云原生API网关

目录

产品介绍

计费说明

快速入门

用户指南

最佳实践

产品介绍

产品简介

概述

云原生API网关分为API网关和AI网关两个产品。API网关是一款面向现代应用架构设计的高性能网关解决方案，集成流量治理、安全防护、服务集成与可观测能力于一体。通过动态路由、插件热插拔、配置热更新等关键特性，实现对海量API请求的高效处理与灵活控制，全面适配微服务、Kubernetes、AI推理等多样化场景，助力企业构建稳定安全、可演进的API基础设施，全面提升系统的敏捷性与运维效率。AI网关是一款面向AI场景的高信念网关解决方案，统一代理大模型API、MCP Server和Agent API，可以作为AI应用与大模型服务、工具和其他Agent之间的桥梁，助力企业提升AI服务的集成效率和治理能力。

API网关

API网关在使用上，分为HTTP API、REST API、WebSocket API。

HTTP API

以路由为核心，适用于接口规范不统一的场景。推荐用于流量转发、Kubernetes Ingress、微服务间通信等应用。若您关注请求路由控制和流量管理，建议选择此类型。

REST API

遵循统一的OpenAPI 规范，适用于前后端协作、系统集成以及API的精细化管理。适合需要对外提供标准接口文档、SDK 等协作支持的场景。若您希望构建标准化、可治理的API接口体系，建议选择此类型。

WebSocket API

基于WebSocket协议，支持双向实时通信，适用于AI、IoT、即时通讯等对实时性要求高的场景。相较于传统HTTP接口，具备长连接和低延迟优势，已内置相关默认配置。若您的应用需实现高效的实时数据交互，建议选择此类型。

AI网关

Model API

专为高效实现大模型代理而设计，提供大模型代理、鉴权、可观测、策略与插件等能力。若您需要安全、高效的调用大模型服务，建议选择此类型。

MCP 服务

MCP（Model Context Protocol）是一种开源协议，旨在实现大语言模型与外部数据源和工具的集成，用来在大模型和数据源之间建立安全双向的连接，AI网关支持直接代理MCP 服务。若您需要对MCP服务的访问协议和入口实施统一管控，建议选择此类型。

产品介绍

Agent API

专为解决AI应用的协同和管理问题，可以作为AI应用的统一代理，提供丰富的流量治理策略和安全能力。若您需要统一纳管AI应用的访问入口，建议选择此类型。

产品特性

API网关

API网关同时具备传统的流量网关和业务网关功能，提供全局性和独立业务域级别的流量管理策略，支持Nacos、K8s和函数计算等多种服务发现方式，支持TLS加密通信和多种身份认证方式，构筑安全的流量入口。

功能集	功能	功能详情
实例管理	实例管理	支持网关实例的订购、查看、退订、版本升级、升降配操作
	接入点管理	支持关联ELB，实现网关公网访问以及入站负载均衡
	策略管理	支持实例级绑定IP黑白名单、自定义响应等策略
	可观测	支持实例级监控分析
API管理	API生命周期管理	支持API在线设计、调试、发布、下线等生命周期管理
	多版本管理	支持多版本共存
	快照管理	支持发布历史查看、在线版本切换等功能
	策略管理	支持API绑定限流、跨域、重写、认证、熔断等多种策略
	可观测	支持API级监控分析
路由管理	匹配管理	支持请求路径、请求头、请求方法、请求参数、Cookie等匹配条件
	策略管理	支持路由绑定限流、跨域、重写、认证、熔断等多种策略
	可观测	支持路由级监控分析
服务来源与服务管理	容器服务	支持从CCE发现并导入服务
	注册配置中心	支持从Nacos和Eureka发现并导入服务
	函数计算	支持从函数计算发现并导入服务
	固定地址/域名	指定IP地址或域名作为上游服务
域名管理	域名管理	支持域名管理，支持HTTP、HTTPS协议
	证书管理	支持证书管理

插件市场	插件管理	支持插件安装、卸载
	插件配置管理	支持插件配置启用、停用、绑定、解绑
	插件分类	丰富的插件分类，传输协议、安全防护、认证鉴权、流量管控等多种插件扩展支持
消费者管理	消费者管理	支持消费者新增、编辑、删除、启用、停用等
	消费者授权	支持消费者授权、解除授权，支持实例级、API级、路由级多维授权范围

AI网关

AI网关是一款面向AI场景的高信念网关解决方案，统一代理大模型API、MCP Server和Agent API，可以作为AI应用与大模型服务、工具和其他Agent之间的桥梁，助力企业提升AI服务的集成效率和治理能力。

功能集	功能	功能详情
实例管理	实例管理	支持网关实例的订购、查看、退订、版本升级、升降配等操作
	接入点管理	支持关联ELB，实现网关公网访问以及入站负载均衡
	可观测	支持实例级监控分析
	消费者认证	支持实例级的消费者认证
Model API管理	管理Model API	支持创建、编辑、调试和删除Model API
	策略与插件	支持绑定限流、跨域、熔断、黑白名单等多种策略
	消费者认证	支持为Model API开启消费者认证鉴权
	可观测	支持查看Model API的QPS、请求成功率和平均延迟等指标
MCP 管理	MCP服务	支持创建、编辑、发布、删除MCP服务
	策略与插件	支持绑定限流、跨域、熔断、黑白名单等多种策略
	消费者认证	支持为MCP服务开启消费者认证鉴权
	可观测	支持查看MCP服务的QPS、请求成功率和平均延迟等指标
Agent API管理	管理Agent API	支持创建、编辑、调试和删除Agent API
	策略与插件	支持绑定限流、跨域、熔断、黑白名单等多种策略
	消费者认证	支持为Agent API开启消费者认证鉴权

	可观测	支持查看Agent API的QPS、请求成功率和平均延迟等指标
服务管理	LLM 服务	支持从大模型服务商导入服务
	Agent服务	支持从Agent服务提供商导入服务
	DNS 域名	指定域名作为上游服务
	固定地址	指定IP地址作为上游服务
	容器服务	支持从CCE发现并导入服务
	注册配置中心	支持从Nacos发现并导入服务
	函数计算	支持从函数计算发现并导入服务
域名管理	域名管理	支持域名管理，支持HTTP、HTTPS协议
	证书管理	支持证书管理
插件市场	插件管理	支持插件安装、卸载
	插件配置管理	支持插件配置启用、停用、绑定、解绑
	插件分类	丰富的插件分类，传输协议、安全防护、认证鉴权、流量管控等多种插件扩展支持
消费者管理	消费者管理	支持消费者新增、编辑、删除、启用、停用等
	消费者授权	支持消费者授权、解除授权，支持实例级、API级、路由级多维授权范围

产品优势

高性能

云原生架构，Nginx + LuaJIT内核，低内存占用，事件驱动架构可高效支持 高吞吐、低延迟业务场景。

高可用

集群 & 多可用区部署，内置实时健康检查、流量管控、熔断、自动故障转移等能力，上游服务异常时，可自动切换流量或降级处理，确保业务连续性。

强扩展性

丰富的插件列表，支持热插拔，无需重启网关即可动态调整 API 处理能力。

易用性

开箱即用，提供可视化配置界面，简化API运营，降低使用门槛。

生态集成

深度融合天翼云微服务、可观测产品，构建一体化云原生API生态，助力企业快速构建数字化能力。

AI 代理

内置AI网关能力，提供 安全、高效、可扩展的AI访问和管理方案，加速企业AI业务落地。

应用场景

融合流量网关、微服务网关、API网关，提升效率、降低资源开销

传统网关架构中，流量网关（如Nginx）、微服务网关（如Spring Cloud Gateway）与API网关各司其职，承担不同职责：

- 流量网关：提供全局性、与业务无关的能力，如HTTPS卸载、Web防火墙、全局流量监控等。
- 微服务网关：紧耦合业务系统，支持注册中心集成，负责服务治理、认证鉴权等业务域策略。
- API网关：聚焦API的全生命周期管理，支持API设计、开发、测试、发布及接口级策略配置。

随着云原生架构的发展，云原生 API 网关将三者能力融合，形成统一网关层，具备以下优势和应用场景：

- API 全生命周期管理：覆盖从设计、开发到发布、下线的完整流程，简化API管理。
- 灵活流量调度：同时处理南北向（用户与服务）与东西向（服务间）流量，实现高效路由。
- 多层安全防护：支持HTTPS、IP黑白名单、身份认证等机制，强化服务安全。
- 服务治理能力：支持服务发现、健康检查与负载均衡。
- 精细化策略管理：支持网关全局、API与接口级别的限流、配额、访问控制等策略。
- 资源高效利用：统一网关架构减少系统冗余，降低维护成本，提升性能表现。

REST API生命周期管理

在微服务架构中，API 是系统集成与对外服务的关键接口。云原生 API 网关通过提供标准 REST API，支持从 API 设计到发布再到下线的全流程管理，使开发、测试、运维等各角色都能以统一方式高效管理接口。

WebSocket流量代理

WebSocket 协议支持客户端与服务器之间的持续、双向通信，具备连接持久性强、延迟低的优势。在K8s集群外部访问WebSocket服务时，云原生API网关作为入口，负责接收外部请求，进行认证鉴权、安全防护、流量管控等一系列自定义扩展处理，并根据预定义的路由规则将流量转发至对应的后端服务。

提供丰富的认证鉴权能力，降低安全接入成本

在云原生架构中，API网关作为统一入口，需要对所有外部请求进行身份认证和权限校验。云原生API网关支持WT、HMAC、API Key、Basic等多种认证方式，并可对接OAuth2等外部认证服务，适配不同接入方的安全需求。

AI 代理

随着人工智能技术的快速发展，越来越多的企业开始将AI应用于各个业务场景，以提升效率、优化决策并创造新的价值。然而，企业在应用AI的过程中也面临着诸多挑战，例如模型管理复杂、数据安全风险等。AI网关是企业应用AI的桥梁，它充当着连接企业应用与AI模型、算法和数据的中间层，为企业提供安全、高效、可扩展的AI能力接入和管理服务。

产品介绍

名词解析

域名

一串用点分隔的字符，作为互联网中某个实体的名称。在云原生API网关中可以根据请求中的域名信息进行路由匹配，域名还可以关联证书，实现TLS加密访问。

服务来源

网关转发请求到的服务的来源，当前支持天翼云容器引擎CCE、天翼云注册配置中心（Nacos引擎、Eureka引擎）、函数计算等服务来源。

服务

网关路由转发到后端的服务，可以是部署在K8s、注册到Nacos、注册到Eureka的服务，也可以是固定地址服务等。

路由规则

一个路由规则可以基于匹配路径、header等配置参数，将请求转发到指定的后端服务；同时可以配置跨域、限流等策略。

认证鉴权

网关需要对请求的身份进行校验，当前云原生API网关支持Key-Auth、Basic-Auth、JWT、HMAC、OIDC等方式实现对请求的身份认证和鉴权。

产品规格

云原生API网关规格

规格名称	主机类型：X86架构-通用型
基础版	#
标准版-1	#
标准版-2	#
标准版-3	#
企业版-1	#
企业版-2	#
企业版-3	#
铂金版-1	#
铂金版-2	#
铂金版-3	#
铂金版-4	#

X86架构通用型性能数据

规格名称	QPS（30% CPU，应答包1K，HTTP请求）	最大客户端连接数	最大HTTPS每秒新建连接数
基础版	2500	28000	1000
标准版-1	5000	56000	2000
标准版-2	10000	112000	4000
标准版-3	20000	224000	8000
企业版-1	40000	448000	16000
企业版-2	80000	896000	32000
企业版-3	120000	1320000	48000
铂金版-1	160000	1760000	64000
铂金版-2	320000	3520000	128000
铂金版-3	480000	5280000	192000
铂金版-4	600000	6600000	240000

AI网关规格

规格名称	主机架构
	X86架构
	通用型
标准版-1	#
标准版-2	#
标准版-3	#
企业版-1	#
企业版-2	#
企业版-3	#
铂金版-1	#
铂金版-2	#
铂金版-3	#
铂金版-4	#

X86架构通用型性能数据

规格名称	QPS（30% CPU，应答包1K，HTTP请求）	最大客户端连接数	最大HTTPS每秒新建连接数
标准版-1	1600	21000	700
标准版-2	3200	42000	1400

规格名称	QPS（30% CPU，应答包1K，HTTP请求）	最大客户端连接数	最大HTTPS每秒新建连接数
标准版-3	6400	84000	2800
企业版-1	12800	168000	5600
企业版-2	25600	336000	11200
企业版-3	38400	504000	16800
铂金版-1	51200	672000	22400
铂金版-2	102400	1344000	44800
铂金版-3	153600	2016000	67200
铂金版-4	192000	2520000	84000

引擎版本记录

云原生API网关引擎版本说明

版本号	发布时间	描述	升级网关对业务的影响
1.10.0	2025年11月29日	• 支持基本路由管理功能。 • 支持API全生命周期管理。 • 支持消费者管理功能。 • 支持Nacos和Kubernetes服务发现。 • 支持插件扩展。 • 支持日志、指标与链路追踪。	无影响

计费说明

计费项

云原生API网关的计费项为网关实例费用，目前云原生API网关产品提供两种网关类型：云原生API网关和AI网关。

计费方式

提供按需付费和包年包月两种付费模式。

系列1：云原生API网关

按需付费

按需付费是后付费模式，您只需按实际情况进行使用，然后按照使用账单付费即可。 为避免造成您的损失，如果您不想再使用此产品，则需要您在应用管理页面内删除应用，系统才会正式停止计费。

云原生API网关实例按需计费具体计费规则如下：

实例规格	主机类型	按需（元/每小时）
基础版	X86-通用型	0.8
标准版-1	X86-通用型	1.9
标准版-2	X86-通用型	3.7
标准版-3	X86-通用型	7
企业版-1	X86-通用型	14
企业版-2	X86-通用型	27.6
企业版-3	X86-通用型	41.2
铂金版-1	X86-通用型	54.8
铂金版-2	X86-通用型	109.1
铂金版-3	X86-通用型	163.5
铂金版-4	X86-通用型	217.9

包年包月

包年包月是预付费模式，按照包年包月的方式进行付费购买。 只要您实际接入的实例数不超过您购买的实例数，不会造成额外的费用。 包年包月的模式下，您需要在到期前进行续费或选择自动续费的方式，确保产品持续可用。

云原生API网关实例包年包月计费规则如下：

实例规格	主机类型	包年包月（元/每月）
基础版	X86-通用型	416

实例规格	主机类型	包年包月（元/每月）
标准版-1	X86-通用型	975
标准版-2	X86-通用型	1,866
标准版-3	X86-通用型	3,531
企业版-1	X86-通用型	7,069
企业版-2	X86-通用型	13,921
企业版-3	X86-通用型	20,773
铂金版-1	X86-通用型	27,625
铂金版-2	X86-通用型	55,033
铂金版-3	X86-通用型	82,442
铂金版-4	X86-通用型	109,850

系列2：AI网关

按需付费

按需付费是后付费模式，您只需按实际情况进行使用，然后按照使用账单付费即可。 为避免造成您的损失，如果您不想再使用此产品，则需要您在应用管理页面内删除应用，系统才会正式停止计费。

AI网关实例按需计费具体计费规则如下：

实例规格	主机类型	按需（元/每小时）
标准版-1	X86-通用型	7.93
标准版-2	X86-通用型	15.18
标准版-3	X86-通用型	28.73
企业版-1	X86-通用型	56.10
企业版-2	X86-通用型	110.48
企业版-3	X86-通用型	164.86
铂金版-1	X86-通用型	213.77
铂金版-2	X86-通用型	425.85
铂金版-3	X86-通用型	637.94
铂金版-4	X86-通用型	850.03

包年包月

包年包月是预付费模式，按照包年包月的方式进行付费购买。 只要您实际接入的实例数不超过您购买的实例数，不会造成额外的费用。 包年包月的模式下，您需要在到期前进行续费或选择自动续费的方式，确保产品持续可用。

AI网关实例包年包月计费规则如下：

计费说明

实例规格	主机类型	包年包月（元/每月）
标准版-1	X86-通用型	3997.50
标准版-2	X86-通用型	7650.60
标准版-3	X86-通用型	14477.10
企业版-1	X86-通用型	28276.00
企业版-2	X86-通用型	55684.00
企业版-3	X86-通用型	83092.00
铂金版-1	X86-通用型	107737.50
铂金版-2	X86-通用型	214628.70
铂金版-3	X86-通用型	321523.80
铂金版-4	X86-通用型	428415.00

订购与退订

订购

进入天翼云控制中心，搜索云原生API网关，点击购买图标，即可进入到订购配置页面。

退订

退订规则

产品退订相关规则详细参见：退订规则说明。

退订操作

1. 进入云原生API网关控制台，在左侧导航栏中单击云原生API网关-实例，进入实例列表。
2. 在列表中针对某个实例点击退订按钮。
3. 跳转平台退订页面，在您确认退订后，单击确定按钮。
4. 退订后，可在个人中心查看退订订单状态及金额情况。

欠费与续费

续订规则

产品续订规则详细参见：续订规则说明。

按需计费

当您的账户余额不足以支付账单金额，且云原生API网关实例处于欠费状态时实例状态显示为欠费冻结，系统将限制您访问欠费的实例且无法通过API进行应用的变更等操作。

在欠费期间，云原生API网关不会对您的应用实例及数据做任何变动，但会持续进行计费。

您可以通过账号充值的方式，恢复正常使用。

注意

云公司将保留该实例资源、继续存储客户的数据十五（15）日（即自操作权限被暂停之日的暂停开始时刻至第十五（15）日相同时刻为期限届满）；如前述十五（15）日期间届满仍未充值、缴纳足额服务费用，云公司有权在前述期间届满时立即释放客户的实例资源，并删除实例数据。

包年包月计费

当您云原生API网关实例的包年包月订单即将到期，且未选择自动续订时，您可以进行手工续订操作。到期的实例状态更新为已到期。

注意

服务期限届满后，云公司将保留该实例资源、继续存储客户的数据十五（15）日（即自操作权限被暂停之日的暂停开始时刻至第十五（15）日相同时刻为期限届满）；如前述十五（15）日期间届满仍未续订和续费，云公司有权在前述期间届满时立即释放客户的实例资源，并删除实例数据。

续订操作

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例，在列表点击续订。
4. 跳转至实例续订页，选择续订时长，提交订单，完成续费。

容量说明

API网关

容量阈值

以下为不同网关规格下的容量阈值。当网关容量指标处于警戒水位以下时，可以得到完整的ELB保障。对于核心业务，建议将网关容量指标控制在安全水位以下，从而获得更好的稳定性。

安全水位：能够在突发流量增长至双倍的情况下，依然确保网关系统维持高吞吐量和低延迟性能。

警戒水位：当水位达到警戒线以上时，网关的延迟可能会增加，并且在突发流量下可能存在稳定性风险。

基础版网关，仅限测试场景使用。请确保线上业务使用部署了多个节点的网关规格。

网关规格		基础版	标准版-1	标准版-2	标准版-3	企业版-1	企业版-2	企业版-3	铂金版-1	铂金版-2	铂金版-3	铂金版-4
客户端连接数	安全水位	14000	28000	56000	112000	224000	448000	660000	880000	1760000	2640000	3080000
	警戒水位	28000	56000	112000	224000	448000	896000	1320000	1760000	3520000	5280000	6160000

网关规格		基础版	标准版-1	标准版-2	标准版-3	企业版-1	企业版-2	企业版-3	铂金版-1	铂金版-2	铂金版-3	铂金版-4
HTTPS每秒新建连接数	安全水位	500	1000	2000	4000	8000	16000	24000	32000	64000	96000	120000
	警戒水位	1000	2000	4000	8000	16000	32000	48000	64000	128000	192000	240000
CPU使用率	安全水位	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%
	警戒水位	60%	60%	60%	60%	60%	60%	60%	60%	60%	60%	60%
内存使用率	安全水位	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%
	警戒水位	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%	75%

QPS性能参考

网关QPS吞吐受多种因素影响，如应答大小、是否开启HTTPS、是否开启gzip等。下表是网关处于30%CPU水位的QPS悲观值（最差情况下）参考。

说明

HTTPS新建连接会占用较多CPU资源。对于瞬时大量HTTPS并发连接的业务场景，请参考下表中HTTPS短连接的数据评估网关容量。

网关规格				基础版	标准版-1	标准版-2	标准版-3	企业版-1	企业版-2	企业版-3	铂金版-1	铂金版-2	铂金版-3	铂金版-4
连接类型	应答字节数（KBytes）	是否启用Https	是否启用gzip	CPU处于安全水位（30%）的QPS参考										
短连接	1	否	否	1900	3800	7600	15200	30400	60800	91200	121600	243200	364800	456000
		是	否	600	1200	2400	4800	9600	19200	28800	38400	76800	115200	144000
长连接	1	否	否	2500	5000	10000	20000	40000	80000	120000	160000	320000	480000	600000
		是	否	2250	4500	9000	18000	36000	72000	108000	144000	288000	432000	540000
	10	否	否	2000	4000	8000	16000	32000	64000	96000	128000	256000	384000	480000
		是	否	1900	3800	7600	15200	30400	60800	91200	121600	243200	364800	456000

AI网关

对于AI 网关实例，根据不同实例规格的QPS、客户端连接数的性能差异，提供不同的实例规格。

以下是不同网关实例规格下各项参数详情（CPU处于安全水位（30%）时的参考）。

网关规格	QPS	最大客户端连接数	最大HTTPS每秒新建连接数
标准版-1	1600	21000	700
标准版-2	3200	42000	1400
标准版-3	6400	84000	2800
企业版-1	12800	168000	5600
企业版-2	25600	336000	11200
企业版-3	38400	504000	16800
铂金版-1	51200	672000	22400
铂金版-2	102400	1344000	44800
铂金版-3	153600	2016000	67200
铂金版-4	192000	2520000	84000

