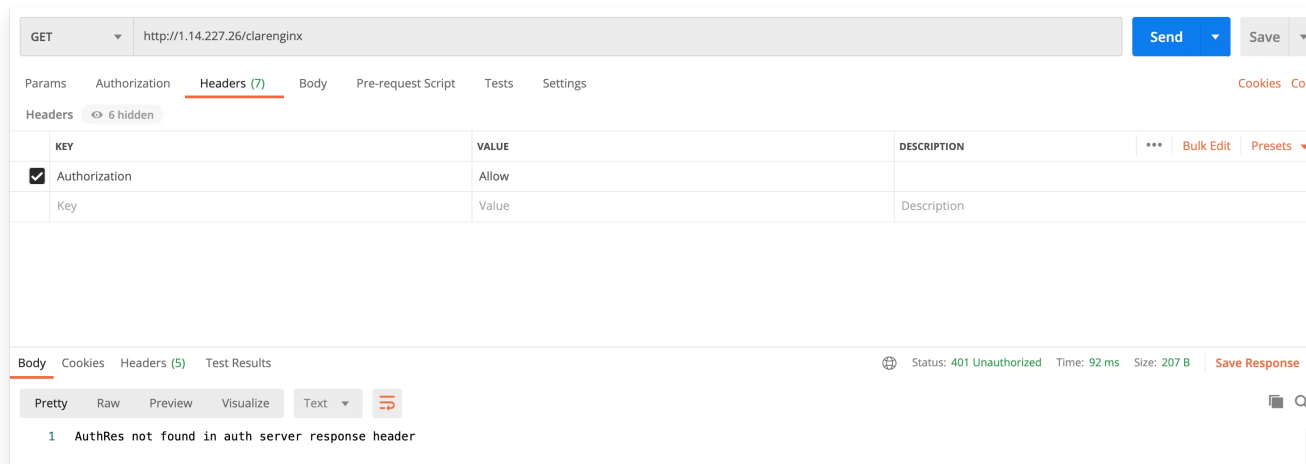
7. 确认插件已绑定后端服务。



8. 鉴权服务返回响应头 AuthRes=true 时，鉴权通过。



9. 鉴权服务未返回响应头 AuthRes 时，鉴权失败。

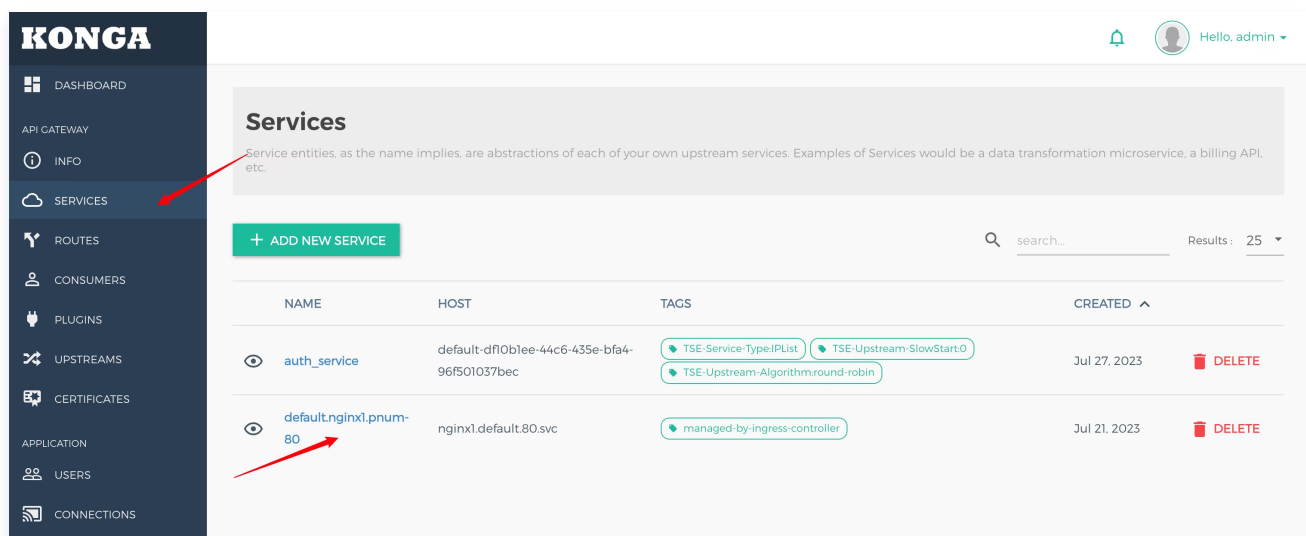


场景三：鉴权服务不可用场景

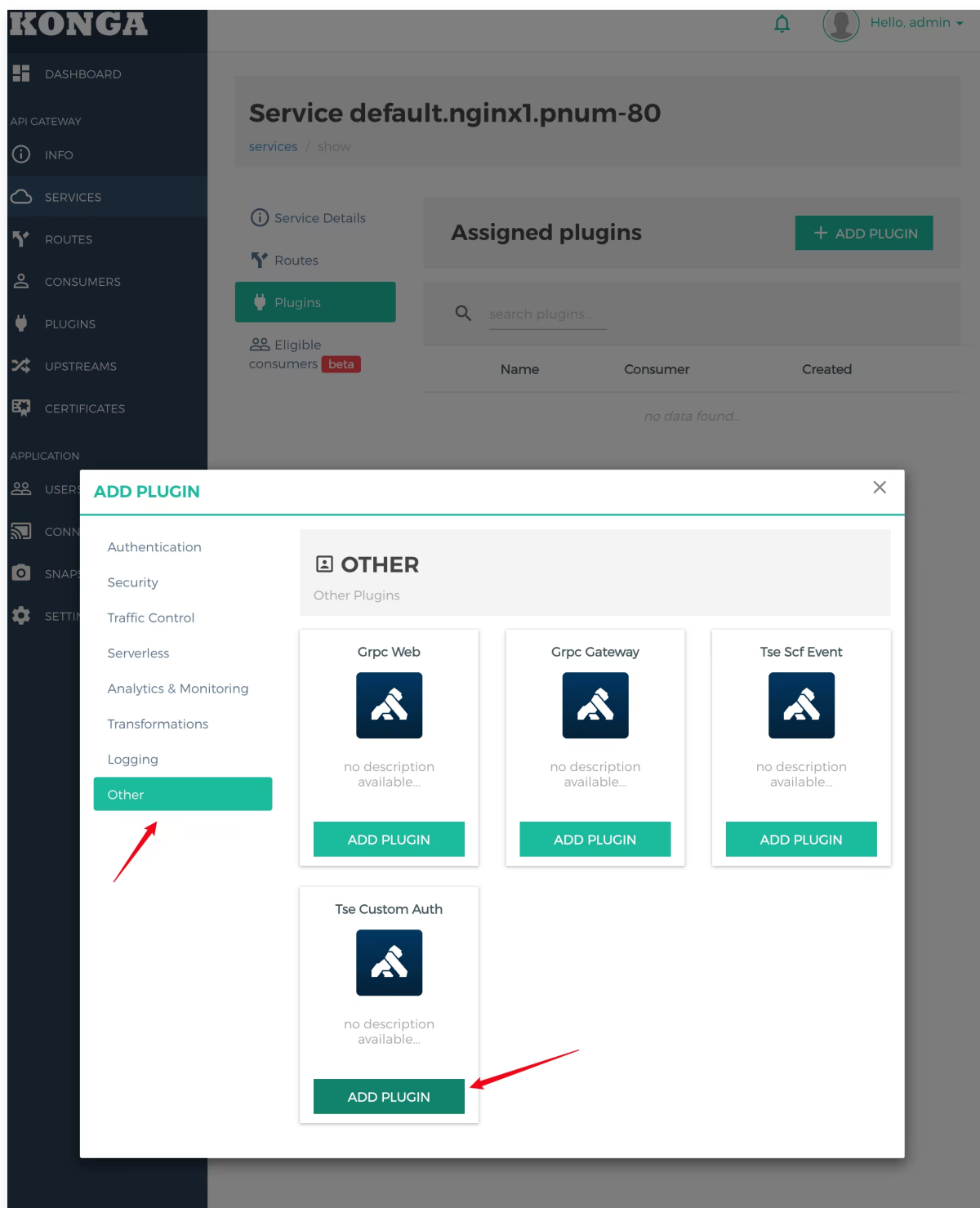
场景说明：当鉴权服务不可用（没有回包）时，网关会根据您的配置对请求进行全部放通/全部拒绝。

操作步骤

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。
4. 单击链接进入 Konga 控制台登录页面，输入用户名密码进入 Konga 控制台。



5. 单击 nginx 服务进入详情，单击 **Plugins > +ADD Plugin** 添加自定义鉴权插件。



6. 填写插件配置，单击 ADD PLUGIN，设置鉴权服务不可用策略为直接拒绝，配置超时时间为 5000（即 5 秒）。

EDIT TSE CUSTOM AUTH

Configure the Plugin.

ENABLED

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

description

auth service id

7b833ed6-108e-4361-8470-38a5da15dbe0

auth service path

/oauth/currentUserForTencent

token key name

Authorization

route id whitelist

Tip: Press **Enter** to accept a value.

timeout

5000

auth server unavailable policy

deny

redirect path

allow request body

NO

allow request headers

Tip: Press **Enter** to accept a value.

auth result identifier

response_header

auth result response header

AuthRes

auth body header

allow response headers

Tip: Press **Enter** to accept a value.

✓ SUBMIT CHANGES

7. 模拟鉴权服务故障，携带 Token 请求服务，请求在达到超时时间后拒绝，返回的 Header 中标注失败原因。

GET

http://1.14.227.26/clarenginx

Send

Save

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Code

Headers

6 hidden

KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
☒ Authorization	Allow				
Key	Value	Description			

Body

Cookies

Headers (6)

Test Results

Status: 403 Forbidden

Time: 5.07 s

Size: 201 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

Auth Service Fail

GET

http://1.14.227.26/clarenginx

Send

Save

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Code

Headers

6 hidden

KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
☒ Authorization	Allow				
Key	Value	Description			

Body

Cookies

Headers (6)

Test Results

Status: 403 Forbidden

Time: 5.07 s

Size: 201 B

Save Response

KEY	VALUE
Date ①	Thu, 27 Jul 2023 07:42:46 GMT
Content-Length ①	17
Connection ①	keep-alive
AuthServiceFail ①	timeout
X-Kong-Response-Latency ①	5002
Server ①	kong/2.5.1

8. 修改插件配置，设置鉴权服务不可用策略为直接放通。

EDIT TSE CUSTOM AUTH



Configure the Plugin.

ENABLED

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

description

auth service id

5de0695d-6bfl-4b8b-baef-210299c031b9

auth service path

/oauth/currentUserForTencent

token key name

Authorization

route id whitelist

Tip: Press **Enter** to accept a value.

timeout

5000

auth server unavailable policy

allow

redirect path

allow request body

NO

allow request headers

Tip: Press **Enter** to accept a value.

auth result identifier

response_header

auth result response header

AuthRes

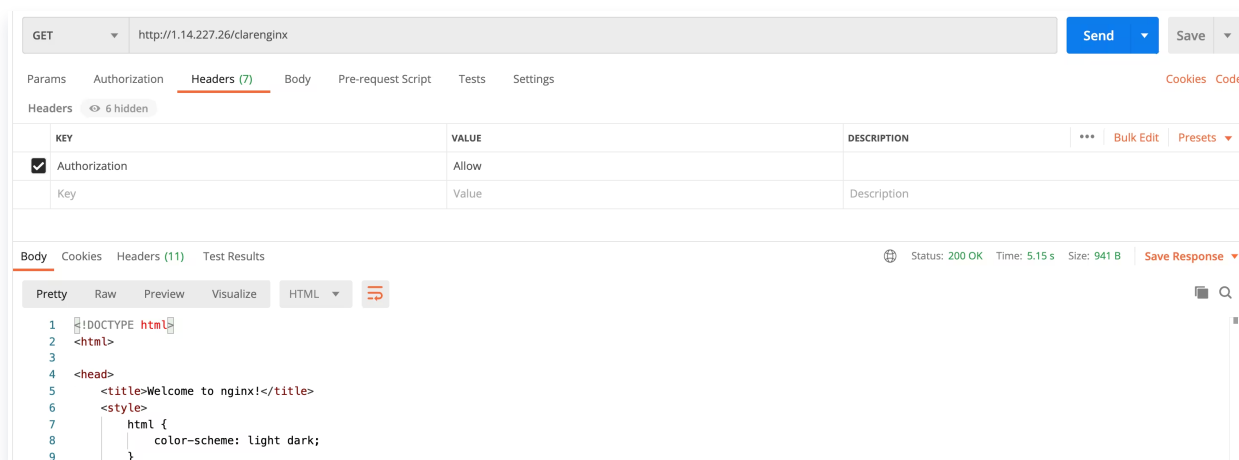
auth body header

allow response headers

Tip: Press **Enter** to accept a value.

✓ SUBMIT CHANGES

9. 模拟鉴权服务故障，携带 Token 请求服务，请求在达到超时时间后认为鉴权服务无响应，直接放通请求。



Ingress 配置方式

1. 定义 KongPlugin 资源，可参考如下示例：

```
apiVersion: configuration.konghq.com/v1
kind: KongPlugin
metadata:
  name: my-test-auth
config:
  timeout: 10000
  allow_request_body: true
  token_key_name: Authorization
  auth_result_identifider: response_header
  auth_result_response_header: AuthResponse
  auth_service_path: /
  auth_body_header: AuthBody
  redirect_path: null
  allow_response_headers:
    - myRespHeader1
    - myRespHeader2
    - myRespHeader3
  allow_request_headers:
    - myReqHeader
  auth_service_id: 73efd5bb-7468-43bc-9ca6-03f118169ca2
  route_id_whitelist: null
  description: null
  auth_server_unavailable_policy: allow
```

```
plugin: tse-custom-auth
```

2. 将插件绑定在 service 或 route 上。

```
kubectl apply -f tseauth.yaml  
kubectl annotate service nginx konghq.com/plugins=my-test-auth
```

流量控制

使用缓存加速访问

最近更新时间：2026-07-23 17:07:33

操作场景

本文介绍如何在云原生网关上通过缓存（Proxy Cache）插件实现下述常见缓存场景：

- 针对指定 API 开启缓存插件。
- 通过 Admin API 查看指定缓存。
- 通过 Admin API 删除指定缓存。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

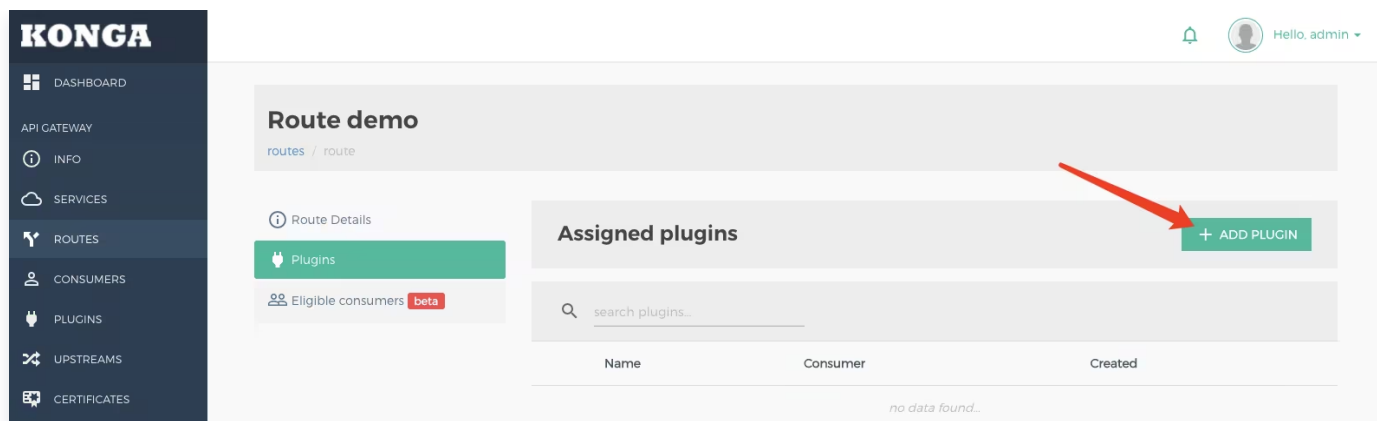
操作步骤

场景一：针对指定 API 开启缓存插件

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择 **云原生网关**，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 登录云原生网关管理控制台，进入需要缓存的 **Route** 详情页，单击 **Add Plugin** 按钮创建插件，在 **Traffic Control** 分组下选择 **Proxy Cache** 插件。



5. 在插件配置中填写以下配置，按 Enter 键，并保存。

- `strategy` : `memory` // 缓存存储策略（必填），目前仅支持 `memory` 模式。
- `response code` : `200` // 允许缓存的后端响应码列表。建议输入正常响应码（例如 200），避免异常响应内容被缓存。
- `request method` : `GET` // 允许缓存的请求方法。
- `content type` : `application/json; charset=utf-8` // 允许缓存的后端响应 Content-Type，注意该值必须和响应头完全匹配，例如 "application/json" 和 "application/json; charset=utf-8" 是不匹配的。
- `cache ttl` : `60` // 缓存时长，单位秒。

6. 如果需要根据参数维度缓存，则需要额外配置。

- `vary query params` : `name, department` // 用于缓存 key 的 query 参数名称列表，默认为空。
- `vary headers` : `x-cache-header` // 用于缓存 key 的 header 参数名称列表，默认为空。

vary query params

name X department X

Tip: Press **Enter** to accept a value.

vary headers

Tip: Press **Enter** to accept a value.