按量付费与包年包月互转

概述

根据业务的使用情况，用户可以灵活的调整云原生API网关的计费方式。如果您计划的云原生API网关使用量较多且超出一个月，建议您使用更加优惠的付费模式（包年包月）。

按量付费转包年包月

只有按量付费且业务状态为运行中的实例才可以进行此操作。该操作提交后，待平台处理完成后，实例将立即按包年包月方式计费、管理。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例，在列表点击转包周期。
4. 跳转至实例转包周期订单页，选择购买时长，提交订单并完成付费。
5. 返回实例列表，刷新实例列表，计费模式变更为包年包月。

包年包月转按量付费

只有包年包月且业务状态为运行中的实例才可以进行此操作。该操作提交后，待平台处理完成后，实例将在原本包年包月到期后转为按量计费、管理。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例，在列表点击转按需。

4. 跳转至用户费用中心-订单管理-续订管理页，通过实例id查找相关到资源，在操作项中点击到期转按需，确认订单并提交。
5. 在实例原本的包年包月时间到期后将自动转换为按需计费模式。

变更实例规格

背景信息

云原生API网关包年包月模式和按量付费模式支持扩容和缩容。关于云原生API网关实例规格变更的计费详情，请参见计费说明。

变更限制

1. 基础版暂不支持进行扩容，或者从其他规格缩容到基础版。
2. 当前版本暂只支持在同系列下进行扩容或者缩容，如当前实例为标准版1时可进行扩容到标准版2、标准版3；当前实例为标准版2时可进行缩容到标准版1。

操作步骤

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择实例，并在顶部菜单栏选择地域。
3. 在实例列表页面，从待变更的网关实例操作中选择扩容或者缩容。
4. 跳转至扩容或者缩容页后，选择可选目标规格，确认订单、提交并完成付费。
5. 返回实例列表，查看实例业务状态变更为“扩容中”或者“缩容中”，待变更结束后，实例业务状态恢复至“运行中”，此时可恢复对实例的操作。

重要

- 网关实例规格变更持续时间5~10分钟左右，期间无法在控制台对该实例进行任何操作。
- 建议在业务量较少时进行升级，避免升级对业务造成影响。

API网关

通过HTTP API访问容器服务中的应用

概述

当您的容器服务中的应用需要通过外部访问时，可以通过创建HTTP API并配置路由，实现应用的互联网访问。本文以容器服务CCE为例，介绍如何通过云原生API网关实现微服务的外部访问。

前提条件

1. 已具备云容器引擎CCE实例，参见创建一个CCE应用集群。
2. 部署微服务demo到云容器引擎CCE实例，参见创建工作负载及服务或者使用容器镜像服务发布容器应用。

方案概览

通过创建云原生API网关实例，将云原生API网关与需要暴露的容器服务进行关联，在网关中设置API的路由规则，确保请求能够正确地路由到对应的容器服务，配置完成后，客户端即可通过API网关访问容器中的应用。

1. 新建云原生API网关实例：根据已有微服务环境，创建云原生API网关实例。
2. 创建服务来源：在云原生API网关中添加服务来源，根据实际情况选择云容器引擎CCE。
3. 添加服务：云原生API网关能够根据云容器引擎CCE来源获取服务的命名空间，将已有的服务添加到云原生API网关，作为备选服务。
4. 创建HTTP API：为网关实例创建HTTP API。
5. 创建路由：为该服务添加路由策略并发布。
6. 路由调试：测试微服务路由功能。

步骤一：新建云原生API网关实例

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在概览页，点击新建实例；或者在左侧导航栏，选择实例，单击新建实例。
3. 跳转至订购页，选择相关配置，然后点击下一步。
4. 跳转至配置总览页，确认配置信息，点击提交订单。
5. 跳转至支付页，完成费用支付。
6. 返回云原生API网关控制台，左侧导航栏选择实例，刷新列表查看创建的网关信息和状态。
7. 当网关信息和创建时一致，且状态为运行中，则表示网关创建成功。

步骤二：创建服务来源

1. 在左侧导航栏的实例页面中，在点击左侧导航栏服务页。
2. 点击服务来源标签页，点击创建来源，选择容器服务来源，选择目标云容器引擎实例集群。
3. 点击确认，完成添加。

步骤三：创建服务

1. 在左侧导航栏的实例页面中，在点击左侧导航栏服务页。
2. 点击服务标签页，点击添加服务，选择容器服务来源，选择目标服务所在命名空间，选择目标目标服务。

3. 点击确认，完成添加。

步骤四：创建HTTP API

1. 在左侧导航栏的API页面中，单击创建API。
2. 单击HTTP API卡片中的创建按钮，在创建HTTP API面板中配置API名称进行创建。

步骤五：创建路由

1. 在左侧导航栏的API页面中，单击目标API名称。
2. 单击创建路由，在创建路由面板，配置相关参数。

步骤六：路由调试

1. 在左侧导航栏的API页面中，单击目标API名称。
2. 在路由列表中，单击目标路由httpbin-demo操作列下的调试，进入调试页面。
3. 接口参数输入version，单击发送，可看到服务接口的返回结果如下所示。

REST API生命周期管理

概述

在微服务架构中，API 是系统集成与对外服务的关键接口。云原生 API 网关通过提供标准 REST API，支持从API设计到发布再到下线的全流程管理，使开发、测试、运维等各角色都能以统一方式高效管理接口。

前提条件

已创建云原生API网关实例，具体操作，请参见创建网关实例。

API生命周期管理

REST API支持通过控制台或者基于OpenAPI导入的方式创建。

步骤一：创建API

基于控制台创建REST API

1. 登录云原生API网关控制台。先在顶部菜单栏选择 "地域"，然后在左侧导航栏选择 "API"。
2. 单击 "创建API"。
3. 单击 "REST API" 卡片中的 "创建"，在 "新增API(REST)" 弹出框中配置相关参数，单击 "确定"。

配置项	说明
API名称	自定义创建的API名称 注意 API名称需要保证唯一。
API协议	API允许接受的HTTP协议
Base Path	API的基本路径，访问具体接口时，完整路径为 http(s)://{##}/{BasePath}/{##Path}

配置项	说明
版本管理	是否启用API版本管理能力，不同版本的API拥有相同的API名称，但参数定义、接口配置可以不同，编辑当前API版本的参数定义、接口配置和策略，不会影响其他版本。不同版本的API可以视为独立的API，访问时需要指定版本标识符 开启版本管理功能后，需要配置使用方式
使用方式	支持Path、Query、Header三种方式 • 使用Path时，需要完整访问路径为：/{##}/{BasePath}/###/#### • 使用Query时，完整访问路径为：/{##}/{BasePath}/####，请求参数中需要配置参数项添加Query为版本号 • 使用Header时，完整访问路径为：/{##}/{BasePath}/####，请求头中需要配置参数项添加Header为版本号
描述	填写API的相关描述

通过导入OpenAPI文件创建REST API

1. 首先按照上文中通过控制台创建REST API步骤创建一个REST API。
2. 单击刚才新建成功的 "API名称"，进入 API详情页面，单击右上角的 "更多操作"，单击 "导入"。
3. 在弹出框中单击 "从文件导入"，选择待导入的OpenAPI文件，然后单击 "预检并创建"。
4. 选择 "合并逻辑"，各选项说明如下：

选项	说明
智能合并	将在已存在的API基础之上，创建新增的接口，更新重复的接口，但不删除仅在原API中存在的接口
仅导入新增接口	将在已存在的API基础之上，仅创建新增的接口，对重复或原有的接口不做处理
覆盖当前API	基于当前导入的文件重新创建API，完全覆盖已有API

5. 如果预检结果为失败，需要您修改配置文件并重复上述步骤。如果预检结果为成功，可以单击 "创建API" 根据选择的合并逻辑进行API导入 。

步骤二：添加接口

1. 在REST API中，单击 "添加接口" 。
2. 在 "创建接口" 弹出框中配置相关参数，单击 "保存" 。

配置项	说明
接口名称	自定义创建的接口名称，在API下需要唯一
接口Path	接口的具体路径
方法	接口的请求方法。接口的路径+接口的方法，需要在API下唯一
描述	接口的描述信息

配置项	说明
请求定义	支持定义 "Header" 、"Query"、"Parameter Path" 参数以及 "Body" 参数 其中Path参数支持在接口Path中按照如下方式进行变量定义： • /books/:bookId 说明 请求定义仅用于文档展示和生成，不对运行时进行校验。
响应定义	定义不同响应码的数据结构 说明 响应码定义仅用于生成文档，不对运行时进行校验。
Mock	Mock配置仅在API发布Mock场景下生效

步骤三：发布 API

1. 录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上方的 "发布API" 。
3. 在发布API弹出框中配置相关参数，然后单击 "发布" 。

配置项	说明
域名	选择域名进行发布，发布后，可以通过域名访问API
所属实例	选择所创建的云原生API网关实例。不同的业务环境可用不同的实例区分，实现API在多环境上的发布
使用场景	使用场景包括基础场景和灰度场景两类 基础场景 • Mock：接口响应将返回接口定义中的Mock配置，若接口未定义Mock配置，则将无法访问该接口 说明 发布Mock场景时，要求当前API中至少有一个接口开启了Mock配置，否则将发布失败。 • 单服务：所有流量请求将转发到某一具体的后端服务，这个场景为最常使用的场景 **灰度场景** • 按比例（多服务）：流量将按比例分发到对应的后端服务中，常用于切流及灰度发布场景 说明 要求服务权重之和等于100。
后端服务	关联该网关实例下的后端服务
发布描述	填写API的发布描述

步骤四：调试 API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。
3. 在 "API设计" 标签页的 "接口列表"栏，选择需要调试的接口，单击右侧的 "调试"。
4. 在 "接口调试" 弹出框中，配置相关参数，然后单击 "发送请求" 进行调试。

步骤五：添加 API 版本

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API，单击右上角的 "更多操作" > "添加版本"，完成相关配置：

配置项	说明
使用方式	支持Path、Query、Header三种方式 • 使用Path时，需要完整访问路径为：/{##}/{BasePath}/###/#### • 使用Query时，完整访问路径为：/{##}/{BasePath}/###，请求参数中需要配置参数项添加Query为版本号 • 使用Header时，完整访问路径为：/{##}/{BasePath}/###，请求头中需要配置参数项添加Header为版本号

3. 在添加完成后，您可以单击页面上方的 "版本" 下拉框选择版本切换。

步骤六：导出 API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "导出"。

查看发布历史

说明

发布历史保留最近10次。

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API，然后单击 "发布历史" 标签页。
3. 单击 "版本" 和 "实例下拉框" 可以选择不同API版本和实例下的发布历史。
4. 单击目标历史版本 "操作" 列下的 "查看"，可查看历史版本详情。
5. 单击目标历史版本 "操作" 列下的 "切换至该版本"，可将当前API的在当前实例上的已发布版本切换到选定版本。

步骤七：下线 API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "下线"。

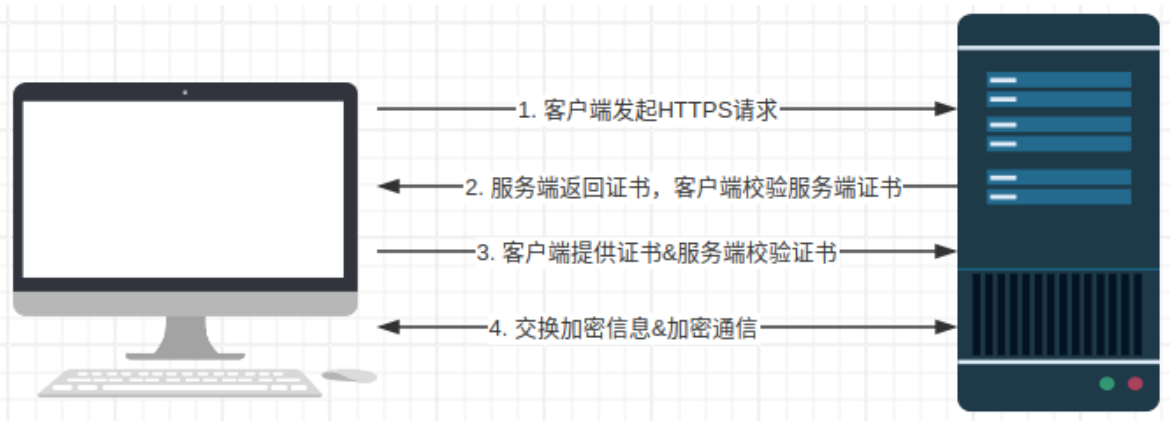
步骤八：删除 API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "删除" 。

使用云原生API网关实现后端双向TLS认证

概述

云原生API网关作为代理服务接收下游服务（DownStream）请求，并转发到上游服务（Upstream）；对于上游服务来说，云原生API网关收敛了外部的请求，统一经过网关转发到上游。从安全角度考虑，上游服务可以添加认证鉴权规则实现对网关的身份认证和访问鉴权，确保上游服务安全性；其中一种实现方式就是在上游服务配置TLS策略，要求网关转发请求时提供客户端证书，实现对网关的身份认证，具体流程如下：



云原生API网关支持配置证书，用于实现请求后端服务时服务端对网关的证书认证，本文说明具体配置流程。

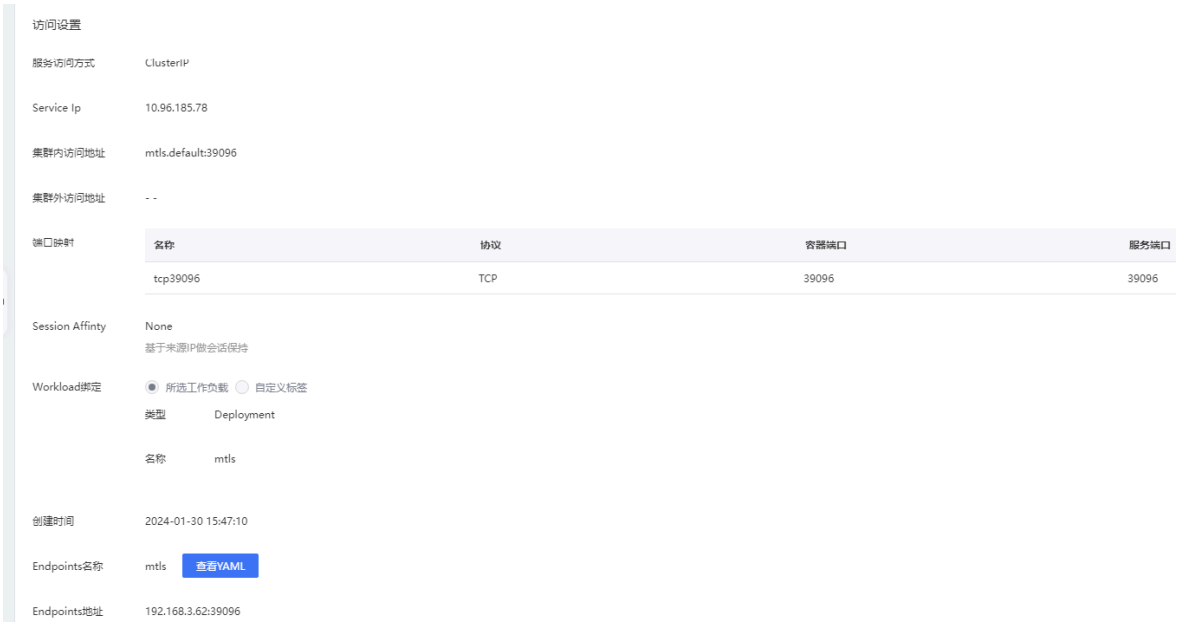
前提条件

1. 部署后端HTTPS服务，并配置要求请求客户端提供证书。
2. 已开通云原生API网关实例。

部署后端服务

我们采用云容器引擎部署后端服务，镜像采用我们的demo应用（已配置证书，并要求客户端请求时提供证书），部署完成后在云容器引擎控制台可以看到容器启动：

还需要部署关联到工作负载的Service，用于暴露工作负载，如下



添加路由配置

使用云原生API网关配置路由转发

1. 首先进入云原生API网关控制台 > 实例 > 服务 > 服务来源标签页，点击添加来源，选择容器服务来源类型，选择我们部署了后端服务的云容器引擎集群，添加完成。
2. 进入云原生API网关控制台 > 实例 > 服务 > 服务列表-> 创建服务 功能，选择从容器创建后端服务，选择我们部署的命名空间和服务，服务协议选择HTTPS（暂时先关闭mTLS选项）。
3. 进入云原生API网关控制台 > API > 创建HTTP API > 进入目标API > 添加路由，配置路由转发规则（匹配路径/api/1/reviews，请求转发到上面创建的服务）> 保存并发布。

结果验证

通过云原生API网关访问路由 http://192.168.4.96:27151/api/1/reviews

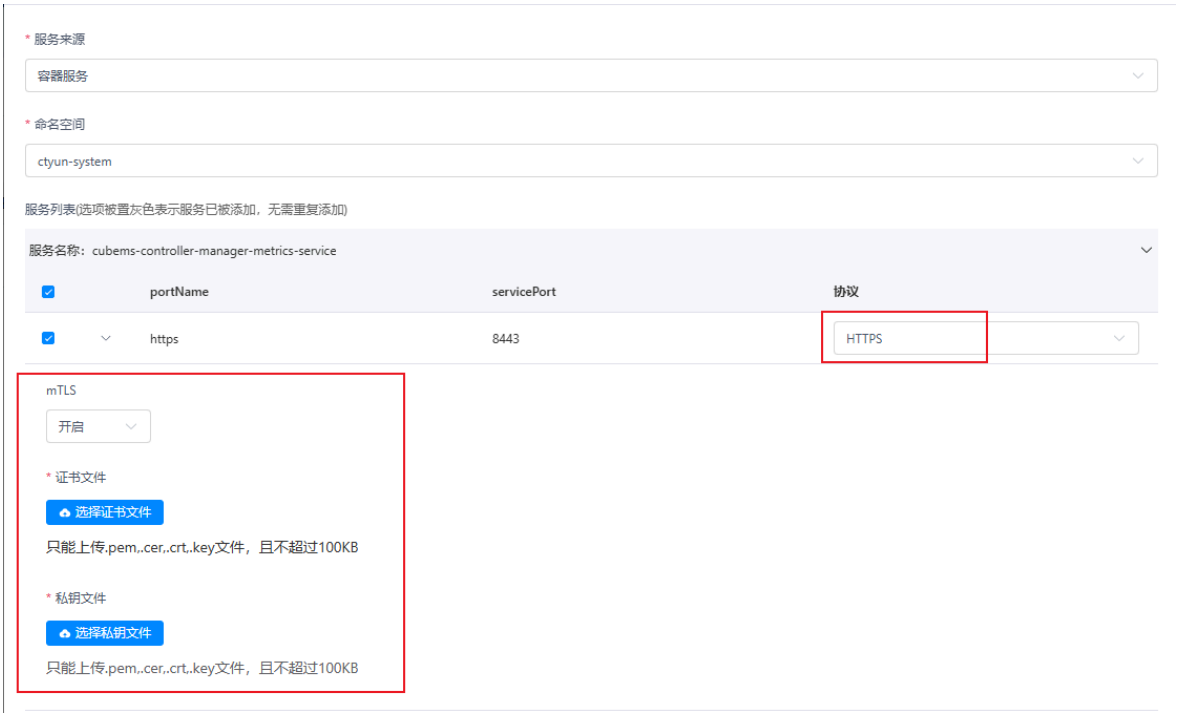
```
curl http://192.168.4.96:27151/api/1/reviews -sv
*   Trying 192.168.4.96:27151...
* Connected to 192.168.4.96 (192.168.4.96) port 27151 (#0)
> GET /api/1/reviews HTTP/1.1
> Host: 192.168.4.96:27151
> User-Agent: curl/7.71.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 502 Bad Gateway
< Date: Tue, 30 Jan 2024 08:07:10 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 154
< Connection: keep-alive
<
<html>
<head><title>502 Bad Gateway</title></head>
<body>
<center><h1>502 Bad Gateway</h1></center>
<hr><center>openresty</center>
</body>
</html>
* Connection #0 to host 192.168.4.96 left intact
```

可以看到请求报错了，后台查看debug日志可以看到错误信息如下（SSL握手报错）：

sslv3 alert bad certificate:SSL alert number 42) while SSL handshaking to upstream

mTLS证书配置&验证

进入服务列表菜单，编辑刚才创建的服务，开启服务mTLS认证并上传相关证书。



保存后再次请求接口，可以看到返回了正确的结果：

```
curl http://192.168.4.96:27151/api/1/reviews -sv
*   Trying 192.168.4.96:27151...
* Connected to 192.168.4.96 (192.168.4.96) port 27151 (#0)
> GET /api/1/reviews HTTP/1.1
> Host: 192.168.4.96:27151
> User-Agent: curl/7.71.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Content-Type: application/json
< Transfer-Encoding: chunked
< Connection: keep-alive
< Date: Tue, 30 Jan 2024 08:13:57 GMT
<
* Connection #0 to host 192.168.4.96 left intact
[{"id":1,"productId":1,"reviewer":"Reviewer1","text":"This is the 1st reviewer"}]
```

使用云原生API网关实现WebSocket流量代理

概述

WebSocket协议允许客户端和服务器之间进行实时的双向数据传输，从而确保了连接的持久性和低延迟。可以在云原生API网关中创建WebSocket API，实现WebSocket流量代理。

前提

1. 已开通云原生API网关实例。
2. 已部署后端WebSocket server服务。

云原生API网关中开启WebSocket支持

我们采用在MSE Nacos中注册后端WebSocket服务的方式进行服务部署，后端服务示例可部署demo应用（暴漏WebSocket应用路径为/ws/server）。