7. 发起 API 请求，验证缓存生效。

- 首次请求 API URL `GET /cache/it`，缓存不存在，会请求到后端，响应头 `X-Cache-Status` 为 `Miss`

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 335
X-Cache-Key: 6fb1b6a4440980c758b0eff31177a41d
X-Cache-Status: Miss
Date: Tue, 26 Apr 2022 16:04:19 GMT
X-Kong-Upstream-Latency: 948
X-Kong-Proxy-Latency: 321
Via: kong/2.4.1
```

- 缓存 TTL 之内再次请求 `GET /cache/it`，会命中缓存，响应头 `X-Cache-Status` 为 `Hit`

```
HTTP/1.1 200 OK
Server: openresty
Content-Type: application/json; charset=utf-8
X-Cache-Key: 6fb1b6a4440980c758b0eff31177a41d
Age: 2
X-Cache-Status: Hit
Date: Tue, 26 Apr 2022 16:09:23 GMT
Vary: Accept-Encoding
Content-Length: 335
X-Kong-Upstream-Latency: 0
X-Kong-Proxy-Latency: 0
Via: kong/2.4.1
```

- 当 `vary query params` 配置了 `name` 参数：

- 首次请求 `GET /cache/it?name=john`，缓存不存在，会请求到后端，响应头 `X-Cache-Status` 为 `Miss`。
- 缓存 TTL 之内再次请求 `GET /cache/it?name=john`，会命中缓存，响应头 `X-Cache-Status` 为 `Hit`。
- 缓存 TTL 之内携带不同 `name` 参数请求 `GET /cache/it?name=tom`，由于 `name` 参数不同，响应头 `X-Cache-Status` 为 `Miss`。

场景二：通过 Admin API 查看指定缓存

1. 通过 Proxy 代理方式开放 `admin api`（建议开启安全认证）。
2. 调用 Proxy Cache API `GET /proxy-cache/:cache_key` 查询特定 Cache Key 的缓存内容：
 - Cache Key 从 API 请求的响应头 `X-Cache-Key` 获取。
 - 当 Cache 不存在时，返回 `404`。
 - 当 Cache 存在时，返回 `200`，响应 Body 如下：

```
{
  "ttl": 60,
  "req_body": "",
  "headers": {
    "Date": "Wed, 27 Apr 2022 03:09:39 GMT",
    "X-Cache-Status": "Miss",
    "ETag": "W/\"14f-3sRRcm7jenKggj3qhgagNHIwiqw\"",
    "Vary": "Accept-Encoding",
    "connection": "keep-alive",
    "set-cookie": "sails.sid=s%3AT_YQH4rJgm2TYpJ5o6Ql1nH-vlA6pR19.JR3bgC6WfzW01OqDPwzaNn%2FXX6PcGdg2vTaT6d1BbfQ; Path=/; HttpOnly",
    "content-length": "335",
    "content-type": "application/json; charset=utf-8",
    "X-Cache-Key": "6fb1b6a4440980c758b0eff31177a41d"
  },
  "body": "<cached response body>",
  "status": 200,
  "timestamp": 1651028979,
  "version": 1,
  "body_len": 335
}
```

注意事项

- 缓存内容包含响应 Body 和响应 Headers（不缓存 [Hop-By-Hop headers](#)）。
- 如需在客户端中通过 `Cache-Control` 请求头对缓存行为进行动态控制，则需要在插件中启用 `cache-control` 配置项。
- `X-Cache-Status` 响应头说明：
 - `Miss`：请求匹配缓存策略，但未命中缓存。
 - `Hit`：请求命中缓存。
 - `Bypass`：请求和缓存插件配置规则不匹配。
 - `Refresh`：缓存未过期，但由于请求头中 `Cache-Control` 原因请求绕过了缓存。

参考

更多相关说明请参见 [Kong Proxy Cache 插件文档](#)。

使用熔断保护后端

最近更新时间：2026-07-23 17:07:33

操作场景

本文介绍如何在云原生网关上通过熔断插件实现下述常见熔断场景：

- 根据后端响应时间进行熔断。
- 根据错误码进行熔断。

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

插件说明

- 插件名称：TSE BREAKER
- 分类：Traffic Control
- 概述：实现对客户端的请求进行熔断，从而保护上游业务服务。
- 具体能力：
 - 支持响应时间模式和错误码模式。针对不同模式配置熔断开启条件，错误条件支持响应时间模式和错误码模式。
 - 响应时间模式：需配置后端服务响应时间阈值，超出响应时间阈值为一次错误，达到熔断开启条件触发熔断。
 - 错误码模式：需配置异常状态 HTTP 状态码，状态码使用英文逗号隔开，请求返回异常状态码为一次错误，到达熔断达到启条件触发熔断。
 - 支持熔断后半开探测：第一次触发熔断，熔断2s，超过时间后，重新转发后端，进行错误条件和健康条件判断。
 - 满足错误条件时，熔断4s，以此类推，直到达到预设的最大探测等待时间，停止指数递增。
 - 满足健康条件（即正常状态达到连续正常请求次数）时，结束熔断。
 - 支持自定义熔断返回：可自定义配置熔断状态 HTTP 状态码、熔断状态 HTTP 响应内容、熔断状态响应头。

插件配置

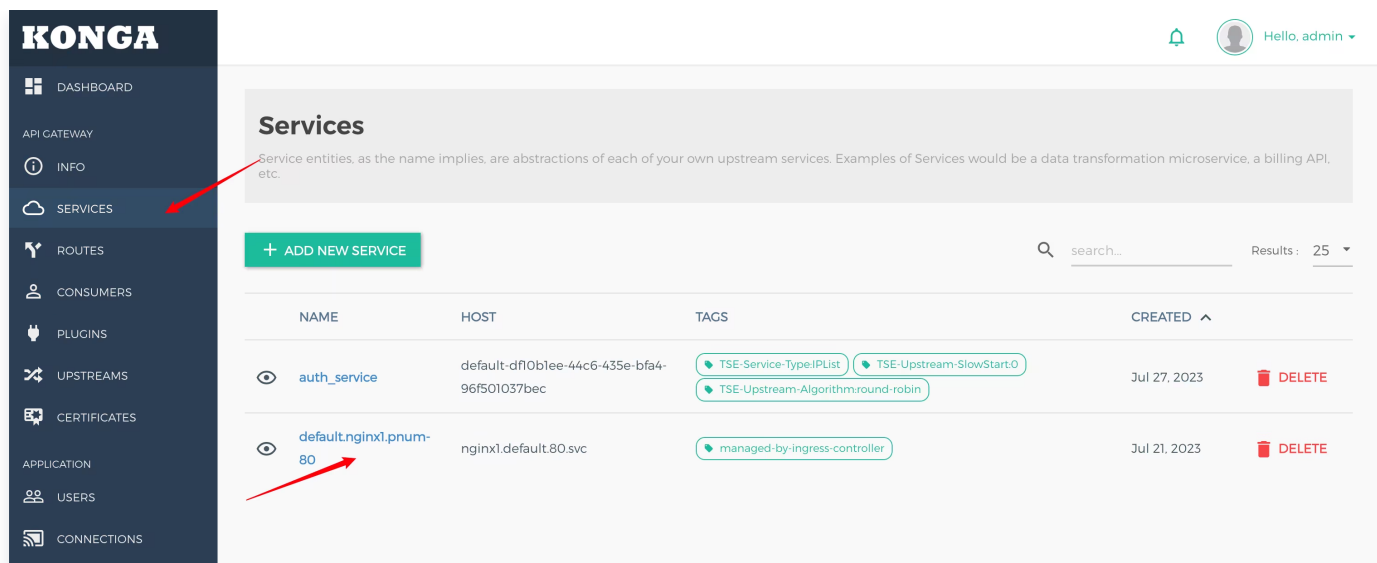
字段名称	是否必填	字段说明
响应时间阈值	否	>=1。上游服务后端最大响应时间，单位为秒。

异常状态 HTTP 状态码	否	默认503，支持[500,……599]。上游服务处于异常状态时返回的 HTTP 状态码范围。
错误率阈值	否	上游服务在一定时间内触发异常状态的请求次数比例阈值。
连续错误数阈值	否	>=1，上游服务连续触发异常状态的请求次数阈值。
最大探测等待时间	否	>=1，最大 300，单位秒，上游服务熔断的最大持续时间。
连续正常请求次数	是	>=1，上游服务处于健康状态时的连续正常请求次数。
熔断状态 HTTP 状态码	否	当上游服务处于不健康状态时返回的 HTTP 错误码。
熔断状态 HTTP 响应内容	否	当上游服务处于不健康状态时返回的 HTTP 响应体信息。
熔断状态响应头	否	当上游服务处于不健康状态时返回的 HTTP 响应头信息。该字段仅在配置了 break_response_body 属性时生效。

使用场景与操作步骤

场景一：根据错误码进行熔断

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。
4. 单击链接进入 Konga 控制台登录页面，输入用户名密码进入 Konga 控制台。



5. 进入需要配置熔断的 service 详情页。

6. 单击 **Add Plugin** 添加插件，在 Traffic Control 分组下选择 TSE BREAKER 插件。

7. 本场景模拟当后端服务返回500，501，502时认为后端异常，连续3次请求错误时触发熔断，熔断返回自定义内容。

在插件中填写以下配置：

- 错误判断条件：后端服务返回500,501,502。
- 熔断开启条件：连续错误数超过3次。
- 熔断响应配置（可选）。
 - 状态码：503
 - 响应头：example-resp-header:kong-cb
 - 响应内容：Service is broken

8. 发起 API 请求，查看3次返回500后熔断成功，返回自定义熔断响应。