创建注册中心服务

前置条件

已开通云原生API网关实例和同VPC下的注册配置中心实例（Nacos引擎），已添加注册配置中心实例作为云原生API网关的服务来源。

后端服务已部署并注册到注册配置中心。

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择容器服务来源，选择要添加的服务所在的命名空间。
7. 在服务列表中选择要添加目标服务，请求协议设置为WebSocket。

注册中心服务配置说明

配置	说明
命名空间	后端服务注册到注册配置中心的命名空间
请求协议	后端服务的协议，支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
mTLS	是否打开后端服务双向TLS认证（后端服务对网关证书认证）
证书文件	后端服务开启双向TLS认证后，云原生网关提供的证书文件
私钥文件	后端服务开启双向TLS认证后，云原生网关提供的私钥文件

创建 WebSocketAPI

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。

2. 单击 "创建API"，然后单击WebSocket API卡片中的 "创建"。
3. 在 "创建WebSocket API" 面板中配置相关参数，单击 "确定"。

配置项	说明
API名称	定义创建的API名称，API名称需要保证唯一
描述	填写API相关的描述

创建 WebSocket路由配置

路由请求path可设置为“/*”

WebSocket API下的路由管理方式和HTTP API下路由管理方式基本一致，可以参考相关文档。

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 选择已创建的HTTP API，单击API名称进入API详情页。
3. 单击 "创建路由"。
4. 在 "创建路由" 弹出框填写路由相关配置，并单击 "保存" 或 "保存并发布" 按钮，如您单击的是 "保存" 按钮，则需要在 "路由列表" 页，单击操作列 "发布" 按钮发布路由；

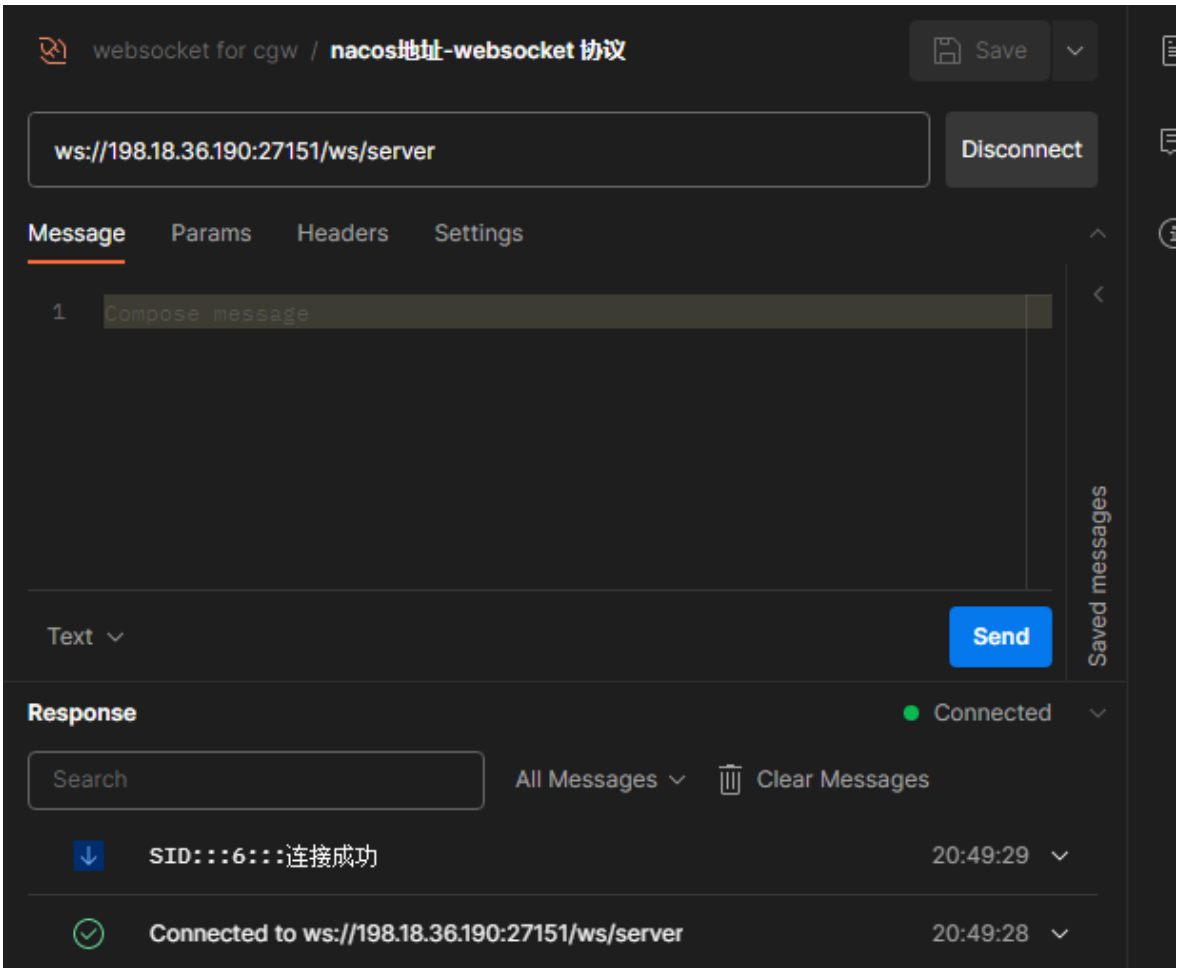
路由配置的规则之间是“与”的关系，必须全部满足才算匹配，路由配置项说明如下：

参数	说明
路由名称	路由名称，用于标识一条路由规则。API内需要保证唯一
描述	路由描述
域名	用于和请求中的域名进行匹配，不填则任何请求都可以匹配；可选项从域名管理中添加的域名选择
路径	支持两种匹配方式： 等于：精确匹配。 前缀是：前缀匹配，如配置 /test，则能够匹配所有路径以 /test开头的请求
方法	匹配请求中的HTTP方法
优先级	当多个路由同时匹配一个请求时，路径匹配深度较大的路由优先；路径匹配相同的情况下，路由优先级高（数字大）的优先匹配
请求头（header）	匹配请求中的HTTP header
请求参数（Query）	匹配请求中的HTTP query参数
Cookie	通过Cookie进行路由匹配，多个参数之间是“与”的关系
是否启用参数规整化匹配	启用后支持对参数进行取模，并根据取模结果进行精确或者范围匹配
参数类型	启用参数规整化匹配选择，支持Header、Query、Cookie
是否Hash	启用参数规整化匹配选择，是否对参数进行哈希处理后再取模，哈希函数为Java String hashCode
取模数值	启用参数规整化匹配选择，自定义填写取模数值

参数	说明
标记类型	支持精确和匹配精确精确匹配，可用英文逗号分隔多个精确值范围：最大值和最小值均是闭区间，[min,max]
所属实例	路由规则所属的实例
场景	当前支持单服务、多服务、标签路由、mock路由、重定向和dubbo代理
后端服务	根据选择的场景选择请求需要转发的后端服务

结果验证

可通过postman软件发起到网关的WebSocket请求，请求协议前缀为ws://或wss://。请求成功。



AI网关

通过Model API访问大模型服务

概述

针对大模型服务访问场景，Model API提供高度灵活和智能的路由配置与调试能力，内置丰富的路由插件，提供消费者鉴权、限流熔断和AI可观测等能力。本文主要介绍如何通过Model API访问大模型服务。

前置条件

1. 已创建AI网关实例，具体操作，请参见创建网关实例。
2. AI网关实例与大模型服务网络已打通。如大模型服务为公网服务，需为AI网关实例所在VPC创建公网NAT网关，请参考NAT网关文档。

创建大模型服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。
3. 配置基本信息。

   在弹窗中先选择服务来源为“LLM服务”，随后编辑其余配置。

   - 服务名称： 自定义服务名称。
   - 大模型供应商： 支持息壤、DeepSeek、OpenAI兼容（OpenAI Compatible）、百炼。
   - 服务地址(base_url)：大模型服务的BaseURL。
   - API-KEY：访问大模型需要的API-KEY凭证。API-KEY的获取请咨询对应服务供应商。
4. 配置完成后单击确定，完成创建。

创建Model API

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击Model API，然后单击创建Model API。
3. 选择使用场景，并单击对应的创建按钮。

   不同场景对应的协议和系统自动创建的默认路由可能不同，当前支持文本生成、图片生成、文本排序（Rerank）、向量化（Embedding）。

4. 配置基本信息。

- API名称：自定义API名称，支持中文、英文、数字、下划线“_”、“-”，且不超过64个字符。
- 协议：每个协议对应该场景下的一组默认路由。
- 路由：协议对应的默认路由。
- BasePath：API的基本请求路径，默认为/。默认启用转发至后端服务时移除。

说明 当转发至后端服务时移除开启后，请求转发至后端服务，系统会自动移除请求部分中的BasePath部分。如：

```
# BasePath设置为/api
# 原始请求路径为/api/chat/completions
# 实际转发到后端的路径为/chat/completions
```

- 域名：访问API的域名，支持同时配置多个域名。
- 描述：API的描述信息，最长不超过256个字符。
- 后端服务：场景支持单模型服务、多模型服务。单模型服务：选择一个大模型服务，支持设置模型名称或透传模型名称；多模型服务：选择多个大模型服务并设置权重，支持设置模型名称或透传模型名称。

5. 确认配置参数并单击确定完成创建。

访问Model API

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击概览，然后单击接入点，获取AI网关入口地址。
3. 以OpenAI 兼容协议为例，通过AI网关入口地址访问Model API，如：

```
curl --location 'http://{网关入口地址}/{Base Path}/chat/completions' \
 --header 'Content-Type: application/json' \
 --data '{
 "stream": false,
 "model": "deepseek-chat",
 "messages": [
  {
   "role": "system",
   "content": "You are a helpful assistant."
  },
  {
   "role": "user",
   "content": "你是谁？"
  }
 ],
 "temperature": 0.7,
 "top_p": 1,
 "max_tokens": 1024
}'
```

访问MCP服务

概述

MCP（Model Context Protocol）是一种开源协议，旨在实现大语言模型与外部数据源和工具的集成，用来在大模型和数据源之间建立安全双向的连接。本文将详细阐述MCP服务的配置与使用方法。

前置条件

1. 已创建AI网关实例，具体操作，请参见创建网关实例。
2. AI网关实例与MCP服务网络已打通。如MCP服务为公网服务，需为AI网关实例所在VPC创建公网NAT网关，请参考NAT网关文档。

创建MCP服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击MCP管理-MCP服务，然后单击创建MCP服务。
3. 配置MCP服务基本信息。

参数	说明	示例
名称	自定义MCP服务名称	mcp-demo
协议	选择创建的服务协议	MCP服务直接代理
描述	创建的服务详细描述	
后端服务		
服务名称	选择后端目标服务，该服务需通过服务页面创建	mcp-service
MCP Transport	MCP传输协议，选择SSE或Streamable HTTP	Streamable HTTP
路径	后端目标服务的路径	
MCP接入点		
域名	访问MCP服务使用的域名，支持选择多个域名	www.testmcp.com
路径	MCP服务接入点的路径，会根据MCP Transport和服务名称生成	/mcp-servers/mcp-demo/sse

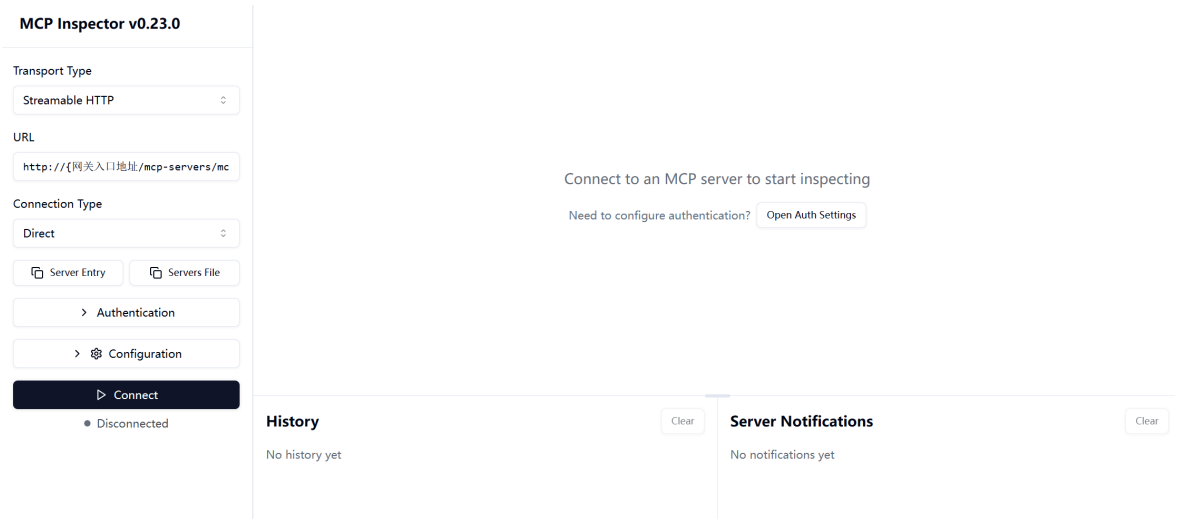
4. 配置完成后单击确定，完成创建。

通过MCP Inspector访问MCP服务

说明 MCP Inspector 是专为MCP服务器设计的交互式调试工具，支持开发者通过多种方式快速测试与优化服务端功能。

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击概览，然后单击接入点，获取AI网关入口地址。
3. 安装Node.js环境。
4. 安装MCP Inspector，安装命令：npx kargnas-mcp-inspector@latest。
5. 通过浏览器登录MCP Inspector界面，如：http://localhost:6274/?MCP_PROXY_AUTH_TOKEN={token}。

6. 配置连接信息，填写Transport Type、URL等信息。



7. 点击Connect进行连接，连接成功后可以获取MCP服务的所有工具。

API网关

实例管理

创建网关实例

概述

云原生API网关支持多种服务发现方式（如云容器引擎CCE、注册配置中心RCC-Nacos引擎、注册配置中心RCC-Eureka引擎、函数计算FaaS等），并集成安全运维能力。本文介绍如何创建云原生API网关实例。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在概览页，点击新建实例；或者在左侧导航栏，选择实例，单击新建实例。
3. 跳转至订购页，选择相关配置（参见下方配置项说明），然后点击下一步。
4. 跳转至配置总览页，确认配置信息，点击提交订单，
5. 跳转至支付页，完成费用支付，
6. 返回云原生API网关控制台，左侧导航栏选择实例，刷新列表查看创建的网关信息和状态。
7. 实例创建大约需要5~10分钟，当网关信息和创建时一致，且状态为运行中，则表示网关创建成功。

实例配置说明

配置项	描述
计费模式	支持包年包月和按需计费方式，费用说明请参照购买指南->计费说明。
购买时长	可以根据实际需求进行选择，支持1个月、2个月、3个月、4个月、5个月、6个月、1年。
自动续期	您可以选择开启自动续期，避免云原生API网关到期后无法使用。
自动续期购买时长	开启自动续期，可以选择续期时长，支持1个月、2个月、3个月、4个月、5个月、6个月、1年。
部署方式	实例节点将按单可用区、多可用区部署方式，分布在单个或者多个可用区中。多可用区部署可增强实例的容灾能力。注意：基础版规格不具备高可用、多AZ容灾能力。
可用区	选择单可用区部署方式时，可用区可任选其一；选择多可用区部署时，必须选择至少3个可用区。
CPU架构	部署实例的主机架构。
主机类型	部署实例的主机类型。
网关规格	参见产品简介-产品规格介绍
虚拟私有云	选择虚拟私有云，若您还没有虚拟私有云，请参照创建虚拟私有云。
所在子网	择所在子网，若您还没有所在子网，请参照创建所在子网。

配置项	描述
启用ipv6	若子网已开启ipv6访问，在此处可选择启用ipv6。未启用ipv6时，将通过ipv4访问。
安全组	选择安全组，若您还没有可用安全组，请参照创建安全组。
实例名称	自定义实例名称，不可重复；实例名称长度4~40个字符，大小写字母开头，只能包含大小写字母、数字及分隔符(-)，大小写字母或数字结尾。
企业项目	网关实例关联的企业项目，可以到IAM控制台创建企业项目。
指标监控	启用后，可在控制台观测分析中查看系统和API的流量、成功率、延迟等监控指标，若您还没有开通应用性能监控产品，可先点击提示链接前往开通。
链路追踪	可选择采集百分比启用，启用后，可在控制台观测分析中查看API请求的链路追踪信息。您可通过委托授权的方式开通此服务。
云日志服务	启用后，可在控制台观测分析中查看访问日志。您可通过委托授权的方式开通此服务。

查看网关详情

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择实例，点击目标实例名称或实例id，进入实例详情。
3. 跳转至实例概览页，查看实例信息、接入点、可观测信息、策略配置等。

实例信息

基本信息

查看实例ID、名称、付费类型、创建时间、更新时间等信息

运行信息

查看实例的业务状态、运行状态、实例规格等信息

网络信息

查看实例的可访问端口、地区、可用区、VPC、子网、安全组等信息

实例节点

查看实例节点列表，包括vpc的ipv4、ipv6、http端口、https端口、分布可用区等信息

接入点

可查看当前实例绑定的弹性负载均衡ELB实例列表

可观测信息

查看实例的链路追踪、日志投递配置项

策略配置

查看实例的策略配置项，如限流、跨域等策略

消费者认证

查看实例授权的消费者列表

网关自定义响应

查看实例配置的响应类型详情

查看网关监控

概述

云原生API网关的监控分为：监控分析和资源监控；以下将分别介绍这两种信息。

监控分析

监控分析的类别

1. 流量监控，如出入流量、QPS等。
2. 请求监控，如请求成功率、请求失败率，404比例、5XX比例等。
3. 延迟监控，如平均延迟、P50延迟、P95延迟、P99延迟等。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择实例，点击目标实例名称或实例id，进入实例详情。
3. 在左侧导航栏，展开观测分析，点击监控分析，查看监控分析信息。

资源监控

监测的主要是节点资源，如CPU使用率、内存使用率、系统盘使用率等信息

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择实例，点击目标实例名称或实例id，进入实例详情。
3. 在左侧导航栏，展开观测分析，点击资源监控，可切换实例节点，查看对应节点的资源监控信息。

修改网关名称

概述

网关名称是网关实例的别名，不影响业务，允许用户灵活修改。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择实例，点击目标实例名称或实例id，进入实例详情。
3. 在实例信息-基础信息中，点击名称旁边的修改按钮；
4. 在弹窗中填写新的名称。实例名称长度4~40个字符，大小写字母开头，只能包含大小写字母、数字及分隔符(-)，大小写字母或数字结尾。点击确认提交修改操作。
5. 弹窗展示任务处理情况，任务完成表示修改成功。

开启IPV6访问

概述

云原生API网关支持通过ipv6地址访问。

前提条件

网关实例所在的vpc子网需要开启ipv6配置

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择实例，在实例列表目标实例的更多操作中点击开启IPv6访问。
3. 阅读弹窗提示（确认开启IPV6访问：test-instance，升级过程需要3-5分钟，请刷新实例列表获取实例的最新状态），点击确认。
4. 提交操作后，会进行变更检查，如果所在VPC子网未开启IPv6配置，则提示当前实例所属VPC未开启ipv6子网，此实例不支持启用ipv6，请先在此VPC上开启ipv6！

退订

退订规则

产品退订相关规则详细参见：退订规则说明。

退订操作

1. 进入云原生API网关控制台，在左侧导航栏中单击云原生API网关-实例，进入实例列表。
2. 在列表中针对某个实例点击退订按钮。
3. 跳转平台退订页面，在您确认退订后，单击确定按钮。
4. 退订后，可在个人中心查看退订订单状态及金额情况。

续订

续订规则

产品续订规则详细参见：续订规则说明。

续订影响

对业务无影响

续订操作

1. 进入云原生API网关控制台，在左侧导航栏中单击云原生API网关-实例，进入实例列表。
2. 在实例列表的某个包年包月付费模式的实例操作中点击续订按钮。
3. 跳转平台续订页面，选择续订时长，单击确定按钮。
4. 跳转支付页，完成费用支付。
5. 返回实例列表，查看实例的过期时间。

标签

概述

云原生API网关已经支持资源标签功能，提供为资源添加标签以及根据标签筛选资源的功能，本章介绍如何使用资源标签功能。

添加标签

1. 登录云原生API网关管理控制台，选择资源池。
2. 在左侧导航栏，选择实例。
3. 光标悬浮至对应实例的标签列图标，点击“添加”按钮，会显示弹出框。

4. 在弹出框中点击“添加标签”按钮，选择已有标签键值对或者创建自定义标签键值对，点击“确定”按钮即可添加对应标签。

批量绑定标签

1. 登录云原生API网关管理控制台，选择资源池。
2. 在左侧导航栏，选择实例。选择对应实例，点击批量标签>绑定标签按钮。
3. 弹出框显示已选择的实例资源，选择对应标签或填写自定义标签，点击便签右侧“确定”按钮，将对应标签加入已选择标签列表。
4. 点击弹出框右下角的确定按钮完成批量绑定标签。

批量解绑标签

1. 登录云原生API网关管理控制台，选择资源池。
2. 在左侧导航栏，选择实例。
3. 选择对应实例，点击批量标签>解绑标签按钮。
4. 弹出框显示已选择的实例资源，选择对应标签，点击解绑按钮即可批量解绑标签。

修改标签

1. 登录云原生API网关管理控制台，选择资源池。
2. 在左侧导航栏，选择实例。
3. 光标悬浮至对应实例的标签列图标，点击编辑按钮，会显示弹出框。
4. 在弹出框中，修改对应的标签键值对，点击修改按钮即可完成修改。