```
HTTP/1.1 503 Service Unavailable //自定义响应状态码
```

```
Example-Resp-Header: kong-cb //自定义响应头
```

```
Service is broken //自定义响应内容
```

```
claire@CLAREZZHANG-MB0 ~ % curl -i http://[redacted]/errorcode
HTTP/1.1 500 Internal Server Error
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Date: Fri, 04 Nov 2022 02:43:09 GMT
Server: apigw/1.0.15
Via: kong/2.5.1
X-API-Id: api-lujd1sbo
X-API-Requestid: 504ab2e4e66acce41de [redacted]
X-Kong-Proxy-Latency: 108
X-Kong-Upstream-Latency: 21
Content-Length: 12
```

第一次请求, 返回500

```
bad request
claire@CLAREZZHANG-MB0 ~ %
claire@CLAREZZHANG-MB0 ~ %
claire@CLAREZZHANG-MB0 ~ % curl -i http://[redacted]/errorcode
HTTP/1.1 500 Internal Server Error
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Date: Fri, 04 Nov 2022 02:43:12 GMT
Server: apigw/1.0.15
Via: kong/2.5.1
X-API-Id: api-lujd1sbo
X-API-Requestid: 030da0bf8ed93c984e [redacted]
X-Kong-Proxy-Latency: 1
X-Kong-Upstream-Latency: 15
Content-Length: 12
```

第二次请求, 返回500

```
bad request
claire@CLAREZZHANG-MB0 ~ %
claire@CLAREZZHANG-MB0 ~ %
claire@CLAREZZHANG-MB0 ~ % curl -i http://[redacted]/errorcode
HTTP/1.1 500 Internal Server Error
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Date: Fri, 04 Nov 2022 02:43:14 GMT
Server: apigw/1.0.15
Via: kong/2.5.1
X-API-Id: api-lujd1sbo
X-API-Requestid: 35dfe437e243063d2f [redacted]
X-Kong-Proxy-Latency: 1
X-Kong-Upstream-Latency: 16
Content-Length: 12
```

第三次请求, 返回500

```
bad request
claire@CLAREZZHANG-MB0 ~ % curl -i http://[redacted]/errorcode
HTTP/1.1 503 Service Unavailable
Connection: keep-alive
```

```
Content-Length: 17
Date: Fri, 04 Nov 2022 02:43:14 GMT
Example-Resp-Header: kong-cb
Server: kong/2.5.1
X-Kong-Response-Latency: 1
Content-Type: text/plain; charset=utf-8

Service is broken%
```

第四次请求，触发熔断，返回自定义熔断响应

场景二：根据响应时间进行熔断

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。
4. 单击链接进入 Konga 控制台登录页面，输入用户名密码进入 Konga 控制台。
5. 进入需要配置熔断的 Service 详情页，单击 **Add Plugin** 添加插件，在 Traffic Control 分组下选择 TSE BREAKER 插件。
6. 本场景模拟当后端服务响应时间超过15秒时认为后端异常，连续4次请求错误时触发熔断，请求连续正确2次即关闭熔断。

在插件中填写以下配置：

- 错误判断条件：后端服务响应时间超过15秒
- 熔断开启条件：连续错误数超过4次
- 熔断响应配置（可选）
 - 状态码：503
 - 响应头：example-resp-header:kong-cb
 - 响应内容：Service is broken
- 熔断关闭条件：连续正常请求达到2次

7. 发起 API 请求，查看4次请求超时后熔断成功，2次请求正常后关闭熔断。

使用流量镜像插件

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍如何在云原生网关上通过流量镜像插件实现下述常见镜像场景：

- 将生产流量的一定比例镜像到测试后端。
- 将生产流量镜像到多个测试后端。

前置条件

- 已购买云原生网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#) 以及 [路由](#)。

插件说明

- 插件名称；TSE TRAFFIC MIRROR。
- 插件分类：Traffic Control。
- 概述：实现对客户端的请求进行流量镜像，在线流量将被镜像到测试后端，镜像后端返回的响应数据会被自动丢弃，保证镜像后端的测试业务不会影响到线上业务，广泛用于在测试环境中模拟在线流量对新功能进行测试的场景。
- 具体能力：
 - 支持镜像到云原生网关 Upstream 或镜像服务地址。
 - 支持根据采样比例镜像：支持配置转发给镜像后端的流量比例。
 - 支持根据请求参数进行镜像：通过将插件配置在 Route 上，匹配到请求参数的请求会进行镜像。
 - 支持镜像到多后端。

插件配置

配置项	是否必填	配置说明
镜像比例	是	转发到镜像服务的请求比例，支持1-100。
后端地址类型	是	镜像服务类型，支持填写 Host 或 Upstream。
后端地址-Host	是	镜像服务地址，需包含 scheme（http/https）和 Host 地址，例如 http://127.0.0.1。
后端地址-	是	镜像服务端口，支持1-65535。

Port		
后端地址- Upstream	是	镜像服务的云原生网关 Upstream，需包含 scheme（http/https）和 Upstream 名称，例如 <code>http://myupstream</code> 。
后端地址- Path	否	镜像服务路径，不指定则默认使用当前请求路径。

操作步骤

场景一：根据采样比例镜像

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息
部署架构
参数配置
弹性伸缩
Konga 控制台

公网访问地址
<https://gateway-1234567890-konga-gz.apigw.tencentcs.com:1337/>

关闭

公网访问策略
白名单: 0.0.0.0
[修改](#)

内网访问地址
<https://gateway-1234567890-konga-in.apigw.tencentcs.com:1337/>

关闭

管理员账号
admin

管理员密码

[显示](#)

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

4. 单击公网访问地址进入 Konga 控制台。
5. 进入需要配置流量镜像的 Service 详情页，输入用户名密码进入 Kong 控制台，单击 **Add Plugin** 添加插件，在 Traffic Control 分组下选择 TSE TRAFFIC MIRROR 插件。
6. 在本场景中，模拟将线上30%的流量镜像到 Host 类型的测试后端。
7. 在插件中填写以下配置，单击 **ADD PLUGIN**：
 - 镜像比例：30%
 - 后端地址类型：Host
 - 后端地址：

- Host: 填写测试服务访问地址, 例如 `http://example-test-service`。
- Port: 填写测试服务端口。
- Path: 选填, 不填使用真实请求路径。

ADD TSE TRAFFIC MIRROR

Configure the Plugin.

镜像比例

30

%

转发到镜像后端的请求比例, 支持1-100

后端地址类型

Host

后端地址

Host

http://

镜像服务地址, 需包含schema (http/https), 如 http://127.0.0.1

Port

80

镜像服务端口, 支持1-65535

Path

/mirror

(可选) 镜像请求的路径, 不指定则默认使用当前请求路径

+ 新增

ADD PLUGIN

8. 发起 API 请求, 查看镜像后端收到一定比例的镜像请求。

场景二：镜像到多个后端

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关, 单击目标实例进入实例详情。
3. 在基础信息页面, 单击 Konga 控制台标签查看管理控制台登录方式。
4. 单击公网访问地址进入 Konga 控制台。
5. 进入需要配置镜像的 Service 详情页, 单击 **Add Plugin** 添加插件, 在 Traffic Control 分组下选择 TSE TRAFFIC MIRROR 插件。
6. 本场景模拟将流量镜像到多个 Upstream。

在插件中填写以下配置：

- 镜像比例：100%。
- 后端地址类型：Upstream。
- 添加多个后端地址，每个后端地址填写：
 - Upstream：填写镜像服务所在的 Upstream。
 - Path：选填，填写镜像请求路径。

ADD TSE TRAFFIC MIRROR ✕

Configure the Plugin.

镜像比例 %
转发到镜像后端的请求比例，支持1-100

后端地址类型

后端地址

Upstream	http://mirror-upstream1	✕
Path	/service1 (可选) 镜像请求的路径，不指定则默认使用当前请求路径	

Upstream	http://mirror-upstream2	✕
Path	/service2 (可选) 镜像请求的路径，不指定则默认使用当前请求路径	

+ 新增

✓ ADD PLUGIN

7. 发起 API 请求，在两个 Upstream 查看发起的镜像请求。

请求路由 进行 URL 重写

最近更新时间：2026-07-23 17:07:33

操作场景

本文介绍如何将 API 发布到 云原生网关 上，并实现 URL 重写。