解绑标签

1. 解绑标签操作会解绑对应标签与实例的关联关系，但不会删除对应标签，在选择标签时依然可见该标签。
2. 登录云原生API网关管理控制台，选择资源池。
3. 在左侧导航栏，选择实例。
4. 光标悬浮至对应实例的标签列图标，点击编辑按钮，会显示弹出框。
5. 在弹出框中，点击解绑按钮即可解绑对应的标签键值对。

一键解绑标签

1. 登录云原生API网关管理控制台，选择资源池。
2. 在左侧导航栏，选择实例。
3. 光标悬浮至对应实例的标签列图标，点击编辑按钮，会显示弹出框。
4. 在弹出框中，点击一键解绑按钮即可解绑该实例与所有标签的关联关系。

删除标签

1. 删除标签操作会解绑对应标签与实例的关联并且删除对应标签，在选择标签时该标签已不存在。
2. 登录云原生API网关管理控制台，选择资源池。
3. 在左侧导航栏，选择实例。
4. 光标悬浮至对应实例的标签列图标，点击编辑按钮，会显示弹出框。
5. 在弹出框中，在对应的标签键值对行的操作项中删除按钮即可完成删除。

标签筛选

1. 标签筛选功能可以根据选择的标签筛选实例资源，每次最多同时筛选5个标签。只要实例资源拥有已选标签中的一个即会被筛选出来。

2. 登录云原生API网关管理控制台，选择资源池。
3. 在左侧导航栏，选择实例。
4. 点击标签筛选按钮，点击对应的标签键值对可将标签加入已选标签，即可根据已选标签筛选出对应实例资源。

企业项目

概述

参见IAM管理中的企业项目管理

前提条件

在IAM管理中已存在可用的企业项目，若没有请在企业项目管理中创建。

关联企业项目

目前云原生API网关只在订购实例时关联可选企业项目，参见创建网关实例-实例配置说明

添加接入点

概述

目前云原生API网关的接入点为弹性负载均衡ELB，实现网关多节点入站负载均衡和公网访问暴露，支持绑定一个或多个ELB作为网关入口，从而将访问流量自动分发到多台网关节点，实现更高水平的应用程序容错性能。

标准版及以上规格的云原生 API 网关均采用高可用部署，具备多节点架构。通过接入天翼云弹性负载均衡产品 ELB，可实现对多个网关节点的流量分发、故障自动剔除等能力，并同时支持 HTTP 和 HTTPS 协议。 对于有公网访问需求的业务场景，可通过为弹性负载均衡绑定 EIP，实现网关服务的公网访问能力。

绑定ELB

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 实例列表 。
4. 您可以在网关列表页单击需要查看的网关实例ID或者 实例名称 。
5. 进入网关详情页 **>** 接入点页。
6. 在绑定ELB面板中配置ELB参数，单击左下角确认按钮。
7. 弹窗展示绑定任务执行情况，刚绑定时关联状态为进行中，系统会自动刷新，直至关联状态更新为 绑定成功。
8. 在网关入口区域可以查看新绑定的ELB 。

ELB配置参数说明

参数	描述
类型	支持公网和私网
ELB	选择同vpc下的ELB实例，若您还没有ELB实例，请先创建ELB实例。
HTTP端口	配置HTTP端口
HTTPS端口	配置HTTPS端口

注意

80、443、8080、8443常见敏感端口需进行备案后才可配置

接入点列表参数说明

参数	描述
ELB ID	ELB的ID，单击ELB ID可以查看ELB详情
入口地址（ip）	ELB的ip地址，即网关入口的访问地址，您可以根据实际需求添加域名
HTTP端口	HTTP端口
HTTPS端口	HTTPS端口
类型	公网或私网
关联状态	网关实例与ELB的关联状态，当关联状态为绑定成功时可以正常访问
关联时间	网关实例与ELB关联的时间
操作	您可以单击解绑ELB进行解绑操作

解绑ELB

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 实例列表 。
4. 您可以在网关列表页单击需要查看的网关实例ID或者 实例名称 ，也可以单击操作列中的管理按钮。
5. 进入网关详情页 > 接入点页 。
6. 点击列表项的解绑ELB操作。
7. 弹窗展示解绑任务执行情况，刚绑定时关联状态为进行中，系统会自动刷新，直至关联状态更新为 解绑成功。
8. 在网关入口区域可以查看最新的ELB列表。

添加实例级别策略

概述

云原生API网关支持在多个维度进行多种策略配置，策略配置维度优先级由高到低，实例级 > API级 > 路由/接口级。

在实例详情页的"策略配置"标签页中可以配置路由策略，当前支持限流、重写、Header配置、跨域、Query参数设置、Cookie重写、外部认证授权、熔断、黑白名单、防重放和Fallback服务策略配置。

具体策略配置说明参见API管理-HTTP API-路由-路由策略配置。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 实例列表 。

4. 您可以在网关列表页单击需要查看的网关实例ID或者 实例名称。
5. 进入网关详情页 > 策略配置页。
6. 单击需要变更的策略，在右侧配置页中修改相关配置后，单击 "保存"。

配置实例级别消费者认证

概述

消费者配置说明参见消费者管理，实例级消费者在API级、路由/接口级消费者认证之前生效。本文介绍实例级消费者的操作步骤。

注意

1. 如果在实例上开启消费者认证，则将对该实例下的所有路由/接口都生效，请谨慎开启。
2. 尽量避免在实例级、API级、路由/接口级上使用同一种认证方式。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 实例列表 。
4. 您可以在网关列表页单击需要查看的网关实例ID或者 实例名称。
5. 进入网关详情页 > 消费者认证页。
6. 在配置信息栏点击编辑，选择启用或停用消费者认证。
   a. 若启用，可选择Key-Auth、Basic-Auth、HMAC、JWT等认证方式。
   b. 可选择从不同位置获取token。
7. 在消费者列表栏，可以为当前实例授权、解除授权多个消费者。

配置实例级别自定义网关响应

概述

您可以自定义各种http响应码的返回内容，当后端服务返回对应响应码时，网关将按照预设的响应码内容返回给客户端。实例级网关响应配置将作用于实例全局，其优先级从高到低，依次为实例级 > API接 > 路由/接口级。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 实例列表 。
4. 您可以在网关列表页单击需要查看的网关实例ID或者 实例名称。
5. 进入网关详情页 > 自定义网关响应页。
6. 点击"新增"，填写配置信息后，点击"新增"。

说明

最多支持20个不同的响应类型

服务来源

创建服务来源

概述

云原生API网关无缝对接天翼云注册配置中心（Nacos引擎、Eureka引擎）和云容器引擎，通过服务发现模块监听服务来源，动态感知后端服务。

说明

每种服务来源类型仅支持添加一个实例集群

创建云容器引擎服务来源

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务来源标签页。
5. 点击创建来源，在配置页选择容器服务来源类型，在容器集群下拉列表中选择要添加的集群（当前仅支持添加与网关实例同VPC内的容器集群）。
6. 根据需要勾选是否监听K8s Ingress选项并配置监听命名空间的标签。

创建MSE Nacos服务来源

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务来源标签页。
5. 点击创建来源，在配置页选择MSE Nacos服务来源类型，在MSE Nacos集群下拉列表中选择要添加的集群（当前仅支持添加与网关实例同VPC内的MSE Nacos集群）。

创建MSE Eureka服务来源

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务来源标签页。
5. 点击创建来源，在配置页选择MSE Eureka服务来源类型，在MSE Eureka集群下拉列表中选择要添加的集群（当前仅支持添加与网关实例同VPC内的MSE Eureka集群）。

管理服务来源

概述

在云原生API网关的服务来源列表页可以修改容器服务来源，删除MSE Nacos服务来源、MSE Eureka服务来源、容器服务来源，并重新添加。

修改容器服务来源

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务来源标签页。

5. 在列表中目标容器服务来源的操作项中点击修改，可以安装、卸载Ingress Controller。

删除服务来源

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务来源标签页。
5. 在列表中目标容器服务来源的操作项中点击删除，点击确认并等待任务执行完成，即可删除服务来源。

注意

1. 当有来自当前服务来源的服务被引用时，此服务来源不可被删除

服务

创建服务

概述

云原生API网关支持将请求转发的预定义的目标服务，本章节介绍如何创建云原生API网关中的服务。

云原生API网关中服务相关概念

概念	说明
服务	网关转发请求的目标，可以通过配置路由引用相应的服务，将请求转发到指定的目标
容器服务	从云容器引擎中发现的服务
注册中心服务	从注册配置中心发现的服务
固定地址服务	固定配置的IP+端口服务，支持添加多个服务地址
LLM服务	大模型供应商服务，提供AI模型能力
Dubbo服务	添加服务时，服务请求协议选择DUBBO
权重	1. 请求到服务节点的权重比例，表示不同节点之间的流量比例 2. 只有在优先级相同时，权重才会生效
优先级	1. 优先访问高优先级节点，只有在高优先级的节点不可用或者尝试过，才会访问一个低优先级的节点 2. 优先级默认值为0，可以取负数表示备份节点
请求协议	请求后端服务的协议，当前支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
服务版本	云原生API网关支持基于容器和注册中心服务中的元数据（容器服务中的POD标签或者注册中心服务的元数据）对服务进行分组，并定义为不同版本，即为服务版本概念

创建服务

云原生API网关支持创建容器服务、注册中心服务和固定地址服务，本章节介绍每种服务创建操作流程。

创建容器服务

前置条件

已开通云原生API网关实例和同VPC下的云容器引擎实例，已添加云容器引擎作为云原生API网关的服务来源

服务已部署到云容器引擎并配置相关Service暴露服务

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。
4. 单击左侧导航栏服务 > 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择容器服务来源，选择要添加的服务所在的命名空间。
7. 在服务列表中选择要添加的服务，根据需求做不同的配置。

容器服务配置说明

配置	说明
命名空间	后端服务部署在云容器引擎中的命名空间
servicePort	对应云容器引擎Service中定义的服务端口，每个端口可以定义为一个独立的云原生API网关服务
请求协议	后端服务的协议，支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
mTLS	是否打开后端服务双向TLS认证（后端服务对网关证书认证）
证书文件	后端服务开启双向TLS认证后，云原生API网关提供的证书文件
私钥文件	后端服务开启双向TLS认证后，云原生API网关提供的私钥文件

创建注册中心服务

前置条件

已开通云原生API网关实例和同VPC下的注册配置中心实例（Nacos引擎），已添加注册配置中心实例作为云原生API网关的服务来源

后端服务已部署并注册到注册配置中心

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。
4. 单击左侧导航栏服务 > 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择容器服务来源，选择要添加的服务所在的命名空间。

7. 在服务列表中选择要添加的服务，根据需求做不同的配置。

注册中心服务配置说明

配置	说明
命名空间	后端服务注册到注册配置中心的命名空间
请求协议	后端服务的协议，支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
mTLS	是否打开后端服务双向TLS认证（后端服务对网关证书认证）
证书文件	后端服务开启双向TLS认证后，云原生API网关提供的证书文件
私钥文件	后端服务开启双向TLS认证后，云原生API网关提供的私钥文件

创建固定地址服务

前置条件

已开通云原生API网关实例

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择固定地址服务。
7. 根据需求做不同的配置。

固定地址服务的配置参数

配置	说明
服务名称	用于唯一标识一个后端服务
服务地址	1.支持配置服务节点的地址、端口、权重和优先级。2.优先访问高优先级节点，只有在高优先级的节点不可用或者尝试过，才会访问一个低优先级的节点；3.只有在优先级相同时，权重才会生效；4.优先级默认值为0，可以取负数表示备份节点，表示只有其他节点均不可用时，才会启用备份节点；5.此处节点权重采用比例形式，表示不同节点之间的流量比例
请求协议	后端服务的协议，支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
mTLS	是否打开后端服务双向TLS认证（后端服务对网关证书认证）
证书文件	后端服务开启双向TLS认证后，云原生API网关提供的证书文件

配置	说明
私钥文件	后端服务开启双向TLS认证后，云原生API网关提供的私钥文件

创建DNS域名服务

前置条件

已开通云原生API网关实例

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。
4. 单击左侧导航栏服务 > 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择DNS域名服务。
7. 填写域名、端口等配置。

DNS域名服务的配置参数

配置	说明
服务名称	用于唯一标识一个后端服务
服务地址	1.支持配置服务节点的地址、端口、权重和优先级。2.优先访问高优先级节点，只有在高优先级的节点不可用或者尝试过，才会访问一个低优先级的节点；3.只有在优先级相同时，权重才会生效；4.优先级默认值为0，可以取负数表示备份节点，表示只有其他节点均不可用时，才会启用备份节点；5.此处节点权重采用比例形式，表示不同节点之间的流量比例
请求协议	后端服务的协议，支持HTTP、HTTPS、GRPC、GRPCS、DUBBO，默认为HTTP
mTLS	是否打开后端服务双向TLS认证（后端服务对网关证书认证）
证书文件	后端服务开启双向TLS认证后，云原生API网关提供的证书文件
私钥文件	后端服务开启双向TLS认证后，云原生API网关提供的私钥文件

创建LLM服务

前置条件

已开通云原生API网关实例

操作流程

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。

4. 单击左侧导航栏服务 > 服务标签页。
5. 单击左上角按钮 创建服务。
6. 在弹出的页面中选择LLM服务。
7. 填写服务名称、大模型供应商、API-KEY等配置。

LLM服务的配置参数

配置	说明
服务名称	用于唯一标识一个后端服务
大模型供应商	支持的大模型供应商有：DeepSeek、OpenAI兼容(OpenAI Compatible)、百炼、息壤、火山方舟、千帆等 需特别说明，本AI服务所集成的大模型能力并非由云原生API网关直接提供。您在接入使用前，需自行确认该服务是否匹配您的业务需求并评估其可靠性，同时应确保严格遵循国家法律法规及平台产品协议的相关约定。若因未合规使用引发任何纠纷或不良后果，我方不承担相应责任。
服务地址(base_url)	选定大模型供应商后会自动载入标准接口，其中OpenAI兼容的供应商需要自行填写服务地址
API-KEY	该API-KEY用于在云原生API网关和大模型服务之间进行身份认证。API-KEY的获取请咨询对应服务供应商。

管理服务

概述

云原生API网关支持将请求转发的预定义的目标服务，本章节介绍云原生网关中的服务生命周期管理。

服务详情

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。
4. 单击左侧导航栏服务 > 服务标签页。
5. 点击目标服务的名称，进入服务详情页。
   a. 固定地址服务类型等服务详情页包括，服务基本信息、服务地址、服务策略、服务健康检查等信息。
   b. 容器服务来源类型、MSE注册配置中心服务来源类型的服务，详情页包含服务的版本管理，可新增、删除服务版本。

编辑服务

云原生API网关支持对已添加的服务进行编辑，本章节介绍服务编辑功能。

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 > 进入实例概览。
4. 单击左侧导航栏服务 > 服务标签页。
5. 点击目标服务的编辑操作，在编辑配置页按需修改。

当前允许编辑的服务选项包括：

- 固定地址服务节点的地址、端口、权重、优先级
- 请求协议
- mTLS配置
- DNS域服务的域名、端口
- 容器服务安装、卸载Ingress Controller

删除服务

前提条件

没有路由在引用要删除的服务（先删除引用当前服务的路由）

操作步骤

1. 进入云原生API网关控制台。
2. 在顶部菜单栏选择资源池。
3. 单击左侧导航栏实例 **>** 进入实例概览。
4. 单击左侧导航栏服务 **>** 服务标签页。
5. 点击目标服务的删除操作，点击确认。

配置服务策略

概述

云原生API网关支持丰富的服务策略配置，可以实现后端服务级别的策略配置，包括负载均衡、超时、重试等配置。

操作步骤

1. 点击服务列表项操作栏的策略配置按钮或者服务名称进入服务详情页面。
2. 点击策略的编辑按钮，进入配置状态。
3. 填写配置项参数，点击保存完成修改。

策略配置项和支持的服务类型

配置项	说明	支持的服务类型
负载均衡策略	后端多实例的负载均衡策略，当前支持： 1. 轮询（Round Robin） 2. 一致性哈希（CHash） 3. 指数加权移动平均法（EWMA） 4. 最小连接数（Least Conn）	容器/注册中心/固定地址/DNS域名服务
超时策略	当前支持配置连接超时（默认10s）、发送超时（默认60s）、接收超时（默认60s）	容器/注册中心/固定地址/DNS域名服务
重试策略	1. 重试次数，默认为0，表示不重试 2. 重试超时时间：原请求和重试的总耗时超过该时间则不再继续重试	容器/注册中心/固定地址/DNS域名服务

配置服务健康检查

概述

云原生API网关对后端服务的健康检查分为主动健康检查和被动健康检查。单节点上游服务默认关闭健康检查，可通过单节点健康检查开关强制启用。

主动健康检查

主动健康检查通过预设的探针，主动探测上游节点的存活性，目前支持 HTTP、HTTPS、TCP 三种探针类型。当发往某个健康节点的若干个连续探测请求都失败时，则该节点将被标记为不健康，不健康的节点将会被网关的负载均衡器忽略，无法收到请求；若发往某个不健康的节点的连续若干个探测请求都成功，则该节点将被重新标记为健康，进而可以被代理。

主动健康检查支持的配置项

配置项	说明
探测类型	当前支持TCP、HTTP、HTTPS探测
超时时间	探测请求超时时间
并行数量	并发主动探测的最大数量
端口	探测的服务端口
请求路径	探测请求的路径，仅对HTTP和HTTPS探测有效
健康状态定义	对于不健康节点的探测配置，用于判断节点是否恢复健康。 1. 探测时间间隔（秒） 2. 成功次数：主动探测成功达到次次数时认为节点是健康的 3. 状态码：对于HTTP和HTTPS探测，定义了哪些状态码是健康的，如2XX，3XX
不健康状态定义	对于健康节点的探测配置，用于判断节点是否监控。 1. 探测时间间隔（秒） 2. 超时次数：超时次数大于或者等于该配置时认为节点不健康 3. 状态码：定义了哪些状态码是异常的，如4XX，5XX 4. HTTP失败次数：HTTP失败次数大于等于该值时认为节点不健康 5. TCP失败次数：TCP失败大于等于该值时认为节点不健康

被动健康检查

被动健康检查是指，通过网关接收到的上游节点的响应状态来判断对应的上游节点是否健康。相对于主动健康检查，被动健康检查的方式无需发起额外的探测请求，缺点是无法提前感知节点状态，需要有一定量的失败请求才能触发故障节点的剔除。若发向某个健康节点的若干个连续请求都被判定为失败，则该节点将被标记为不健康。

被动健康检查的配置说明

配置项	说明
类型	当前支持TCP、HTTP、HTTPS三种类型
健康状态定义	• 状态码：成功对应的状态码，主要是2XX、3XX等。 • 成功次数：成功的次数超过该值则认为节点健康
不健康状态定义	• 超时次数：超时次数超过该值则认为节点不健康。 • TCP失败次数：TCP失败超过该值则认为节点不健康。 • HTTP失败次数：HTTP失败超过该值则认为节点不健康。 • 状态码：失败对应的状态码，包括4XX、5XX等

API管理

API概述

REST API

云原生 API 网关提供的 REST API 支持 API 设计、开发、测试、发布、下线的全生命周期管理，通过标准的 HTTP 方法对资源进行操作，适用于 API First、API 精细化管控等场景。

HTTP API

云原生API网关支持创建HTTP API。HTTP API是基于HTTP协议的接口，以路由为中心。适用于K8s Ingress、微服务架构等场景，实现服务的对外快速暴露。

WebSocket API

云原生API网关支持创建WebSocket API。WebSocket API主要适用于即时交互的业务场景。

HTTP API

HTTP API概述

HTTP API

HTTP API是基于HTTP协议的接口，以路由为中心。适用于微服务架构等场景，支持单服务、多服务、Mock、重定向等多种使用场景。

若业务系统无需细粒度的API管理，可通过设置路由规则的方式，指定特定请求由哪个后端服务承接。相较于REST API接口，路由路径的粒度通常更粗，例如可以通过前缀匹配的方式配置路由（/user/*），这种特性支持快速设定访问路径，助力系统间高效协作，具有较高的灵活性，业务系统间的调用逻辑也更简单。

管理HTTP API

创建HTTP API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "创建API"，然后单击HTTP API卡片中的 "创建"。
3. 在 "创建HTTP API" 面板中配置相关参数，单击 "确定"。

配置项	说明
API名称	定义创建的API名称，API名称需要保证唯一。

配置项	说明
描述	填写API相关的描述

编辑HTTP API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API右侧操作栏的 "编辑"，目前仅能编辑描述信息。

删除HTTP API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API右侧操作栏的 "删除"。

路由

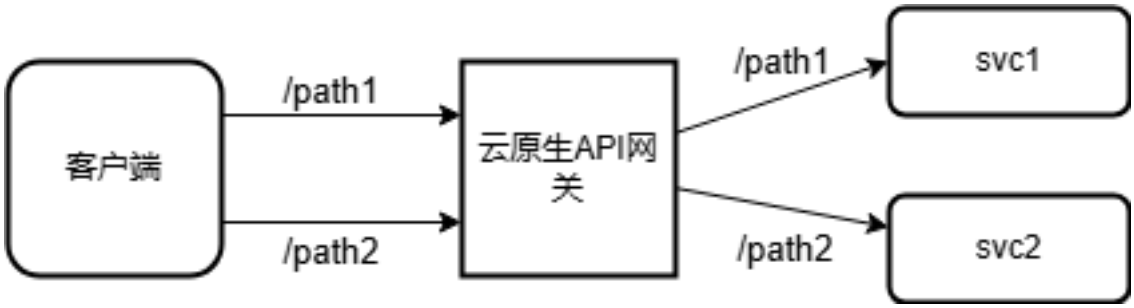
路由概述

概述

云原生API网关基于用户配置的host，path，query，header，cookie等信息配置特定的转发规则，当前云原生API网关支持多种路由方式，包括单服务路由、多服务路由、标签路由、mock和重定向路由等。