前置条件

- 1. 已购买网关实例，详情请参见 [新建网关](#)。
- 2. 登录 [TSF 控制台](#)。
- 3. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
- 4. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-...-konga.gz.apigw.tencentcs.com:1337/> 关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-...-3-konga-in.gz.apigw.tencentcs.com:1337/> 关闭

管理员账号

admin

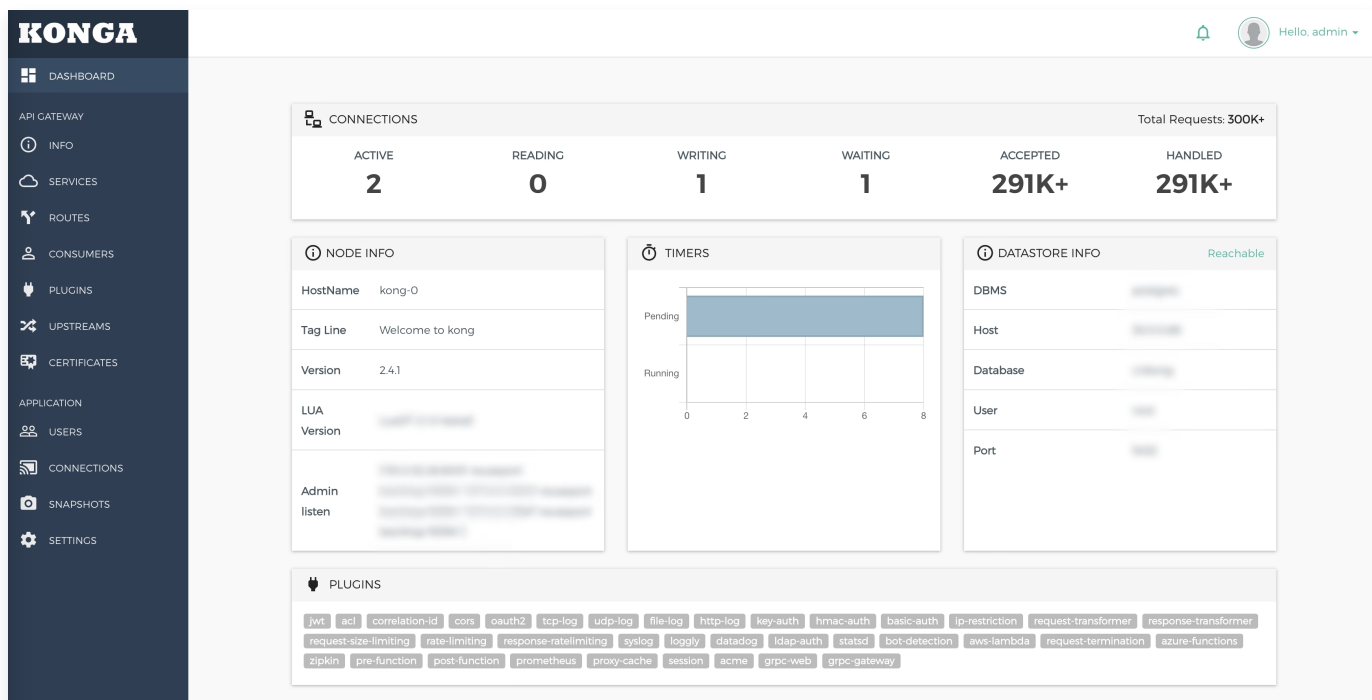
管理员密码

*****[显示](#)

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

✱ 加载中...

- 5. 访问 Konga 管理控制台。



操作步骤

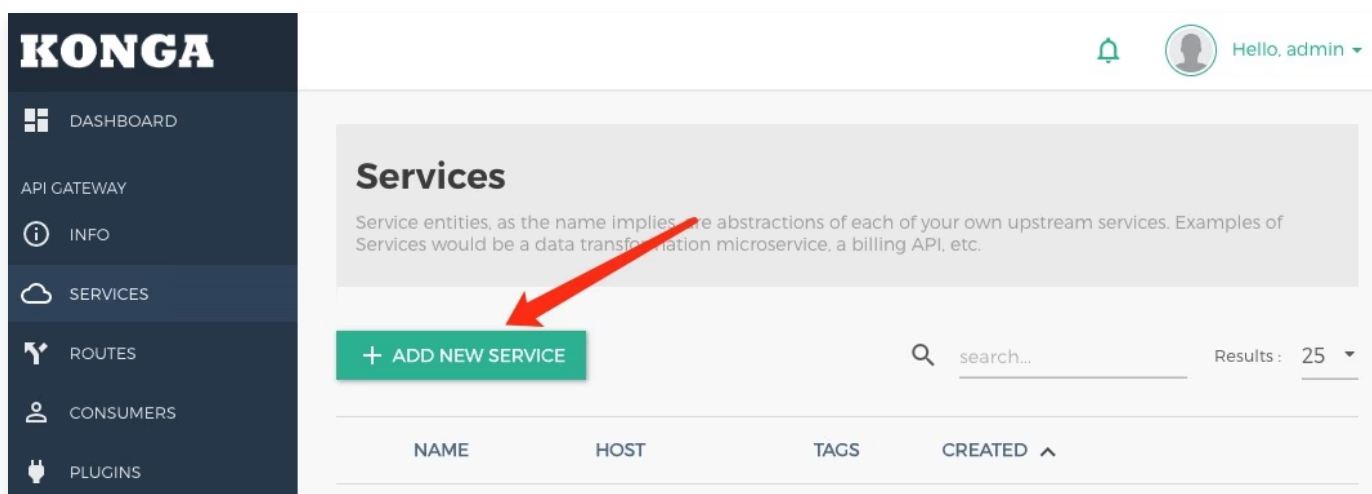
场景一：通过 Service 和 Route 配置实现 URL 前缀改写

❗ 示例

原始 API 调用 URL 为 `http://<backend>/anything/api/<sub_path>`

发布到 Kong 网关后，调用 URL 改写为 `http://<kong>/new/path/<sub_path>`

1. 单击 ADD NEW SERVICE 创建 Service。



填写相应信息并设置 Path 为 `/anything/api`。

CREATE SERVICE×

Name
(optional)

The service name.

Description
(optional)

An optional service description.

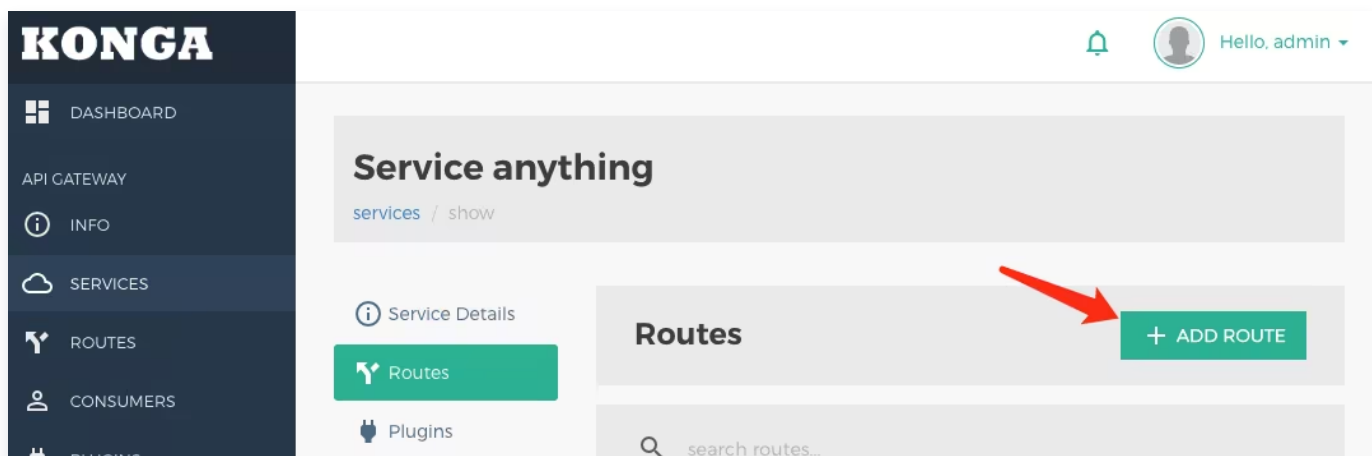
Tags
(optional)

Optionally add tags to the service

Url
(shorthand-attribute)

Shorthand attribute to set `protocol`, `host`, `port` and `path` at once.
This attribute is write-only (the Admin API never "returns" the url).

2. 单击 **ADD ROUTE** 在该 Service 上创建 Route，



设置 Path 为 `/new/path`，且开启 Strip Path 选项（默认开启）。

ADD ROUTE TO ANYTHING

* For hosts, paths, methods and protocols, snis, sources, headers and destinations press enter to apply every value you type

Name

(optional)

The name of the Route.

Tags

(optional)

Optionally add tags to the route

Hosts

(semi-optional)

A list of domain names that match this Route. For example: example.com. At least one of hosts, paths, or methods must be set.

Paths

(semi-optional)

/new/path

A list of paths that match this Route. For example: /my-path. At least one of `hosts`, `paths`, or `methods` must be set.

3. 请求 `http://<kong>/new/path/user` 时，后端接收到的请求路径为 `/anything/api/user`。

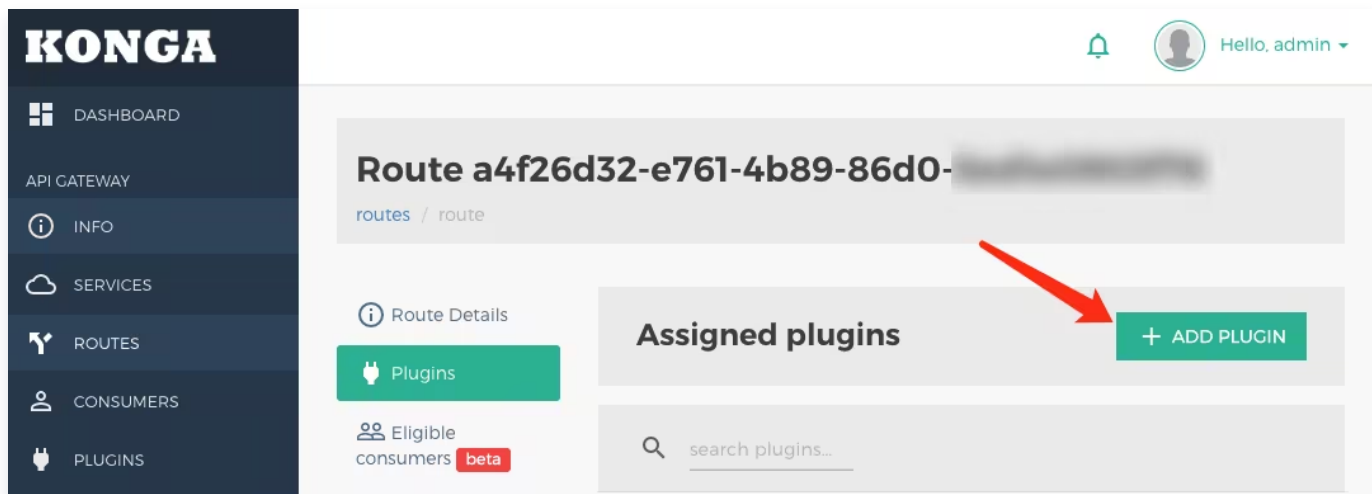
场景二：通过 Request Transformer 插件实现完整 URL 改写

❗ 示例

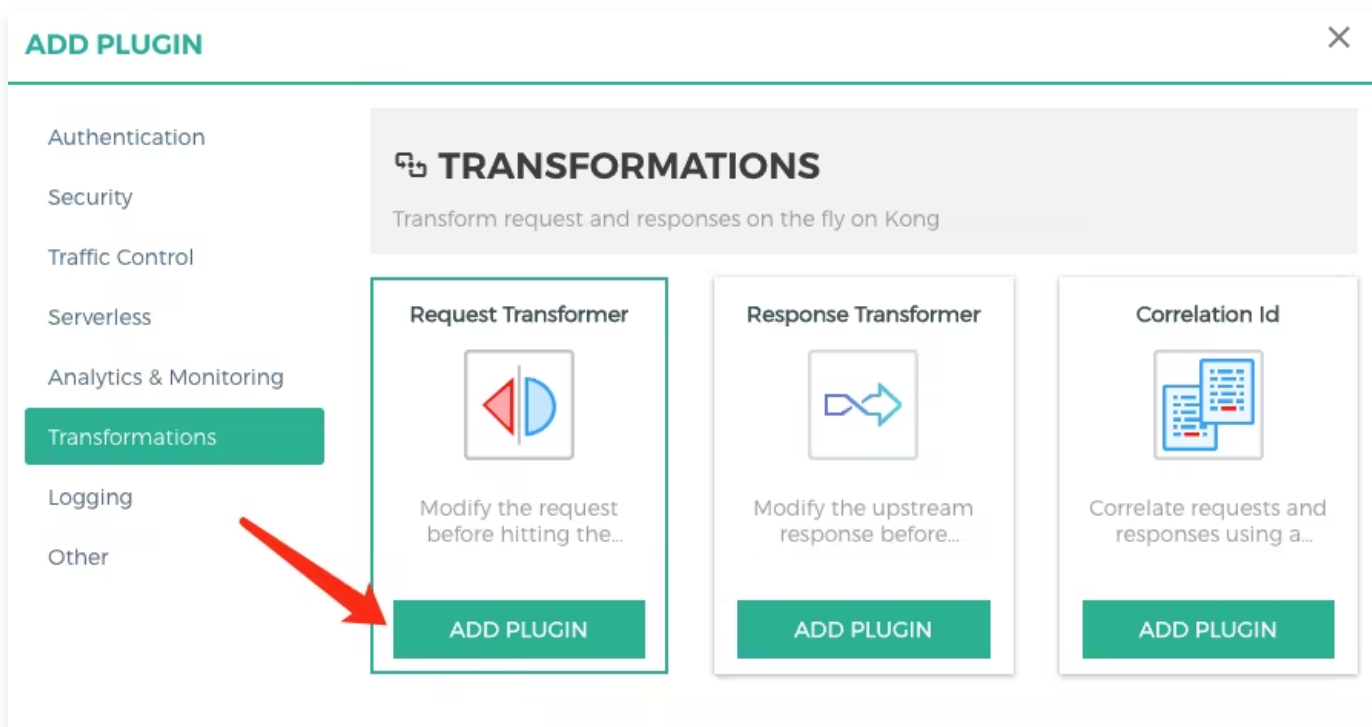
原始 API 调用 URL 为 `http://<backend>/anything/user_list`
发布到 Kong 网关后，调用 URL 改写为 `http://<kong>/users`

1. 创建 Service，Path 可以任意设置。
2. 在该 Service 上创建 Route，设置 Path 为 `/users`。

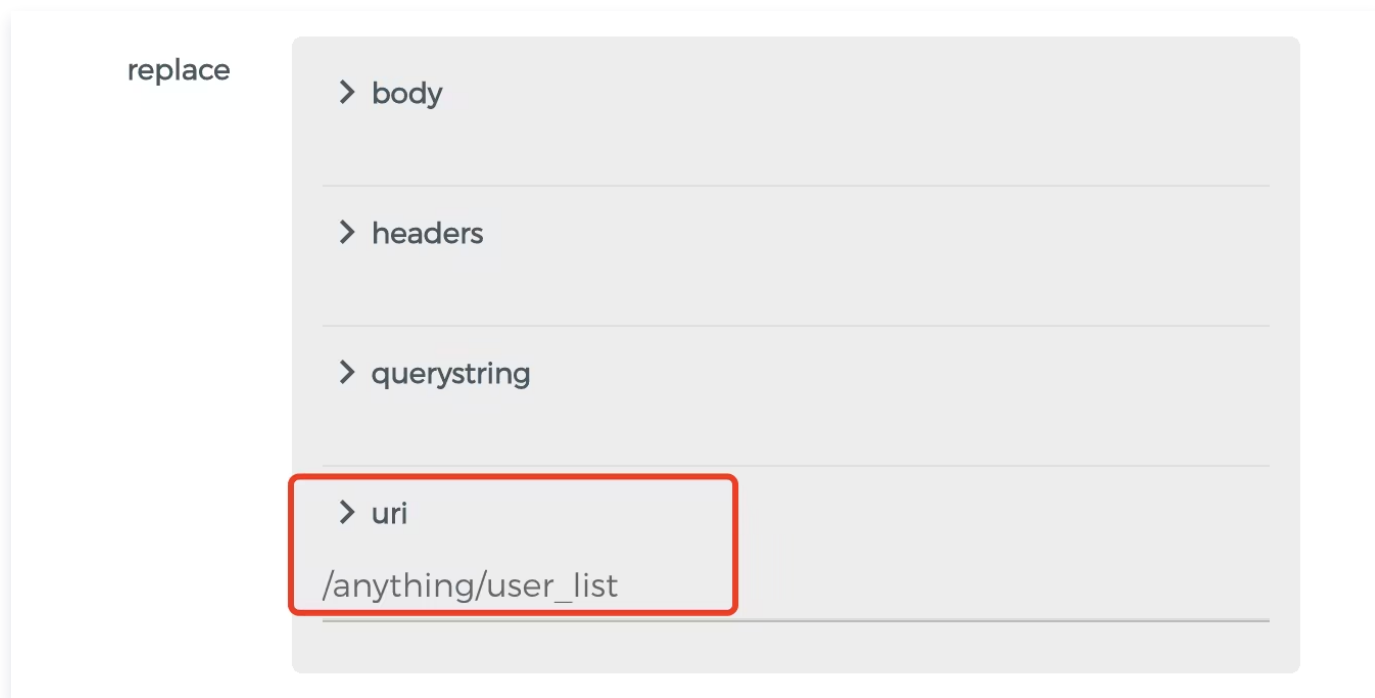
3. 单击 ADD PLUGIN 新建插件。



在该 Route 上创建 Request Transformer 插件。



并配置 `replace.uri` 为 `/anything/user_list`，该路径为后端接收到的实际请求路径。



The image shows a configuration interface for a 'replace' action. On the left, the word 'replace' is displayed. On the right, there is a list of expandable sections: 'body', 'headers', 'querystring', and 'uri'. The 'uri' section is currently expanded, showing the text '/anything/user_list'. This section and its content are enclosed in a red rectangular border.

4. 请求 Route 后，后端接收到的请求路径为 `/anything/path`（原始 Service Path 配置被覆盖）。

场景三：通过 Request Transformer 插件实现部分 URL 改写

❗ 示例

原始 API 调用 URL 为 `http://<backend>/anything/<user_id>/get`
发布到 Kong 网关后，调用 URL 改写为 `http://<kong>/user/<user_id>`

1. 创建 Service，Path 可以任意设置。

2. 在该 Service 上创建 Route，设置 Path 为 `/user/(?<user_id>\w+)`。

Route Details

Plugins

Eligible consumers beta

Route details