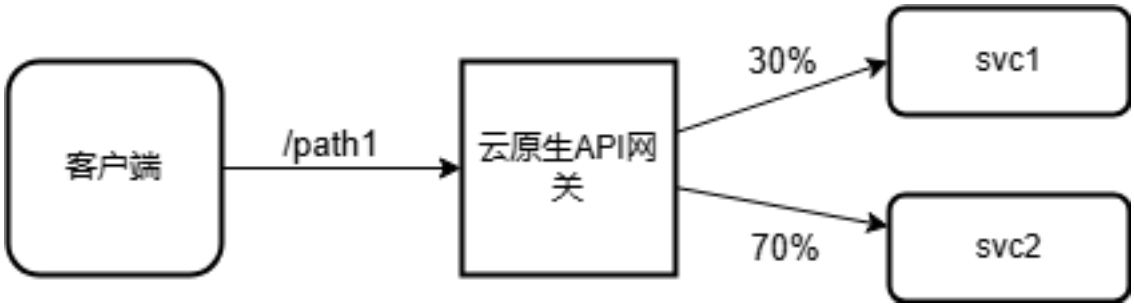
单服务路由

单服务路由根据用户配置的路由规则将请求转发到单个后端服务，例如请求匹配/path1转发到svc1，匹配/path2转发到svc2。



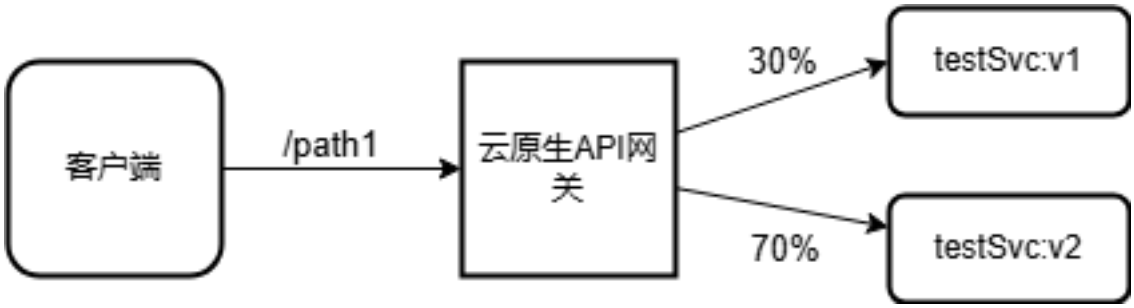
多服务路由

多服务路由模式下支持将请求转发到多个不同的后端服务，每个后端服务配置不同的访问权重（总权重之和为100%）；该模式可用于灰度发布等场景；如对于到/path的请求，30%转发到svc1,70%转发到svc2。



标签路由

针对同一个服务，可以根据nacos或者k8s中记录的服务标签信息将服务分组；标签路由模式下，支持将请求在同一个服务的不同标签分组之间分流；例如，根据服务标签信息将testSvc打上v1和v2两个标签，可以在匹配路由时30%的流量转发到v1版本，70%的流量转发到v2版本：

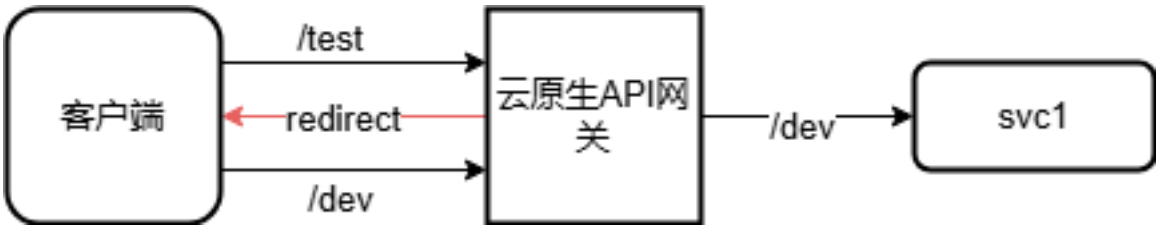


Mock路由

该模式主要用于测试场景，可以配置路由直接返回mock内容，包括HTTP状态码，返回内容等信息，例如针对/path的请求返回HTTP 200，内容为"Hello!"：

重定向路由

返回3xx状态码以及重定向地址，引导调用方访问重定向的地址，如下：



Dubbo代理路由

通过内部dubbo服务代理插件请求dubbo服务，并返回结果。

本功能目前对后端dubbo服务有如下限制：

1. 本功能使用hessian2作为反序列化协议，请确保后端dubbo服务使用该协议作为默认的数据序列化协议。
2. 本功能对dubbo服务的数据返回有相关要求，来确保能将数据结果正确映射成http响应，示例代码如下所示：
   a. 使用Map<String, Object>作为结果返回体；
   b. 其中固定key值body映射为http响应结果；
   c. 固定key值status映射为http响应状态码；
   d. 自定义key值和value值映射为http响应体header。

```
public Map<String, Object> tengineDubbo(Map<String, Object> context) {
  for (Map.Entry<String, Object> entry : context.entrySet()) {
    System.out.println("Key = " + entry.getKey() + ", Value = " + entry.getValue());
  }

  Map<String, Object> ret = new HashMap<String, Object>();
  ret.put("body", "dubbo success\n");
  ret.put("status", "200");
  ret.put("test", "123");

  return ret;
}
```

路由匹配规则

概述

云原生网关支持多种路由匹配规则。彼此之间可以相互组合，路由将会根据所有条件进行“与”的关系匹配。

基于域名的匹配

支持根据请求的域名进行匹配，支持绑定多个域名。也支持类似*.ctyun.com的泛域名。

基于路径的匹配

路径匹配支持精确匹配和前缀匹配模式，云原生网关会优先尝试精确匹配，若无法命中精确匹配，再尝试前缀匹配。

精确匹配

完整匹配给定的路径，如/foo/bar匹配请求路径为/foo/bar的请求。

前缀匹配

末尾使用'*'代表使用前缀匹配，如/foo/bar/*匹配/foo/bar、/foo/bar/baz、/foo/bar/a/b/c等请求。

基于请求方法的匹配

支持根据HTTP请求方法的匹配，例如GET、POST、PUT、DELETE等。可绑定多种请求方法。

基于请求头的匹配

支持根据HTTP请求头进行匹配。多个请求头之间是“与”的匹配关系。

基于请求参数的匹配

支持URL参数进行匹配。多个参数之间是“与”的匹配关系。

基于Cookie的匹配

支持Cookie进行匹配。多个Cookie之间是“与”的匹配关系。

基于参数规整化的匹配

支持从Header、Query或者Cookie中获取参数，可以进一步对参数求哈希值（Java String hashCode方法）、取模等；对于运算的结果支持匹配一组枚举值或者范围；该种匹配方式可用于实现多活容灾路由等场景。

创建路由

概述

输入路由名称，匹配域名、路径、方法、header、query等参数，目标转发地址为服务列表里面配置的地址；路径匹配支持精确匹配和前缀匹配模式，精确匹配如/foo/bar匹配请求路径为/foo/bar的请求，前缀匹配/foo/bar/能够匹配/foo/bar、/foo/bar/baz、/foo/bar/a/b/c等请求。

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 选择已创建的HTTP API，单击API名称进入API详情页。
3. 单击 "创建路由"。
4. 在 "创建路由" 弹出框填写路由相关配置，并单击 "保存" 或 "保存并发布" 按钮，如您单击的是 "保存" 按钮，则需要在 "路由列表" 页，单击操作列 "发布" 按钮发布路由；

路由配置的规则之间是“与”的关系，必须全部满足才算匹配，路由配置项说明如下：

参数	说明
路由名称	路由名称，用于标识一条路由规则。API内需要保证唯一。
描述	路由描述。
域名	用于和请求中的域名进行匹配，不填则任何请求都可以匹配；可选项从域名管理中添加的域名选择。
路径	支持两种匹配方式： 等于：精确匹配。 前缀是：前缀匹配，如配置 /test，则能够匹配所有路径以 /test开头的请求。
方法	匹配请求中的HTTP方法。
优先级	当多个路由同时匹配一个请求时，路径匹配深度较大的路由优先；路径匹配相同的情况下，路由优先级高（数字大）的优先匹配。
请求头（header）	匹配请求中的HTTP header。
请求参数（Query）	匹配请求中的HTTP query参数。
Cookie	通过Cookie进行路由匹配，多个参数之间是“与”的关系。
是否启用参数规整化匹配	启用后支持对参数进行取模，并根据取模结果进行精确或者范围匹配。
参数类型	启用参数规整化匹配选择，支持 Header、Query、Cookie。

参数	说明
是否Hash	启用参数规整化匹配选择，是否对参数进行哈希处理后再取模，哈希函数为Java String hashCode。
取模数值	启用参数规整化匹配选择，自定义填写取模数值。
标记类型	支持精确和匹配精确。精确匹配，可用英文逗号分隔多个精确值。范围：最大值和最小值均是闭区间，[min,max]。
所属实例	路由规则所属的实例。
场景	当前支持单服务、多服务、标签路由、mock路由、重定向和dubbo代理。
后端服务	根据选择的场景选择请求需要转发的后端服务。

管理路由

查看路由详情

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击 "目标路由名称"，可查看路由的 "基本信息" 、"匹配规则" 和 "后端服务" 。

变更路由

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击路由列表的右侧操作栏下的 "编辑" 按钮，可以编辑目标路由。
4. 在编辑路由的弹出框中，您可以修改路由配置，然后点击 "保存" 或 "保存并发布"。

发布路由

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击路由列表的右侧操作栏下的 "发布" 按钮，可以发布目标路由。
4. 在发布操作的确认框单击 "确认"，路由列表的状态栏显示 "已发布" ，表示发布成功。

下线路由

注意

下线后，该路由在实例中将无法访问。请谨慎执行。

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击路由列表的右侧操作栏下的 "下线" 按钮。
4. 在下线操作的确认框单击 "确认"，路由列表的状态栏显示 "未发布" ，表示下线成功。

调试路由

注意

在线调试超时配置默认为30s，因此不适用于调试有超长等待机制的服务场景！

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击路由列表的右侧操作栏下的 "更多" >"调试" 按钮。未发布的路由不能进行调试操作。
4. 在"路由调试" 面板中，配置相关参数，然后单击 "发送请求" 进行调试。

删除路由

注意

- 若当前路由已发布，需要将路由下线后再进行删除。
- 删除路由后不可恢复，请谨慎执行此操作。

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "目标API"，进入API详情，可以查看已添加的路由列表。
3. 单击路由列表的右侧操作栏下的 "更多" >"删除" 按钮。
4. 在对话框中单击 "确定" 。

路由策略配置

概述

在路由详情页的 "策略配置" 标签页中可以配置路由策略，当前支持限流、重写、Header配置、跨域、Query参数设置、Cookie重写、外部认证授权、熔断、黑白名单、防重放和Fallback服务策略配置。

操作步骤

1. 进入目标路由的 "策略配置" 标签页。
2. 单击需要变更的策略，在右侧配置页中修改相关配置后，单击 "保存"。

限流

当前实现为单机限流，基于时间窗口实现，可以配置时间窗口大小（秒）以及在一个时间窗口内限制的请求数。

配置	说明
时间窗口	进行限流统计的时间窗口
限制请求	时间窗口内允许的最大请求次数，超出的请求将会被拒绝。

重写

重写策略可以实现请求向上游转发请求时重写path和host。

配置参数说明如下：

配置	说明
重写路径匹配类型	支持精确匹配、前缀匹配和正则匹配，只有路径匹配的请求才会对路径进行重写。
重写路径	重写的目标路径。
重写主机域名	重写的目标主机域名。

Header配置

Header配置支持对请求和响应的头部做修改。

配置参数说明：

配置	说明
开启状态	开启时策略才生效。
Header类型	网关与后端交互时支持对请求和应答的头部做修改。
操作类型	支持新增、修改、删除操作。新增：若header key已存在，则在末尾追加header value；否则新增。修改：若header key不存在，则新增header kv；否则覆盖已有header value值。删除：若header key存在，则删除；否则忽略该header key。
Header Key	头部Key。
Header Value	头部Value。

跨域设置

云原生网关支持路由级别的跨域资源共享（CORS）。

CORS配置说明如下：

配置项	说明
允许访问的来源	作用于Access-Control-Allow-Origin头部，格式如：scheme://host:port，比如:https://foo.ctyun.com:8080，多个值使用','分割，'*'表示所有Origin均允许通过。
允许的方法	作用于Access-Control-Allow-Methods头部，表示允许的访问方法。
允许的请求头部	作用于Access-Control-Allow-Headers头部，允许跨域访问时请求方携带哪些CORS规范以外的Header，多个值使用','分割，'*'来表示所有Header均允许通过。
允许的响应头部	作用于Access-Control-Expose-Headers头部，允许浏览器和js脚本访问的响应头部。
允许携带凭证	作用于Access-Control-Allow-Credentials头部。
预检的过期时间	作用于Access-Control-Max-Age头部。
开启状态	开启时才生效。

ProxyCookie配置

该配置支持对上游响应Set-Cookie头部重写，当前支持对Set-Cookie头部里的Domain和Path进行重写。

配置项说明：

配置项	说明
proxy_cookie_domain匹配规则	匹配上游应答Set-Cookie头部的Domain字段，支持正则匹配。
proxy_cookie_domain替换值	如果匹配，Set-Cookie头部Domain字段将被替换成该配置值。

配置项	说明
proxy_cookie_path匹配规则	匹配上游应答Set-Cookie头部的Path字段，支持正则匹配。
proxy_cookie_path替换值	如果匹配，Set-Cookie头部Path字段将被替换成该配置值。

Query参数设置

该配置支持对请求参数进行修改。

配置项说明：

配置	说明
开启状态	开启时策略才生效。
操作类型	支持新增、修改、删除操作。新增：若请求参数已存在，则在末尾追加；否则新增。修改：若请求参数不存在，则新增该参数；否则覆盖已有参数值。删除：若请求参数存在，则删除；否则忽略该参数。
参数名	请求参数key。
参数值	请求参数Value。

外部认证授权

该配置支持通过第三方外部服务进行身份认证与授权。当身份认证失败时，可以实现自定义错误或者重定向到认证页面的场景。

配置项说明：

配置	说明
开启状态	开启时配置才生效。
服务地址	设置外部认证服务的地址（例如：https://localhost:9188）。
请求方法	客户端向认证服务发送请求的方法。当设置为POST时，会将请求体转发给认证服务。
转发到认证服务的请求头	设置需要由客户端转发给认证服务的请求头。如果没有设置，则只发送如X-Forwarded-XXX的请求头。
转发给上游服务的请求头	认证通过时，由认证服务转发给上游服务的响应头。如果不设置则不转发任何响应头。
转发给客户端的请求头	认证失败时，由认证服务向客户端发送的响应头。如果不设置则不转发任何响应头。
验证ssl证书	当开启时，验证SSL证书，默认开启。
认证服务请求超时时间	认证服务请求超时时间。
长连接超时时间	长连接超时时间。

熔断设置

该配置支持在触发上游服务不健康状态时进行熔断，从而保护上游业务服务。

配置项说明：

配置	说明
开启状态	开启时配置才生效。
上游服务健康状态码	上游服务处于健康状态时的HTTP状态码。
上游服务连续正常请求次数	上游服务触发健康状态的连续正常请求次数。
上游服务不健康状态码	上游服务处于不健康状态时的HTTP状态码。
触发异常请求次数	上游服务在一定时间内触发不健康状态的异常请求次数。
熔断最大持续时间	上游服务熔断的最大持续时间，以秒为单位。
不健康返回错误码	当上游服务处于不健康状态时返回的HTTP错误码。
不健康返回响应体信息	当上游服务处于不健康状态时返回的HTTP响应体信息。
不健康返回响应头信息	当上游服务处于不健康状态时返回的HTTP响应头信息。该字段仅在配置了不健康返回响应体信息时才生效。

黑白名单

云原生网关支持通过配置IP黑名单和白名单的方式限制客户端访问网关；黑白名单不能同时开启，同时只有一种能生效。

云原生网关默认读取请求中的Remote_addr字段值作为客户端IP（即网络层IP）；如果您的客户端访问出口存在七层代理，此时Remote_addr字段值为出口代理地址，可通过开启从xff头部获取ip配置选项，从X-Forwarded-For字段中获取客户端真实IP。

配置	说明
是否从xff头部获取IP	是否从X-Forwarded-For字段中获取客户端真实IP。
黑名单	黑名单IP配置。
白名单	白名单IP配置。

防重放

防止攻击者重复发送已截获的合法请求，避免重复操作或数据异常。

开启后，请求头必须包含x-ca-timestamp和x-ca-nonce参数。

配置	说明
时间窗口	时间窗口内不可重复请求，请求时间超过时间窗口为无效请求。

Fallback服务

云原生API网关支持对HTTP类型的API下路由在单服务场景下配置Fallback服务，当后端目标服务异常不可用时，网关会将流量转发到Fallback服务中。

配置	说明
服务	当请求的目标服务不可用时，请求需要转发的fallback服务。

REST API

REST API概述

创建REST API并添加接口

云原生 API 网关提供的 REST API 支持 API 设计、开发、测试、发布、下线的全生命周期管理，通过标准的 HTTP 方法对资源进行操作，适用于 API First、API 精细化管控等场景。

添加策略

添加API接口后，您需要通过控制台为API接口添加策略，提高API的安全性、性能和可维护性。

配置消费者认证策略

REST API支持为路由配置认证。通过JWT、HMAC、KEY、BASIC四种认证方式验证调用者的身份，通过精细化API权限管控，确保敏感数据隔离与合规调用，有效防范资源的未授权访问。

管理REST API

云原生API网关提供了多种REST API管理操作，包括发布、下线、导入和导出API等。实现高效管理 API，提升开发流程的便利性。

接口管理

云原生API网关提供了多种接口管理操作，包括接口参数定义、接口调试、配置策略和插件、指定后端服务、和配置消费者认证等。

创建REST API并添加接口

通过控制台创建REST API

1. 登录云原生API网关控制台。先在顶部菜单栏选择 "地域"，然后在左侧导航栏选择 "API"。
2. 单击 "创建API"。
3. 单击 "REST API" 卡片中的 "创建"，在 "新增API(REST)" 弹出框中配置相关参数，单击 "确定"。

配置项	说明
API名称	自定义创建的API名称。 注意 API名称需要保证唯一。
API协议	API允许接受的HTTP协议。
Base Path	API的基本路径，访问具体接口时，完整路径为 http(s)://{##}/{BasePath}/{##Path}。
版本管理	是否启用API版本管理能力，不同版本的API拥有相同的API名称，但参数定义、接口配置可以不同，编辑当前API版本的参数定义、接口配置和策略，不会影响其他版本。不同版本的API可以视为独立的API，访问时需要指定版本标识符。 开启版本管理功能后，需要配置使用方式。

配置项	说明
使用方式	支持Path、Query、Header三种方式。 • 使用Path时，需要完整访问路径为：/{##}/{BasePath}/###/####。 • 使用Query时，完整访问路径为：/{##}/{BasePath}/####，请求参数中需要配置参数项添加Query为版本号。 • 使用Header时，完整访问路径为：/{##}/{BasePath}/####，请求头中需要配置参数项添加Header为版本号。
描述	填写API的相关描述

通过导入OpenAPI文件创建REST API

1. 首先按照上文中通过控制台创建REST API步骤创建一个REST API。
2. 单击刚才新建成功的 "API名称"，进入 API详情页面，单击右上角的 "更多操作"，单击 "导入"。
3. 在弹出框中单击 "从文件导入"，选择待导入的OpenAPI文件，然后单击 "预检并创建"。
4. 选择 "合并逻辑"，各选项说明如下：

选项	说明
智能合并	将在已存在的API基础之上，创建新增的接口，更新重复的接口，但不删除仅在原API中存在的接口。
仅导入新增接口	将在已存在的API基础之上，仅创建新增的接口，对重复或原有的接口不做处理。
覆盖当前API	基于当前导入的文件重新创建API，完全覆盖已有API。

5. 如果预检结果为失败，需要您修改配置文件并重复上述步骤。如果预检结果为成功，可以单击 "创建API" 根据选择的合并逻辑进行API导入 。

添加接口

1. 在REST API中，单击 "添加接口" 。
2. 在 "创建接口" 弹出框中配置相关参数，单击 "保存" 。

配置项	说明
接口名称	自定义创建的接口名称，在API下需要唯一。
接口Path	接口的具体路径。
方法	接口的请求方法。接口的路径+接口的方法，需要在API下唯一。
描述	接口的描述信息。
请求定义	支持定义 "Header" 、"Query"、"Parameter Path" 参数以及 "Body" 参数。 其中Path参数支持在接口Path中按照如下方式进行变量定义： • /books/:bookId 说明 请求定义仅用于文档展示和生成，不对运行时进行校验。

配置项	说明
响应定义	定义不同响应码的数据结构。 说明 响应码定义仅用于生成文档，不对运行时进行校验。
Mock	Mock配置仅在API发布Mock场景下生效。

接口管理

编辑接口

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。在 "API设计" 标签页的 "接口列表"栏，选择需要编辑的接口，单击 "编辑"。
3. 在 "编辑接口" 弹出框中，可修改接口 "基本信息"、"请求定义"、"响应定义" 和 "Mock" 。

删除接口

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。在 "API设计" 标签页的 "接口列表"栏，选择需要删除的接口，单击右侧的 "删除"，然后在删除对话框单击 "确定" 。

注意

删除接口操作执行后不可恢复，请谨慎执行。

调试接口

注意

在线调试超时配置默认为30s，因此不适用于调试有超长等待机制的服务场景！

前提条件

接口所属API已发布到目标实例。

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。
3. 在 "API设计" 标签页的 "接口列表"栏，选择需要调试的接口，单击右侧的 "调试"。
4. 在 "接口调试" 弹出框中，配置相关参数，然后单击 "发送请求" 进行调试。

管理REST API

发布API

前提条件

在发布API之前，确保API中已经定义并创建了接口。

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上方的 "发布API" 。

3. 在发布API弹出框中配置相关参数，然后单击 "发布" 。

配置项	说明
域名	选择域名进行发布，发布后，可以通过域名访问API。
所属实例	选择所创建的云原生API网关实例。不同的业务环境可用不同的实例区分，实现API在多环境上的发布。
使用场景	使用场景包括基础场景和灰度场景两类。 **基础场景** • **Mock**：接口响应将返回接口定义中的Mock配置，若接口未定义Mock配置，则将无法访问该接口。 说明 发布Mock场景时，要求当前API中至少有一个接口开启了Mock配置，否则将发布失败。 • **单服务**：所有流量请求将转发到某一具体的后端服务，这个场景为最常使用的场景。 **灰度场景** • **按比例（多服务）**：流量将按比例分发到对应的后端服务中，常用于切流及灰度发布场景。 说明 要求服务权重之和等于100。
后端服务	关联该网关实例下的后端服务。
发布描述	填写API的发布描述。

添加API版本

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API，单击右上角的 "更多操作" > "添加版本" ，完成相关配置：

配置项	说明
使用方式	支持Path、Query、Header三种方式。 • 使用Path时，需要完整访问路径为：/{##}/{BasePath}/###/####。 • 使用Query时，完整访问路径为：/{##}/{BasePath}/####。请求参数中需要添加Query参数，值为版本号。 • 使用Header时，完整访问路径为：/{##}/{BasePath}/####。请求头中需要配置添加Header参数，值版本号。

3. 在添加完成后，您可以单击页面上方的 "版本" 下拉框选择版本切换。

导入API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。

2. 单击目标API。单击右上角的 "更多操作" > "导入" 。
3. 在弹出框中单击 "从文件导入"，选择待导入的OpenAPI文件，然后单击 "预检并创建"。
4. 选择 "合并逻辑"，各选项说明如下：

选项	说明
智能合并	将在已存在的API基础之上，创建新增的接口，更新重复的接口，但不删除仅在原API中存在的接口。
仅导入新增接口	将在已存在的API基础之上，仅创建新增的接口，对重复或原有的接口不做处理。
覆盖当前API	基于当前导入的文件重新创建API，完全覆盖已有API。

5. 如果预检结果为失败，需要您修改配置文件并重复上述步骤。如果预检结果为成功，可以单击 "创建API" 根据选择的合并逻辑进行API导入 。

导出API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "导出" 。

查看发布历史

说明

发布历史保留最近10次

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API，然后单击 "发布历史" 标签页。
3. 单击 "版本" 和 "实例下拉框" 可以选择不同API版本和实例下的发布历史。
4. 单击目标历史版本 "操作" 列下的 "查看" ，可查看历史版本详情。
5. 单击目标历史版本 "操作" 列下的 "切换至该版本" ，可将当前API的在当前实例上的已发布版本切换到选定版本。

下线API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "下线" 。

删除API

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API。单击右上角的 "更多操作" > "删除" 。

添加策略

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击目标API，您可以在下拉框中选择需要添加策略的实例。

3. 您可以进行API级或接口级进行策略与插件配置：

   a. **API级**：单击 "API策略配置" 标签页，此标签页将针对全部接口进行API级策略与插件配置，然后单击 "启用策略/插件" 。

   b. **接口级**：在 "API设计" 标签页 "接口列表" 栏单击目标接口，单击 "策略与插件配置" ，然后单击 "启用策略/插件" 。

4. 策略配置

策略名称	描述
限流	通过配置限流策略，您可以在指定时间窗口内的配置请求数量，那么在时间窗口内超出阈值的请求将被阻止，以此来确保后端服务始终可用。
重写	通过配置重写策略，网关可以在请求被网关转发至目标服务之前修改请求路径和主机域，确保请求被正确路由到合适的服务或端点。
Header设置	通过配置Header设置，网关可以在请求转发至目标后端服务之前修改原始请求的头信息，或者在后端服务的响应返回给客户端之前修改响应的头信息。
跨域	通过配置跨域策略，可以在服务端启用跨域资源共享（CORS，Cross-Origin Resource Sharing），允许Web应用服务器进行跨域访问控制，实现安全的数据传输。
ProxyCookie设置	该配置支持对上游响应Set-Cookie头部重写，当前支持对Set-Cookie头部里的Domain和Path进行重写。
Query参数设置	您可以设置请求Query参数的新增、修改和删除三种修改方式，并配置具体的参数名和参数值。网关将对原始请求的Query参数进行新增或修改或删除后，转发到后端目标服务中。
外部认证授权	该配置支持通过第三方外部服务进行身份认证与授权。当身份认证失败时，可以实现自定义错误或者重定向到认证页面的场景。
熔断设置	该配置支持在触发上游服务不健康状态时进行熔断，从而保护上游业务服务。
黑白名单	云原生网关支持通过配置IP黑名单和白名单的方式限制客户端访问网关；黑白名单不能同时开启，同时只有一种能生效。
防重放	通过验证请求的唯一性（如时间戳、序列号等）防止攻击者重复发送已截获的合法请求，避免重复操作或数据异常。

配置网关自定义响应

概述

您可以自定义各种http响应码的返回内容，当后端服务返回对应响应码时，网关将按照预设的响应码内容返回给客户端。

操作步骤

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择 "API" ，并在顶部菜单栏选择地域。
3. 进入REST API详情页>网关自定义响应页签。
4. 点击 "新增" ，填写配置信息后，点击 "新增" 。

WebSocket API

创建WebSocket API

概述

WebSocket API基于WebSocket协议，允许客户端和服务器之间进行双向通信，可用于聊天等场景。

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "创建API"，然后单击WebSocket API卡片中的 "创建"。
3. 在 "创建WebSocket API" 面板中配置相关参数，单击 "确定"。

配置项	说明
API名称	定义创建的API名称，API名称需要保证唯一。
描述	填写API相关的描述

路由

WebSocket API下路由介绍

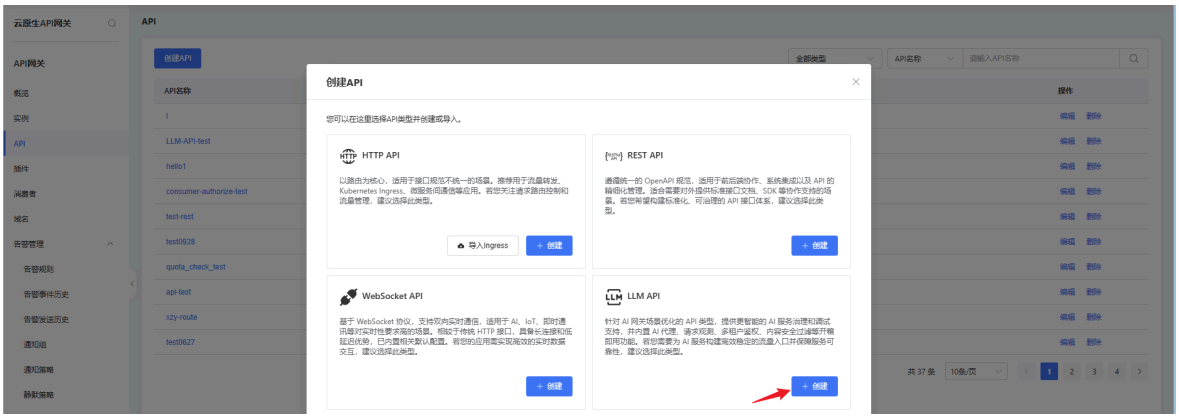
WebSocket API下的路由管理方式和HTTP API下路由管理方式基本一致，可以参考 HTTP 路由创建。

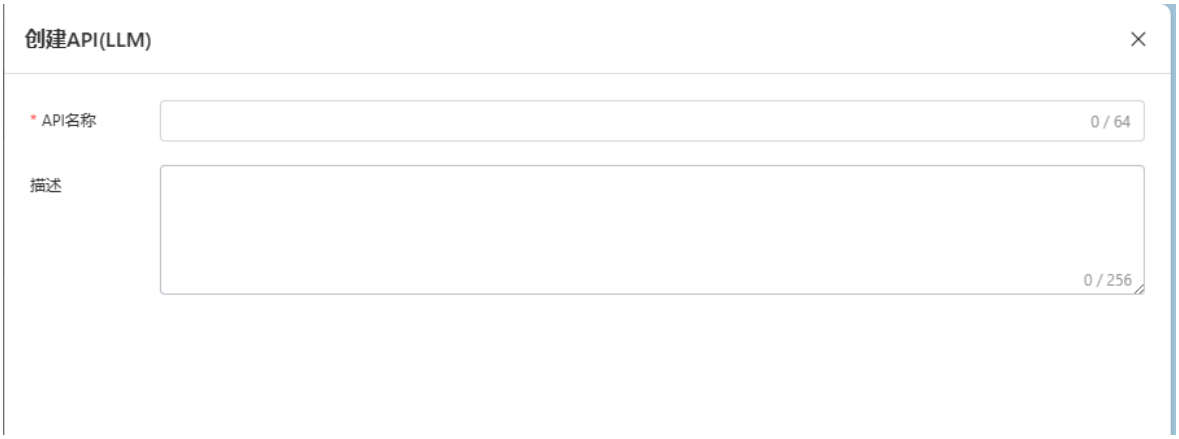
LLM API

创建LLM API

操作步骤

1. 登录云原生API网关控制台。顶部菜单栏选择 "地域"，然后再左侧导航栏选择 "API"。
2. 单击 "创建API"，然后单击LLM API卡片中的 "创建"。
3. 在 "创建LLM API" 面板中配置相关参数，单击 "确定"。





配置项	说明
API名称	定义创建的API名称，API名称需要保证唯一。
描述	填写API相关的描述

管理路由

前提

如果您的网关未绑定 SNAT 服务，则可能无法访问公网。若您已配置VPC 下公网访问方式，可忽略本提示。若您的网关实例无法访问公网，请您前往 创建 NAT 网关。

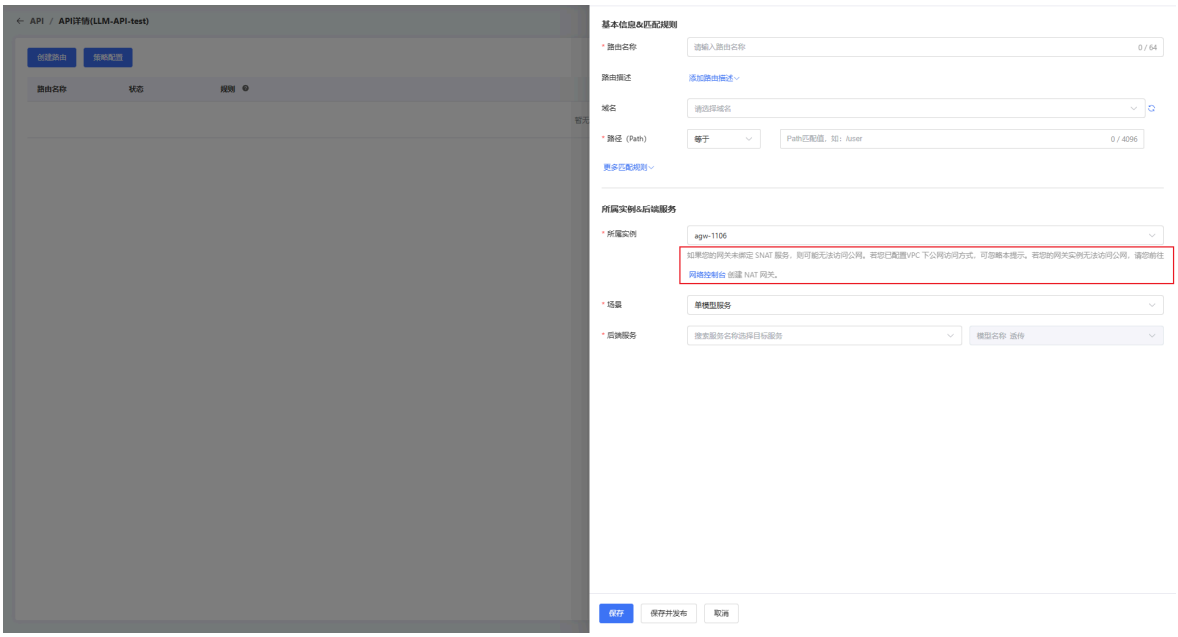
LLM API下路由介绍

路由管理

LLM API下的路由管理方式和HTTP API下路由管理方式基本一致，可以参考 HTTP 路由创建。

区别点

（1）目标实例要关联SNAT服务

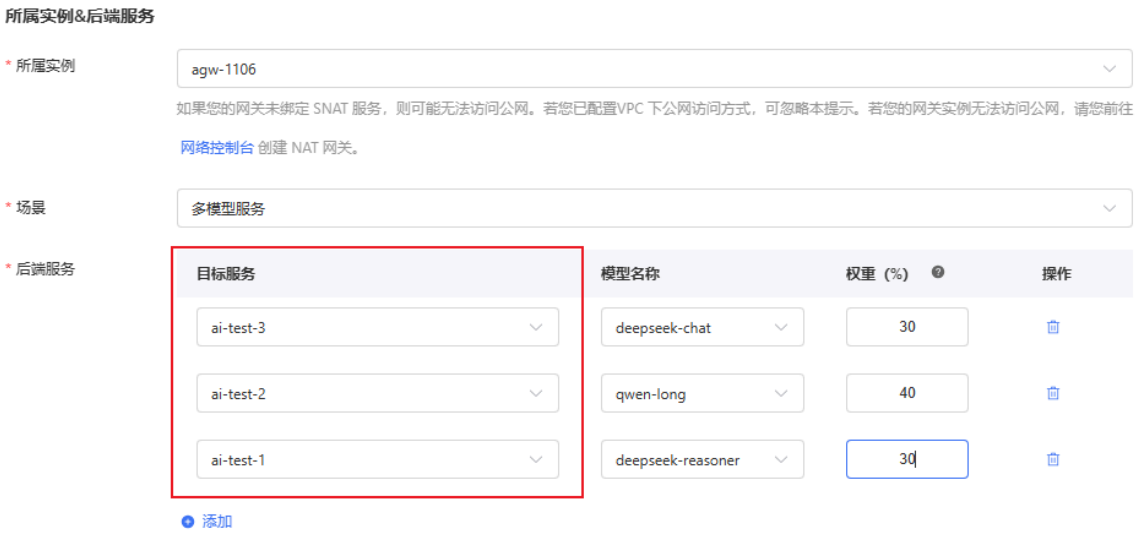


（2）目标场景

- 基础场景：单模型服务，简单理解就是某一模型的单服务，模型类型由服务本身决定
- 灰度场景：多模型服务，简单理解就是不同模型的多个服务，支持配置流量比例

（3）目标服务

- 目标服务配置：按可见列表填入即可



- 模型名称：根据服务对应的供应商来决定，可以配置可调用的模型类型
- 权重：可以为各个服务配置不同的流量比例，所有服务的权重和必须为100%

域名管理

创建域名

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 单击 "添加域名"，在添加域名面板中配置相关参数，然后单击 "确定"。

参数	描述
协议	可选择HTTP或HTTPS协议，不同协议支持的端口为： • HTTP：支持27151端口 • HTTPS：支持27154端口
域名	支持完整域名和泛域名的形式，例如绑定*.ctyun.cn这个域名后，可通过a.ctyun.cn，1.ctyun.cn等来访问。 域名填写规则为：域名包含英文字母、数字和连接符(-)，连接符不能在每段首尾或连续出现，每段长度不得超过63个字符且最后一段为2-6个英文字符
选择协议为HTTPS协议时	

参数	描述
证书	选择HTTPS协议时需上传证书或选择已有证书。 注意 所选择的证书域名需与所填写域名相匹配。如证书域名为*.ctyun.cn，则所填写域名可以是*.ctyun.cn，a.ctyun.cn等。

绑定场景

- 创建HTTP/WebSocket类型的路由时可绑定域名
- 发布REST类型的API时可绑定域名

删除域名

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 在域名列表页面，选择目标域名名称的操作列下单击"删除"，然后在确认删除对话框中输入当前的域名，然后单击"删除"。

说明

删除域名前请先确保没有在线路由或API正在使用该域名。

更换域名协议或证书

当发生证书到期、域名所有者有变更、网站需从HTTP升级到HTTPS等情况时，需要变更域名的证书和协议，来保障网站或平台的安全性与合规性。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 在域名列表页面，选择目标域名名称的操作列下单击"编辑"。

- 更换协议：单击域名右侧的下拉列表选择协议，然后单击"确定"。
- 更换证书：单击证书右侧的下拉列表选择证书，或上传新的证书，然后单击"确定"。

上传证书

当域名为HTTPS协议时，必须选择对应的证书文件。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 在创建或编辑域名页面，选择HTTPS协议时，点击 "上传证书"。

4. 在上传证书面板中配置相关参数，然后单击 "上传 "。

参数	描述
证书名称	标记证书的名称。
证书文件	证书文件，支持pem、cer，crt，key格式。
私钥文件	私钥文件，支持pem、cer，crt，key格式。需与证书文件相匹配。

编辑证书

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 在创建或编辑域名页面，选择HTTPS协议时，在证书列表中，选择目标证书项的操作列下单击"修改"。
4. 在修改证书面板中重新上传证书文件和私钥文件，然后单击 "上传"。

删除证书

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "域名"，并在顶部菜单栏选择地域。
3. 在创建或编辑域名页面，选择HTTPS协议时，在证书列表中，选择目标证书项的操作列下单击"删除"，然后在确认删除对话框中单击"删除"。

> 说明
>
> 删除证书前请先确保没有域名正在使用该证书。

消费者管理

消费者概述

创建消费者

云原生API网关通过消费者机制启用认证能力，实现访问权限控制。每种消费者均可独立设置 JWT、HMAC、KEY 或 BASIC 认证策略，您可根据业务安全需求灵活选择合适的鉴权方式，确保API访问既安全又高效。具体详情，请参见 创建消费者。

启用消费者

消费者只有在启用状态时才能生效。具体详情，请参见 启用消费者。

停用或删除消费者

如您需要停用或删除消费者。具体详情，请参见 停用或删除消费者。

消费者授权管理

每个消费者均可配置独立的身份标识和授权策略，确保仅通过认证的请求才能访问对应资源。云原生API网关支持在路由、接口、API 及实例多个维度配置消费者授权机制，实现细粒度的访问控制。具体详情，请参见 消费者授权管理。

创建消费者

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "消费者" ，并在顶部菜单栏选择地域。
3. 单击 "创建消费者" ，在创建消费者面板中配置相关参数，然后单击 "保存" 。

配置项	描述
消费者名称	自定义消费者名称，要保证唯一性。 命名规则：支持英文字母、数字和下划线，以英文字母或数字开头及结尾，4-32个字符。
启用状态	消费者状态包括已启用和已停用。创建完成后可以手动启停。 注意 消费者只有启用状态时才生效。
描述	对消费者进行描述。

配置项	描述
认证方式	当前消费者支持JWT、HMAC、KEY和BASIC四种认证方式。创建认证时必须选择至少一种认证方式。删除消费者下的认证时也要保证至少存在一种认证方式。对于HMAC认证的AppKey，KEY认证的key，以及BASIC认证的用户名填写时需保证唯一，例如消费者1和消费者2下的KEY认证不可设置相同的key值。 **JWT认证**：JSON Web Token (JWT) 是一种用于在客户端和服务器之间以紧凑且自包含的JSON对象形式安全地传输信息。JWT通过数字签名确保信息的完整性和真实性，常用于在网关中验证用户身份并控制对资源的授权访问，从而实现无状态的身份验证和授权机制。JWT的规范说明可参考：JSON Web Key (JWK) 秘钥类型：可选择对称秘钥和非对称秘钥。 加密算法：对称秘钥加密算法可选择HS256和HS512。非对称秘钥加密算法可选择RS256和ES256。 秘钥：对称秘钥时需填写secret，如果开启base64secret则需按照base64编码格式填写。非对称秘钥时需填写RSA/ES公钥和私钥。 **HMAC认证**：HMAC是一种基于哈希函数的消息认证码算法，用于验证消息的完整性和真实性。客户端使用签名密钥对请求内容生成签名并发送给服务器，服务器通过相同密钥和算法验证签名，确保请求未被篡改且来源可信。 AppKey：AK配置，可由系统生成。 AppSecret：SK配置，可由系统生成。 **KEY认证**：KEY认证是一种基础的认证机制，客户端在调用API时需将凭证（如API Key）以特定方式添加到请求中，网关接收到请求后会校验Key的有效性和权限范围。这种认证方式安全性较低，不建议用于涉及敏感操作的场景。 Bearer前缀：如果勾选，则访问时需携带"Bearer XXXXXXXXXX"的Key值；如果不勾选，则直接携带Key值访问。 Key：自定义Key凭证。 **BASIC认证**：BASIC认证是一种基础的认证机制，客户端在访问时需携带用户名和密码的信息，网关接收到请求后会校验用户名和密码的有效性和权限范围。这种认证方式安全性较低，不建议用于涉及敏感操作的场景。 用户名：BASIC认证用户名。 密码：BASIC认证密码。

启用消费者

操作步骤

1. 登录 云原生API网关控制台。

2. 在左侧导航栏，选择 "消费者" ，并在顶部菜单栏选择地域。
3. 在消费者列表页操作栏中点击 "启用"，在弹出提示框中点击 "确认"。
4. 或进入消费者详情页，在基本信息栏的状态 点击 "启用"，在弹出提示框中点击 "确认"。