* For hosts, paths, methods and protocols, snis, sources, headers and destinations press enter to apply every value you type

Name
(optional)

The name of the Route.

Tags
(optional)

Optionally add tags to the route

Hosts
(semi-optional)

A list of domain names that match this Route. For example: example.com. At least one of hosts, paths, or methods must be set.

Paths
(semi-optional)

/user/(?<user_id>\w+)

A list of paths that match this Route. For example: /my-path. At least one of **hosts**, **paths**, or **methods** must be set.

3. 在该 Route 上创建 Request Transformer 插件。

ADD PLUGIN

Authentication

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

TRANSFORMATIONS

Transform request and responses on the fly on Kong

Request Transformer



Modify the request before hitting the...

ADD PLUGIN

Response Transformer



Modify the upstream response before...

ADD PLUGIN

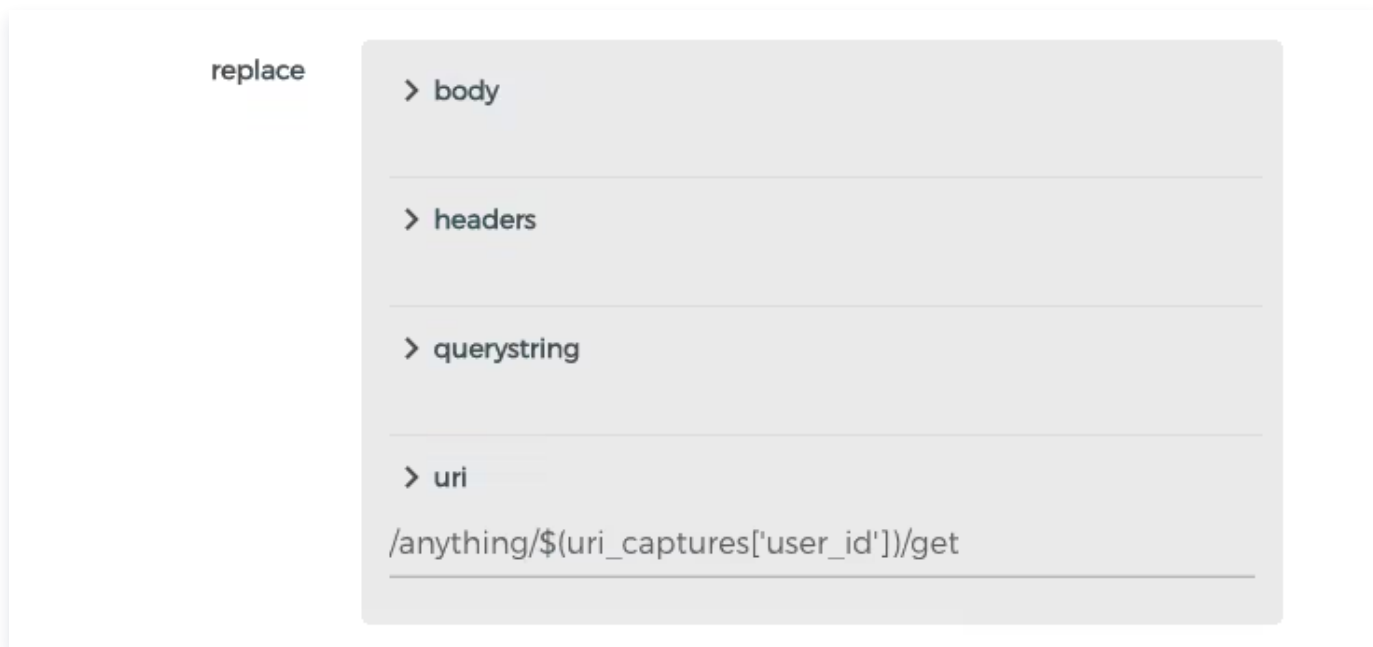
Correlation Id



Correlate requests and responses using a...

ADD PLUGIN

并配置 `replace.uri` 为 `/anything/${uri_captures['user_id']}/get`，该路径为后端接收到的实际请求路径。



4. 请求 `http://<kong>/user/user_a` 后，后端接收到的请求路径为 `/anything/user_a/get`。

相关参考

[Kong 插件文档](#)

使用重写插件改写请求或响应

最近更新时间：2026-07-23 17:07:35

操作场景

本文介绍如何在 云原生网关 上通过 Kong 插件实现下述常见请求/响应改写场景：

- 支持在请求阶段对 header/query/body 进行改写
- 支持在响应阶段对 header/body 进行改写

前置条件

- 已购买网关实例，详情请参见 [新建网关](#)。
- 配置了 [服务](#)（Service）以及 [路由](#)（Route）。

插件说明

- 插件名称：TSE PROXY REWRITE
- 插件分类：Transformations
- 概述：实现对客户端的请求和响应进行改写，支持使用 Nginx 内置变量。
- 具体能力：
 - 支持在请求或响应阶段对 query/header 参数进行新增、编辑、删除。
 - 当 content-type 为 application/json, multipart/form-data, application/x-www-form-urlencoded 时，支持在请求或响应阶段对 body 字段进行新增、编辑、删除。
 - 支持使用 NGINX 变量重写请求/响应。

插件配置

配置项	是否必填	配置说明
Header 参数	否	支持配置多条，每条配置项包括生效阶段、操作类型、键、值。 • 生效阶段：支持请求、响应。 • 操作类型：支持新增、删除、编辑。 • 键值：输入请求/响应的 Header 参数名和值。 • 支持以 \$var 的格式包含 NGINX 变量，例如 \$remote_addr \$balancer_ip。
Query 参数	否	支持配置多条，每条配置项包括生效阶段、操作类型、键、值。 • 生效阶段：支持请求。 • 操作类型：支持新增、删除、编辑。 • 键值：输入请求的 Query 参数名和值。

		支持以 \$var 的格式包含 NGINX 变量，例如 \$remote_addr \$balancer_ip。
Body	否	支持配置多条，每条配置项填写生效阶段、操作类型、键、值。 - 生效阶段：支持请求、响应。 - 操作类型：支持新增、删除、编辑。 - 键值：当 content-type 为 application/json, multipart/form-data, application/x-www-form-urlencoded 时，输入 body 字段的参数名和值。

⚠ 注意

当操作类型为编辑时，如果指定的参数不存在，则会按照新增操作新增键和值；如果指定的参数已存在，则会进行值替换。

操作步骤

场景一：实现响应阶段增加 Header 参数

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 单击公网访问地址进入 Konga 控制台
5. 进入需要配置流量镜像的 Service 详情页，单击 **Add Plugin** 添加插件，在 Transformations 分组下选择 TSE PROXY REWRITE 插件。

6. 在插件中填写以下配置，实现响应阶段新增一个 Header 参数。

The screenshot shows the 'ADD TSE PROXY REWRITE' configuration window. It has a title bar with a close button. Below the title bar is a subtitle 'Configure the Plugin.' The main content area is divided into three sections: 'Header参数', 'Query参数', and 'Body'. Each section has a '请求阶段' (Request Phase) and '响应阶段' (Response Phase) dropdown menu, followed by a '+ 添加配置' (Add Configuration) button. In the 'Header参数' section, the '响应阶段' dropdown is set to '新增' (New), and the configuration is 'addheader' with the value 'test'. At the bottom of the dialog is a green button with a checkmark and the text 'ADD PLUGIN'.

7. 发起 API 请求，查看后端服务的响应是否新增的 Header 参数。