消费者授权管理

授权消费者

操作步骤1：在消费者管理侧授权

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "消费者"，并在顶部菜单栏选择地域。
3. 在消费者页面，单击目标消费者进入详情页，选择消费者授权页签>"添加授权"。
4. 在授权面板中，配置相关参数，单击"确定"。

配置项	描述
授权范围	云原生API网关支持四种维度的授权范围： • 实例：若授权消费者给实例，则作用于该实例下的所有路由和接口。 • API：若授权消费者给API，则作用于该API下的所有接口。目前只支持授权给REST类型的API。 • 接口：若授权消费者给接口，则作用于该接口。 • 路由：若授权消费者给路由，则作用于该路由。 作用范围由大到小为：实例>API>接口=路由
有效期	选择此次消费者授权的有效期，如果不选择，则默认永久有效。
授权实例	选择需要授权的实例。
授权API	选择需要授权的API，如果API有多个版本，则通过API名称/版本名称的形式展示。
授权接口	选择需要授权的接口。
授权路由	选择需要授权的路由。

操作步骤2：在资源端侧授权

1. 登录 云原生API网关控制台。
2. 找到不同资源端的消费者管理页面。
3. 点击授权，选择消费者列表>"确定"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于API，在左侧导航栏，选择 "API"，进入"REST API"详情页>消费者认证页签。
- 对于接口，在左侧导航栏，选择 "API"，进入"REST API"详情页，单击接口>消费者认证页签。
- 对于路由，在左侧导航栏，选择 "API"，进入"HTTP/WebSocket API"详情页，单击路由>消费者认证页签。

解除授权

操作步骤1：在消费者管理侧解除授权

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "消费者"，并在顶部菜单栏选择地域。

3. 在消费者页面，单击目标消费者进入详情页>消费者授权页签。
4. 在已授权列表中，支持批量勾选后点击"解除授权"，也支持对单个资源右侧操作栏点击"解除授权"。

操作步骤2：在资源端侧解除授权

1. 登录 云原生API网关控制台。
2. 找到不同资源端的"消费者管理页面"。
3. 已授权消费者列表右侧操作栏中点击"解除授权"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于API，在左侧导航栏，选择 "API"，进入"REST API"详情页>消费者认证页签。
- 对于接口，在左侧导航栏，选择 "API"，进入"REST API"详情页，单击接口>消费者认证页签。
- 对于路由，在左侧导航栏，选择 "API"，进入"HTTP/WebSocket API"详情页，单击路由>消费者认证页签。

资源端开启消费者认证

注意

在资源端开启消费者认证后，需为当前路由/接口绑定消费者授权关系，否则无法访问。

实例侧开启消费者认证

说明

- 在实例上开启消费者认证后，则将对该实例下的所有路由/接口都生效，请谨慎开启。
- 实例上的认证是独立的，例如在实例上开启了JWT认证，在实例下的路由开启HMAC认证，则访问该路由时需要同时携带两种认证方式。如果在实例和路由上开启的是同一种认证，则需要携带相同的消费者认证才能访问。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "实例"，并在顶部菜单栏选择地域。
3. 进入实例概览页>消费者认证页签。
4. 点击编辑配置信息，填写配置信息后，点击"确认"。

配置项	描述
启用状态	开启后，认证鉴权生效，访问其下资源需要携带对应认证信息。
认证方式	当前路由认证消费者时使用的认证方式。目前支持JWT、HMAC、KEY、BASIC的认证方式。
Token配置	访问携带Token相关配置信息。

配置项	描述
JWT Token	JWT Token 配置信息。 隐藏认证信息：是否将认证信息透传到后端服务。 **Header**：设置从哪个header获取token，优先级最高。默认值为'authorization'，使用方式 -H 'authorization: token值'。 **Query**：设置从哪个query string获取token，优先级低于header。默认值为'jwt'，使用方式 '?jwt=token值'。 **Cookie**：设置从哪个cookie获取token，优先级低于query。默认值为'jwt'，使用方式 '--cookie=jwt值'。
HMAC Token	HMAC Token 配置信息。 **隐藏认证信息**：是否将认证信息透传到后端服务。 **加密算法**：支持"hmac-sha1"，"hmac-sha256"，"hmac-sha512"。
KEY Token	KEY Token 配置信息。 **隐藏认证信息**：是否将认证信息透传到后端服务。 **Header**：设置从哪个header获取token，优先级最高。默认值为'apiKey'，使用方式 -H 'apiKey: token值'。 **Query**：设置从哪个query string获取token，优先级低于header。默认值为'apiKey'，使用方式 '?apiKey=token值'。
BASIC Token	BASIC Token 配置信息。 **隐藏认证信息**：是否将认证信息透传到后端服务。

API侧开启消费者认证

说明

- 在REST API上开启消费者认证后，则将对该API下的所有接口都生效。
- API和接口上只能开启同一种认证方式。因此，如API上开启消费者认证，则其下所有接口都会开启相同认证；如存在接口开启了消费者认证，则无法在该API上开启认证，需要关闭API下所有接口的认证后才能开启。
- API需要发布后才能生效。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入REST API详情页>消费者认证页签。
4. 点击编辑配置信息，填写配置信息后，点击"确认"。

接口侧开启消费者认证

说明

- 接口上如未开启消费者认证，则默认和所在API上的消费者认证状态一致。
- 接口可以额外进行消费者授权。例如接口所在API授权了消费者1，接口本身授权了消费者2，则携带消费者1或消费者2的认证信息都能访问该接口。
- 接口需为发布状态才能生效。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入REST API详情页，单击接口>消费者认证页签。
4. 点击编辑配置信息，填写配置信息后，点击"确认"。

路由侧开启消费者认证

说明

路由需为发布状态才能生效。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入HTTP/WebSocket API详情页，单击路由>消费者认证页签。
4. 点击编辑配置信息，填写配置信息后，点击"确认"。

资源端关闭消费者认证

操作步骤

1. 登录 云原生API网关控制台。
2. 找到不同资源端的消费者管理页面。
3. 点击编辑配置信息，启用状态选择"关闭"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于API，在左侧导航栏，选择 "API"，进入"REST API"详情页>消费者认证页签。
- 对于接口，在左侧导航栏，选择 "API"，进入"REST API"详情页，单击接口>消费者认证页签。
- 对于路由，在左侧导航栏，选择 "API"，进入"HTTP/WebSocket API"详情页，单击路由>消费者认证页签。

停用或删除消费者

停用消费者

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "消费者" ，并在顶部菜单栏选择地域。
3. 在消费者列表页操作栏中点击 "停用"，在弹出提示框中点击 "确认"。
4. 或进入消费者详情页，在基本信息栏的状态 点击 "停用"，在弹出提示框中点击 "确认"。

删除消费者

注意

删除消费者前请先停用消费者。

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "消费者"，并在顶部菜单栏选择地域。
3. 在消费者列表页面，单击目标消费者的操作列下的"删除"，然后在确认删除对话框中输入当前的消费者名称，然后单击"删除"。

观测分析

查看网关监控数据

操作步骤

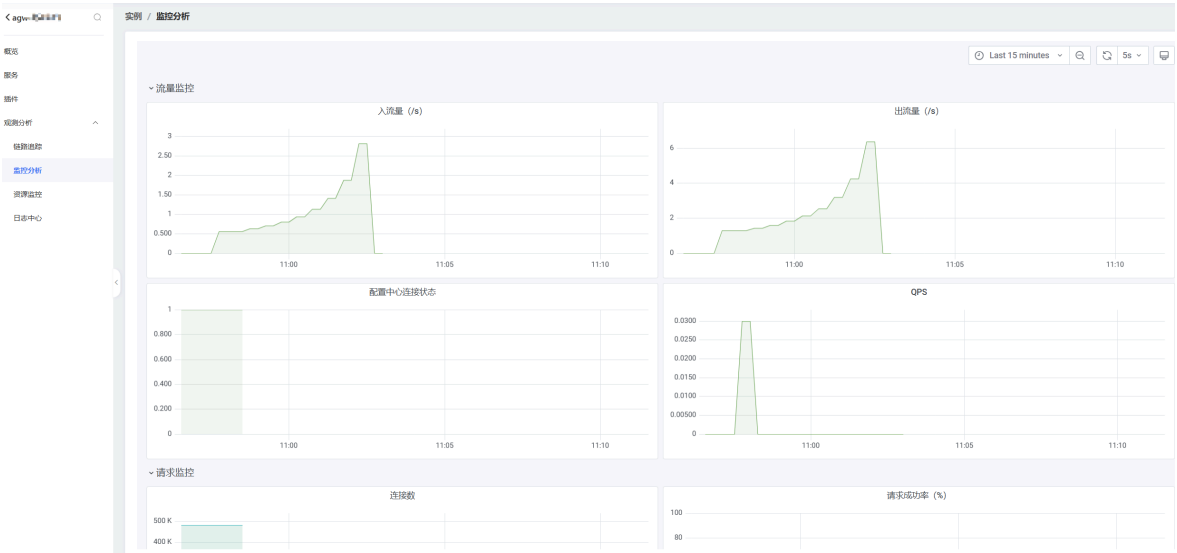
1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "概览"。
3. 在实例监控模块，右上角下拉列表可选择实例。
4. 在实例监控页面，可看到该实例最近时刻的QPS，请求成功率，节点CPU使用率和已使用内存的指标信息。

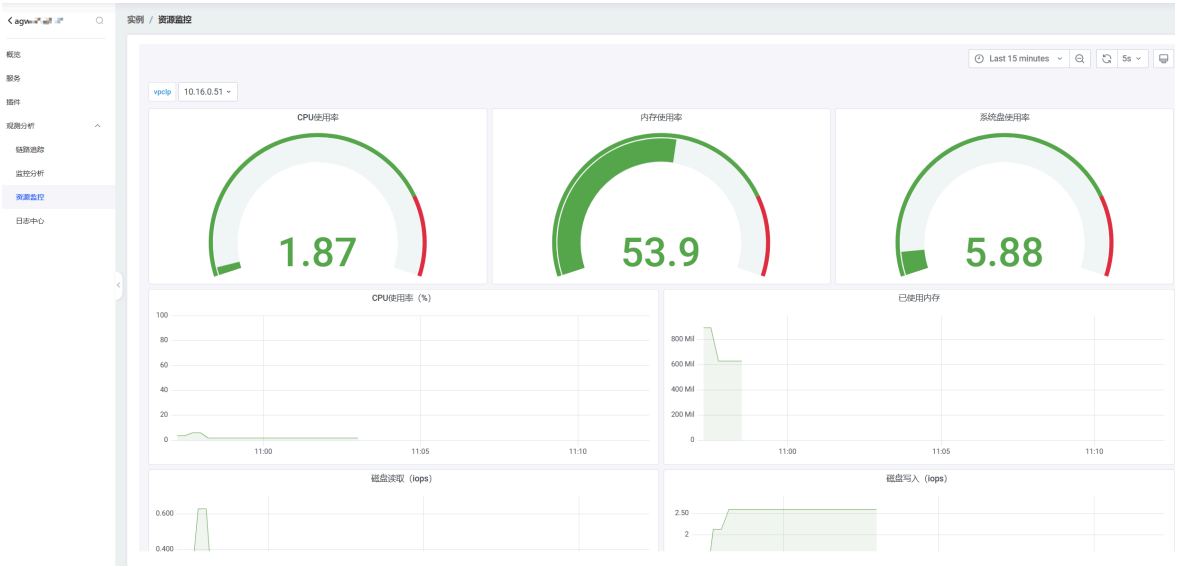
查看网关资源监控数据

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "实例"，并在顶部菜单栏选择地域。

3. 进入实例详情页，在左侧导航栏中，选择 "观测分析"。
4. 选择"监控分析"，可看到该实例最近时刻的流量监控，请求监控和延迟监控等指标信息。
5. 选择"资源监控"，可看到该实例每个节点最近时刻的CPU使用率，内存使用率，磁盘和网络等指标信息。





开启网关日志采集

云原生网关对接天翼云日志服务（LTS）实现了访问日志采集、上报和查询能力。开启日志采集后，您可以通过分析云原生API网关的访问日志了解客户端用户行为，以便排查问题。

操作步骤

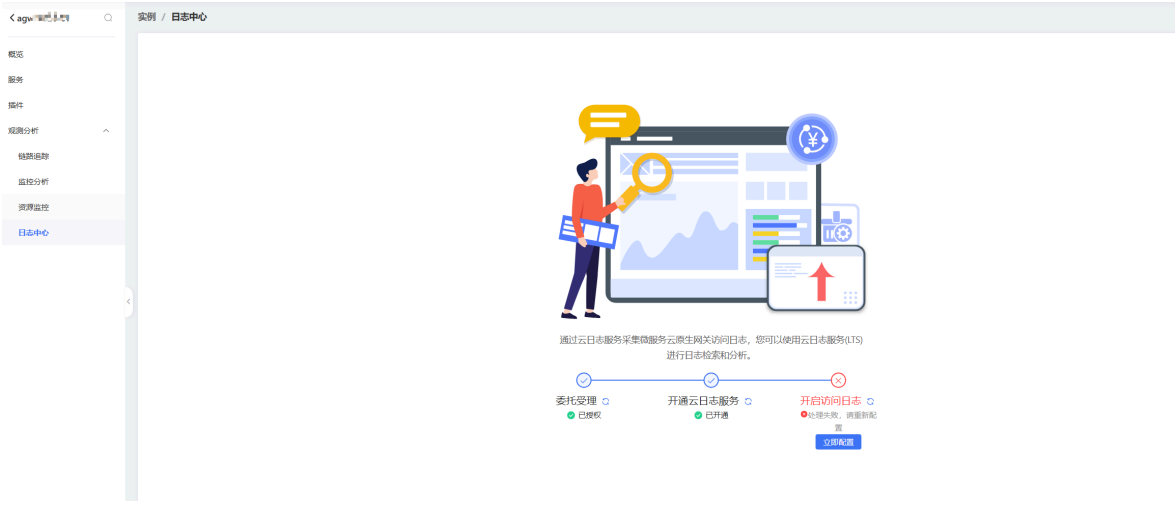
1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "实例"，并在顶部菜单栏选择地域。
3. 进入实例详情页，在左侧导航栏中，选择 "观测分析"。

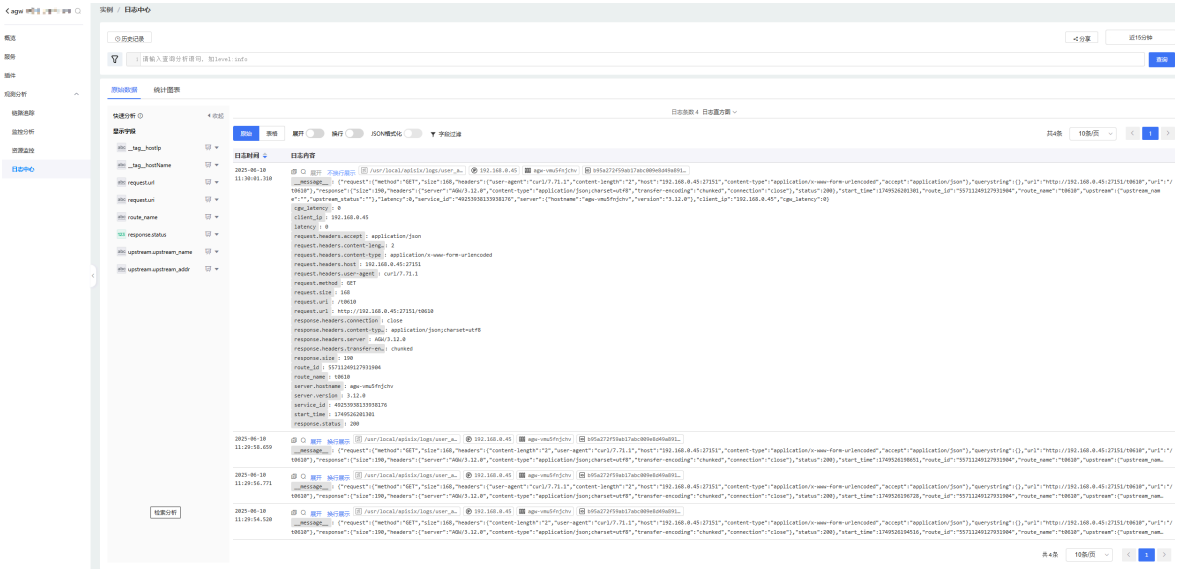
4. 选择 "日志中心"，有三个步骤进行开启：

- 检查是否开启委托授权，如未开启，需要点击 "立即创建"，创建名为CtyunAgwAdminTrust的委托授权。
- 检查是否开通日志服务，如未开通，需要点击 "立即开通"，开通日志服务。
- 检查是否接入日志，如未接入，需要点击 "开启"，接入日志服务。

说明

- 日志和链路追踪的启用需要您授权云原生API网关访问云日志服务和应用性能监控服务的权限，允许使用相关资源。
- 日志和链路追踪服务将根据您的使用量进行收费。





访问日志格式说明

网关访问日志示例


```
{
 "start_time": 1725411921254,
 "request": {
  "size": 335,
  "method": "POST",
  "uri": "/foo/bar",
  "headers": {
   "accept": "text/plain, application/json, application/*+json, */*",
   "content-length": "0",
   "uber-trace-id": "b32bce4ff1f00f7b:899a1fd65c39be02:1c3372663d9eae03:0",
   "connection": "keep-alive",
   "user-agent": "Java/1.8.0_212",
   "host": "foo.ctyun.com",
   "content-type": "application/json"
  },
  "url": "http://foo.ctyun.com:27151/foo/bar",
  "querystring": {}
 },
 "service_id": "a4db25fc03294d5dbeb9e7752381c972",
 "server": {
  "version": "3.2.2",
  "hostname": "agw-vmxxxxxx"
 },
 "agw_latency": 0,
 "latency": 24.00016784668,
 "client_ip": "100.89.x.x",
 "response": {
  "size": 625,
  "headers": {
   "via": "1.1 alb/v3.4.5",
   "content-length": "427",
   "date": "Wed, 04 Sep 2024 01:05:21 GMT",
   "connection": "close",
   "server": "AGW/3.2.2",
   "content-type": "application/json;charset=UTF-8"
  },
  "status": 400
 },
 "upstream": {
  "upstream_addr": "10.121.x.x:80",
  "upstream_status": "400",
  "upstream_latency": 25,
  "upstream_name": "foo-service"
```

},
"route_id":"ddd342a2a3f34405bb8650ad4e",
"route_name":"test-route"
}

日志索引字段说明如下

字段	说明
__tag__hostIp	数据来源主机IP
__tag__hostName	数据来源主机名称
request.url	请求url
request.uri	请求uri
route_name	路由名称
response.status	响应结果状态
upstream.upstream_name	上游服务名称
upstream.upstream_addr	上游服务地址

开启网关链路追踪

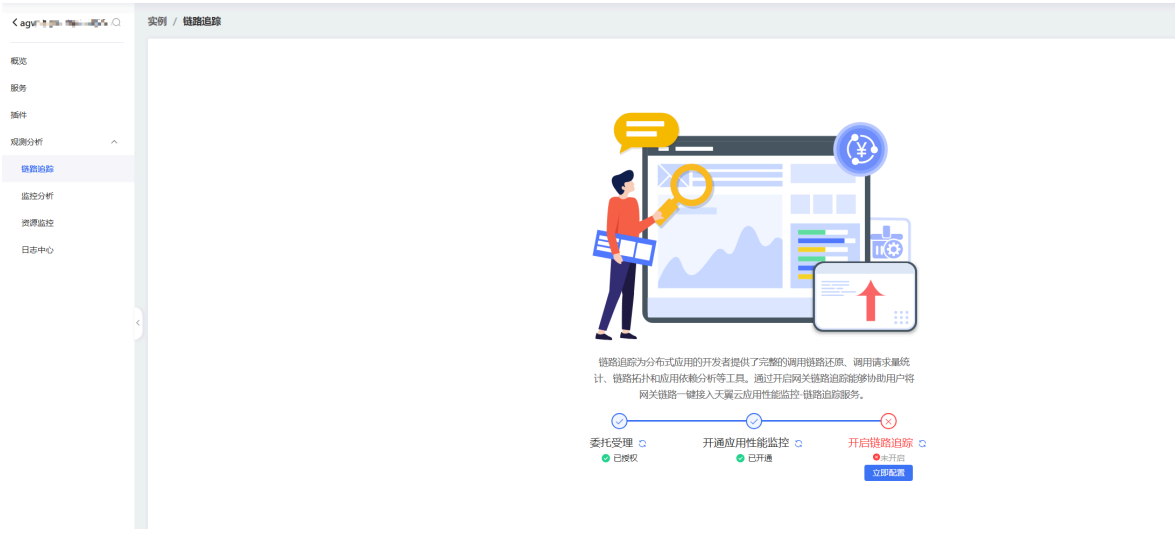
在开启了链路追踪并配置采样率大于0，网关会根据采样率配置上报链路追踪数据，链路追踪基于traceid将调用链上下游串联起来，帮助您分析和诊断分布式应用架构下的性能瓶颈，提高微服务时代下的开发诊断效率。