```
HTTP/1.1 200 OK
Addheader: test
Connection: keep-alive
Content-Length: 20
Content-Type: application/octet-stream
Date: Thu, 22 Dec 2022 06:31:11 GMT
Server: nginx/1.20.1
Via: kong/2.5.1
X-Kong-Proxy-Latency: 30
X-Kong-Upstream-Latency: 66

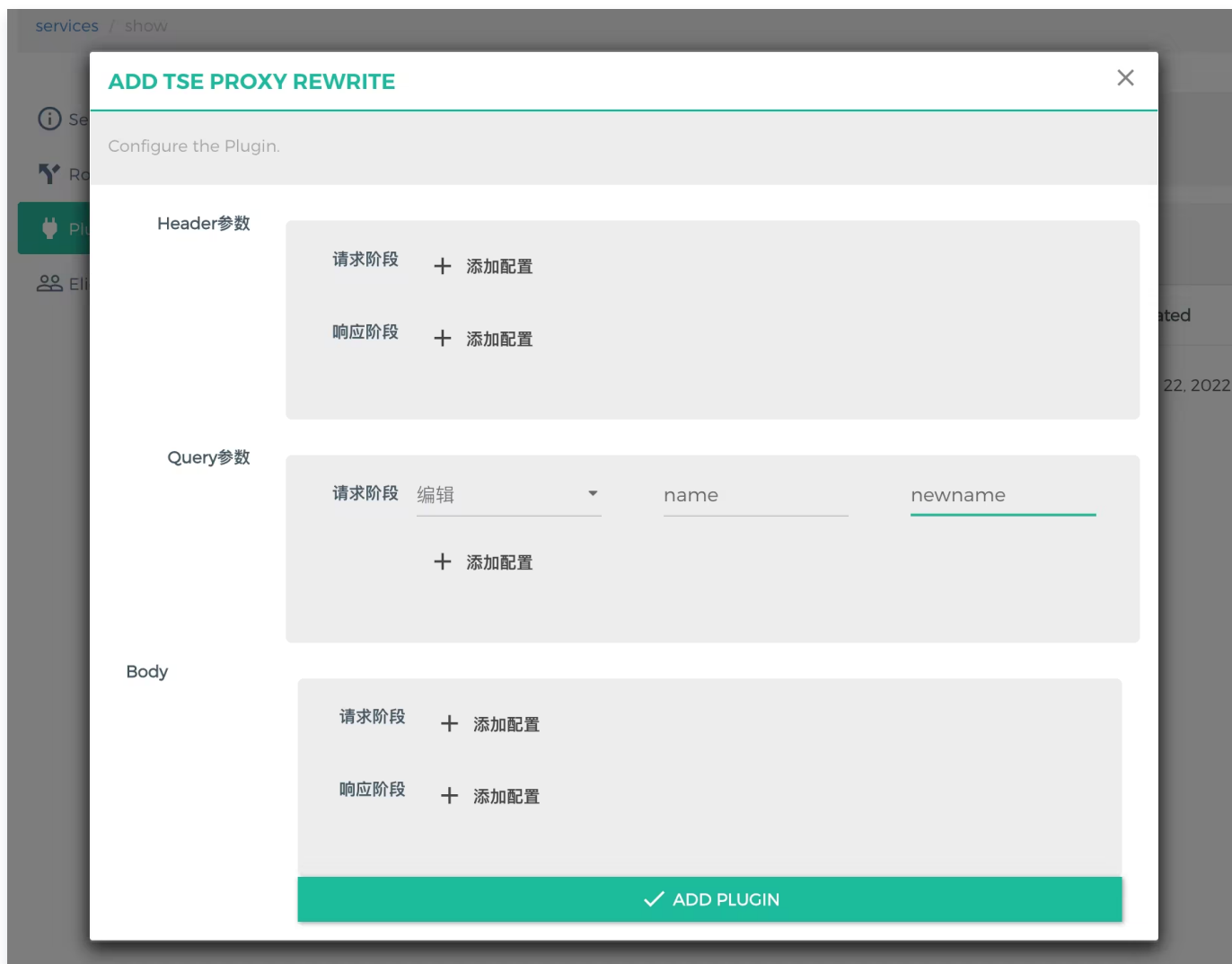
this is test service
```

场景二：实现请求阶段改写 Query 参数

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。



4. 单击公网访问地址进入 Konga 控制台
5. 进入需要配置流量镜像的 Service 详情页，单击 **Add Plugin** 添加插件，在 Transformations 分组下选择 TSE PROXY REWRITE 插件。
6. 在插件中填写以下配置，实现请求阶段改写 Query 参数。



7. 发起 API 请求，查看后端服务是否收到改写的 Query 参数。

场景三：实现响应阶段增加 Body

1. 登录 [TSF 控制台](#)。
2. 在左侧导航栏选择云原生网关，单击目标实例进入实例详情。
3. 在基础信息页面，单击 Konga 控制台标签查看管理控制台登录方式。

实例信息

部署架构

参数配置

弹性伸缩

Konga 控制台

公网访问地址

<https://gateway-1111111111-konga-gz.apigw.tencentcs.com:1337/>  关闭

公网访问策略

白名单: 0.0.0.0 [修改](#)

内网访问地址

<https://gateway-1111111111-konga-in.apigw.tencentcs.com:1337/>  关闭

管理员账号

admin 

管理员密码

***** [显示](#) 

为了您的业务安全，请尽快登录Konga控制台修改初始密码，此处仅展示初始密码。

4. 单击公网访问地址进入 Konga 控制台
5. 进入需要配置流量镜像的 Service 详情页，单击 **Add Plugin** 添加插件，在 Traffic Control 分组下选择 TSE PROXY REWRITE 插件。

6. 在插件中填写以下配置，实现响应阶段增加 Body 参数。

ADD TSE PROXY REWRITE ×

Configure the Plugin.

Header参数

请求阶段 + 添加配置

响应阶段 + 添加配置

Query参数

请求阶段 + 添加配置

Body

请求阶段 + 添加配置

响应阶段 新增 ▼ addbody return

+ 添加配置

✓ ADD PLUGIN

⚠ 注意

对 body 参数的操作，需要请求传入 Content-Type 的 Header。仅当 Content-Type 为 application/json, multipart/form-data, application/x-www-form-urlencoded 时，才会进行 body 字段的改写。

7. 发起 API 请求，查看后端服务的响应是否收到新增的 Body 参数。

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Date: Thu, 22 Dec 2022 06:53:43 GMT
Server: nginx/1.20.1
Via: kong/2.5.1
```

```
X-Kong-Proxy-Latency: 22
X-Kong-Upstream-Latency: 63
Content-Length: 54

{"myreturn":"this is test service","addbody":"return"}
```