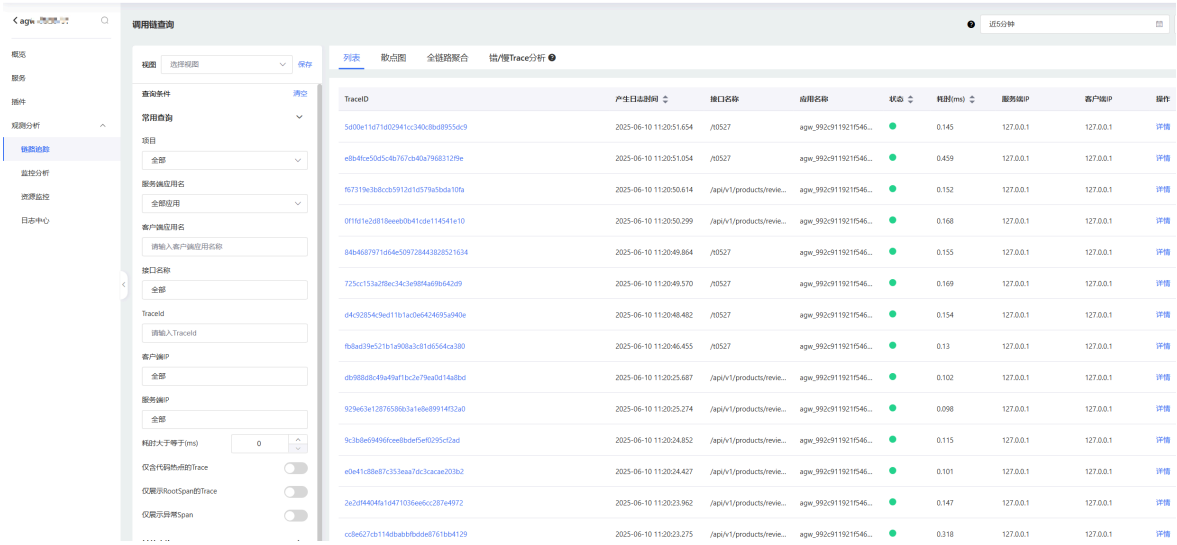
操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "实例"，并在顶部菜单栏选择地域。
3. 进入实例详情页，在左侧导航栏中，选择 "观测分析"。
4. 选择 "链路追踪"，有三个步骤进行开启：

- 检查是否开启委托授权，如未开启，需要点击 "立即创建"，创建名为CtyunAgwAdminTrust的委托授权。
- 检查是否开通应用监控性能服务，如未开通，需要点击 "立即开通"，开通应用监控性能服务。

- 检查是否接入链路追踪，如未接入，需要点击 "开启"，接入链路追踪。



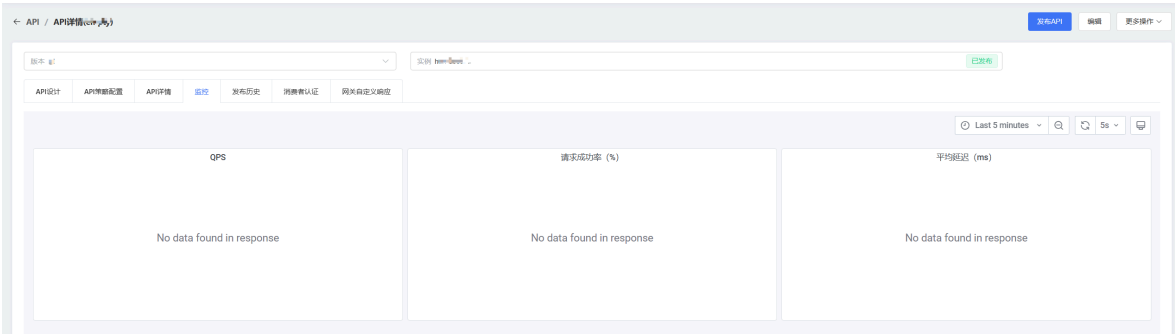


查看API监控数据

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入REST API详情页，选择实例 > "监控"。

4. 在API监控页面，可看到该REST API的QPS，请求成功率和平均延迟指标信息。



查看接口监控数据

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入REST API详情页，选择实例，选择接口>"接口监控"。
4. 在接口监控页面，可看到该接口的QPS，请求成功率和平均延迟指标信息。

查看路由监控数据

操作步骤

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 进入HTTP/WebSocket API详情页，选择路由>"监控"。
4. 在路由监控页面，可看到该路由的QPS，请求成功率和平均延迟指标信息。

> 说明
>
> - REST API/接口/路由监控都默认显示最近30分钟的数据，您可以在右上方设置时间进行筛选。
> - REST API/接口/路由监控都默认每隔5秒自动刷新，您可以在右上角设置刷新间隔。
> - 云原生API网关支持免费查看网关监控、网关资源监控和REST API、接口和路由的监控数据。

插件市场

管理插件

安装插件

安装插件是指将云原生API网关**插件市场**中的插件安装到具体的网关实例的过程。有两种方式可以安装插件：

操作步骤1：在插件市场安装

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。

3. 在插件市场页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击插件卡片上的"安装"，在弹出的安装插件框中选择需要使用此插件的网关实例，单击"确定"。

操作步骤2：在网关实例中安装

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择"实例"，并在顶部菜单栏选择地域。
3. 在实例页面，单击目标网关实例名称。
4. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。
5. 单击"安装插件"，在安装插件页面的快捷导航处，选择要安装的插件类型或者搜索插件名称，单击"插件"卡片，在弹出的安装插件框中，单击"安装并配置"。

卸载插件

当您想要将插件彻底从网关上删除时，您可以选择卸载插件。有两种方式可以卸载插件：

注意

卸载插件时，如果存在启用的插件规则，请先停用插件再卸载；如果插件未启用，卸载插件会将配置的插件规则一并删除。

操作步骤1：在插件市场卸载

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。
3. 在插件市场页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击要卸载的插件卡片。
4. 单击配置栏，在要卸载此插件的网关实例操作栏中，单击"卸载"。
5. 在弹出框中，点击"确认"按钮，页面提示卸载插件成功。

操作步骤2：在网关实例中卸载

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择"实例"，并在顶部菜单栏选择地域。
3. 在实例页面，单击目标网关实例名称。
4. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。
5. 在插件列表中，单击所要卸载插件操作列中的"卸载"。
6. 在弹出框中单击"确认"按钮，页面提示卸载插件成功。

官方插件

通用

real-ip插件

功能说明

real-ip 插件支持动态改变传递到网关的客户端的 IP 地址和端口。其工作方式和 NGINX 中的 ngx_http_realip_module 模块一样，并且更加灵活。

配置字段

名称	类型	填写要求	默认值	有效值	描述
source	string	必填		任何 NGINX 变量， 如 arg_realip 或 http_x_forwarded_for	动态设置客户端的 IP 地址和端口。如果该值不包含端口，则不会更改客户端的端口。如果该值的地址丢失或无效，则不会更改客户端的地址。
trusted_addresses	array[string]	可选		IP 或 CIDR 范围列表。	受信任地址， 用于动态设置 set_real_ip_from 字段。
recursive	boolean	可选	false		如果禁用递归搜索，则与受信任地址之一匹配的原始客户端地址将替换为配置的source中发送的最后一个地址。 如果启用递归搜索，则与受信任地址之一匹配的原始客户端地址将替换为配置的source中发送的最后一个非受信任地址。

配置示例

real-ip 使用示例

```
source:"arg_realip"
trusted_addresses:
 - 127.0.0.0/24
```

该配置表示从请求参数realip中获取客户端的IP地址和端口，并且设置发出请求的受信任地址CIDR范围为127.0.0.0/24，即127.0.0.0-127.0.0.255。

根据该场景请求路由

```
curl http://example.com/test?realip=1.2.3.4:27151
```

若发出请求的受信任地址在配置的CIDR范围内，则传递给网关的客户端IP地址和端口将被设置为1.2.3.4和27151

配置模板

#[必填]指定用于动态设置客户端的IP地址和端口。如果该值不包含端口，则不会更改客户端的端口。如果该值的地址丢失或无效，则不会更改客户端的地址。可填任意NGINX变量，比如arg_realip或http_x_forwarded_for。
source:"arg_realip"
#[可选]受信任地址，指定可信代理服务器的 IP 地址或CIDR范围列表，用于动态设置set_real_ip_from字段。
#trusted_addresses:
  #- 127.0.0.0/24
#[可选]如果禁用递归搜索，则与受信任地址之一匹配的原始客户端地址将替换为配置的source中发送的最后一个地址。如果启用递归搜索，则与受信任地址之一匹配的原始客户端地址将替换为配置的source中发送的最后一个非受信任地址。默认禁用。
#recursive: false

传输协议
fault-injection插件
功能说明

fault-injection 插件是故障注入插件，用于模拟服务故障。它可以在特定条件下人为引入延迟、返回错误状态码或自定义响应，从而帮助开发者和测试人员测试系统的容错能力和服务在异常情况下的表现。该插件可以和其他插件一起使用，并在其他插件执行前被执行。

配置字段

名称	类型	填写要求	默认值	有效值	描述
abort	object	abort与delay至少配置一个			abort 属性将直接返回给客户端指定的响应码并且终止其他插件的执行。
delay	object	abort与delay至少配置一个			delay 属性将延迟某个请求，并且还会执行配置的其他插件。

子项abort中每一项的配置字段说明如下。

名称	类型	填写要求	默认值	有效值	描述
http_status	integer	必填		[200, ...]	返回给客户端的 HTTP 状态码
body	string	可选			返回给客户端的响应数据。支持使用 NGINX 变量，如 client addr: $remote_addr

名称	类型	填写要求	默认值	有效值	描述
headers	object	可选			返回给客户端的响应头，可以包含 NGINX 变量，如 $remote_addr
percentage	integer	可选		[0, 100]	将被中断的请求占比
vars	array[]	可选			执行故障注入的规则，当规则匹配通过后才会执行故障注。vars 是一个表达式的列表，来自 lua-resty-expr。

子项delay中每一项的配置字段说明如下。

名称	类型	填写要求	默认值	有效值	描述
duration	number	必填			延迟时间，可以指定小数
percentage	integer	可选		[0, 100]	将被延迟的请求占比
vars	array[]	可选			执行请求延迟的规则，当规则匹配通过后才会执行故障注。vars 是一个表达式的列表，来自 lua-resty-expr。

注：vars 是由 lua-resty-expr 的表达式组成的列表，它可以灵活的实现规则之间的 AND/OR 关系，示例如下：

```
[
  [
    ["arg_name","==","jack"],
    ["arg_age","==",18]
  ],
  [
    ["arg_name2","==","allen"]
  ]
]
```

以上示例表示前两个表达式之间的关系是 AND，而前两个和第三个表达式之间的关系是 OR。

配置示例
场景1：故障注入

```
abort:
 http_status: 503
 body: "Fault Injection!"
```

根据该配置，路由请求时会被拦截，返回自定义的状态码和响应体。

场景2：请求延迟

```
delay:
 duration: 3
```

根据该配置，路由请求时会延迟3秒后再执行

场景3：带条件的故障注入和请求延迟

```
abort:
 http_status: 504
 body: "Fault Injection!"
 vars: [
           [
             ["arg_name","==","jack"]
           ]
         ]
delay:
 duration: 3
 vars: [
           [
             ["http_age","==","18"]
           ]
         ]
```

根据该配置，当请求参数中name值为jack的路由时会被拦截

```
curl http://example.com/test?name=jack

HTTP/1.1 503 Service Temporarily Unavailable
.....
Fault Injection!
```

当请求header中age值为18的路由时会延迟3秒执行

```
time curl http://example.com/test -H 'age: 18'

HTTP/1.1 200
.....
real 0m3.008s
user 0m0.003s
sys 0m0.003s
```

配置模板

```
#abort和delay属性至少要配置其中一个
#abort属性直接返回给客户端指定的响应码并且终止其他插件的执行
abort:
  #[必填]返回客户端的HTTP状态码，有效范围[200,...]
  http_status: 503
  #[可选]返回给客户端的响应数据
  #body: "Fault Injection!"
  #headers: {"X-Error-Code": "12345","X-Reason": "Invalid request"}
  #[可选]将被中断的请求占比，有效范围[0,100]
  #percentage: 100
  #[可选]执行故障注入的规则，当规则匹配通过后才会执行故障
  #vars: [[[ "arg_name","==","jack" ]]]
#delay属性将延迟某个请求，并且会执行配置的其他插件
delay:
  #[必填]延迟时间，可以指定小数
  duration: 3
  #[可选]将被延迟的请求占比，有效范围[0,100]
  #percentage: 100
  #[可选]执行请求延迟的规则，当规则匹配通过后才会执行故障
  #vars: [[[ "http_age","==","18" ]]]
```

安全防护

uri-blocker 插件

功能说明

uri-blocker 插件通过指定一系列 block_rules 来拦截用户请求，实现了基于URI屏蔽HTTP请求，并且自定义返回码和响应体，可以用于防护部分资源不对外部暴露。

配置字段

名称	类型	填写要求	默认值	有效值	描述
block_rules	array[string]	必填			正则过滤数组。它们都是正则规则，如果当前请求 URI 命中其中任何一个，则将响应代码设置为 rejected_code 以退出当前用户请求。例如：["root.exe", "root.m+"]。
rejected_code	integer	可选	403	[200, ...]	当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 状态代码。

名称	类型	填写要求	默认值	有效值	描述
rejected_msg	string	可选		非空	当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 响应体。
case_insensitive	boolean	可选	false		是否忽略大小写。当设置为 true 时，在匹配请求 URI 时将忽略大小写。

配置示例

uri-blocker 使用示例

```
block_rules:
 - root.m+
 - root.exe
rejected_code: 405
rejected_msg: “This uri is not allowed to be visited”
case_insensitive: true
```

根据该场景请求路由

```
curl http://example.com/root.exe
curl http://example.com/Root.exe
```

当前请求的URI命中了配置中的正则表达式，且在匹配时忽略大小写。请求将返回应答如下

```
HTTP/1.1 405 Not Allowed
......

{"error_msg":"This uri is not allowed to be visited"}
```

配置模板

```
#[必填]正则过滤数组，不可重复。如果当前请求 URI 命中其中任何一个，则将响应代码设置为 rejected_code 以退出当前用户请求
block_rules:
 - test.com+
#[可选]当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 状态代码。有效值[200,...]，默认值403
#rejected_code: 403
#[可选]当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 响应体
#rejected_msg: “This URI is not allowed to be accessed ”
#[可选]匹配URI时是否忽略大小写.默认false
#case_insensitive: false
```

ua-restriction插件

功能说明

ua-restriction 插件通过将指定 User-Agent 列入白名单或黑名单的方式来限制对服务或路由的访问。一种常见的场景是用来设置爬虫规则，User-Agent 是客户端在向服务器发送请求时的身份标识，可以将一些爬虫程序的请求头列入 ua-restriction 插件的白名单或黑名单中。

配置字段

名称	类型	填写要求	默认值	有效值	描述
allowlist	array[string]	可选			加入白名单的 User-Agent。
denylist	array[string]	可选			加入黑名单的 User-Agent。
message	string	可选	"Not allowed"	字符数[1,1024]	当未允许的 User-Agent 访问时返回的信息。
bypass_missing	boolean	可选	false		当设置为 true 时，如果 User-Agent 请求头不存在或格式有误时，将绕过检查。

注意

1. allowlist 和 denylist 可以同时启用。同时启用时，插件会根据 User-Agent 先检查 allowlist，再检查 denylist。
2. 如果黑白名单列表中采用*作为通配符，则需要通过引号添加字段值

配置示例

设置 User-Agent 白名单

```
allowlist:
  - ".*Go-http-client.*"
```

若不做该配置，默认的Golang网络库请求被禁止访问。

设置 User-Agent 黑名单

```
denylist:
  - "tools.*"
```

根据该配置，以下请求会被禁止。

```
curl http://example.com -H 'User-Agent: tools/1.1'
curl http://exmaple.com -H 'User-Agent: tools'
```

该请求的响应结果为

```
HTTP/1.1 403 Forbidden
```

.....

{"message":"Not allowed"}

同时设置白名单与黑名单

```
allowlist:
 - "tools.*"
 - ".*Go-http-client.*"
denylist:
 - "tools.*"
```

由于白名单的优先级更高，所以在请求带有'User-Agent: tools' 的路由时会请求通过。

配置模板

```
#[可选]加入白名单的User-Agent列表，支持正则表达式。和黑名单同时配置时，白名单优先级更高
allowlist:
 - my-bot1
#[可选]加入黑名单的User-Agent列表，支持正则表达式
denylist:
 - my-bot2
#[可选]未允许的User-Agent访问时返回的信息。有效长度1-1024，默认为Not allowed
#message: “Not allowed”
#[可选]当设置为 true 时，如果 User-Agent 请求头不存在或格式有误时，将绕过检查。默认false
#bypass_missing: false
```

referer-restriction插件

功能说明

referer-restriction 插件通过将指定 Referer 列入白名单或黑名单的方式来限制对服务或路由的访问。Referer是 HTTP 请求头的一部分，用于标识当前请求的来源页面（即用户从哪个页面点击链接进入了当前页面）。

配置字段

名称	类型	填写要求	默认值	有效值	描述
whitelist	array[string]	与blacklist二选一			白名单域名列表。域名开头可以用 * 作为通配符。
blacklist	array[string]	与whitelist二选一			黑名单域名列表。域名开头可以用 * 作为通配符。
message	string	可选	"Your referer host is not allowed"	字符数[1,1024]	当未允许访问时返回的信息。

名称	类型	填写要求	默认值	有效值	描述
bypass_missing	boolean	可选	false		当设置为 true 时，如果Referer请求头不存在或格式有误时，将绕过检查。

注意

配置模板中对于黑白名单域名列表，如果采用*作为通配符，则需要通过引号填写字段值。

配置示例

设置 Referer 白名单

```
whitelist:
  - "ctyun.com"
  - "*.ctyun.com"
```

若不做该配置，从ctyun.com页面跳转的链接被禁止访问。

设置 Referer 黑名单

```
blacklist:
  - "black.com"
```

根据该配置，以下请求会被禁止。

```
curl http://example.com -H 'Referer: http://black.com/test'
```

该请求的响应结果为

```
HTTP/1.1 403 Forbidden
......

{"message":"Your referer host is not allowed"}
```

配置模板

```
#[必填，黑白名单二选一]黑/白名单域名列表。域名开头可以用 * 作为通配符。
whitelist:
  - "*.ctyun.com"
#blacklist:
#  - "b.com"
#[可选]未允许访问时返回的信息。有效长度1-1024，默认为Your refer host is not allowed
#message: "Your refer host is not allowed"
#[可选]当设置为 true 时，如果 Referer请求头不存在或格式有误时，将绕过检查。默认false
#bypass_missing: false
```

csrf插件

功能说明

csrf 插件可保护用户的 API 免于 CSRF 攻击。

在此插件运行时，GET、HEAD 和 OPTIONS 会被定义为 safe-methods，其他的请求方法则定义为 unsafe-methods。因此 GET、HEAD 和 OPTIONS 方法的调用不会被检查拦截。

使用 GET 请求 API 时，在响应中会有一个携带了加密 Token 的 Cookie。Token 字段名称为插件配置中的 name 值，默认为 agw-csrf-token。

注意

每一个请求都会返回一个新的 Cookie。

在后续对该路由进行的 unsafe-methods 请求中，需要从 Cookie 中读取加密的 Token，并在请求头中携带该 Token。请求头字段的名称为插件属性中的 name。

配置字段

名称	类型	填写要求	默认值	有效值	描述
key	string	必填			加密 Token 的密钥。
name	string	可选	agw-csrf-token		生成的 Cookie 中的 Token 名称，需要使用此名称在请求头携带 Cookie 中的内容。
expires	number	可选	7200		CSRF Cookie 的过期时间，单位为秒。当设置为 0 时，会忽略 CSRF Cookie 过期时间检查。

配置示例

key:"edd1c9f034335f136f87ad84b625c8f1"

启用插件后，使用 curl 命令尝试直接对该路由发起 POST 请求，会返回 Unauthorized 字样的报错提示：

```
curl -i http://127.0.0.1:27151/hello -X POST
HTTP/1.1 401 Unauthorized

{"error_msg":"no csrf token in headers"}
```

当发起 GET 请求时，返回结果中会有携带 Token 的 Cookie：

```
curl -i http://127.0.0.1:27151/hello -i
HTTP/1.1 200 OK
```

Set-Cookie: agw-csrf-token=eyJyYW5k
b20iOjAuNzgyMDk5MTE4NjUxNjQsInNpZ24iOiIwNTEyZDFkZDRiMmNjYTIxMTRlYWRhYWUxOTkxNmNiZGYzMWE2ZGIOMzRkN
Apr-25 12:01:53 GMT

在请求之前，用户需要从 Cookie 中读取 Token，并在后续的 unsafe-methods 请求的请求头中携带。

例如，你可以在客户端使用 js-cookie 读取 Cookie，使用 axios 发送请求：

```
const token = Cookie.get('agw-csrf-token');

const instance = axios.create({
  headers: {'agw-csrf-token': token}
});
```

使用 curl 命令发送请求，确保请求中携带了 Cookie 信息，如果返回 200 HTTP 状态码则表示请求成功：

```
curl -i http://127.0.0.1:27151/hello -X POST -H 'agw-csrf-token: eyJyYW5k
b20iOjAuNjg4OTcyMzA4ODM1NDMsImV4cGlyZXMiOjcyMDAsInNpZ24iOiJcL09uZEF4WUZDZGYwSnBiNDlKREtnbzVoYkJjb
b 'agw-csrf-token=eyJyYW5k
b20iOjAuNjg4OTcyMzA4ODM1NDMsImV4cGlyZXMiOjcyMDAsInNpZ24iOiJcL09uZEF4WUZDZGYwSnBiNDlKREtnbzVoYkJjb
HTTP/1.1 200 OK
```

配置模板

```
#必填，加密 Token 的密钥
key: "edd1c9f034335f136f87ad84b625c8f1"
#[可选]默认为agw-csrf-token，生成的 Cookie 中的 Token 名称，需要使用此名称在请求头携
带 Cookie 中的内容
#name: "agw-csrf-token"
#[可选]CSRF Cookie 的过期时间
#expires: 7200
```

流量管控

limit-req插件

功能说明

limit-req 插件支持限制单个客户端对服务的请求速率。该插件内部使用的算法为漏桶算法。

配置字段

名称	类型	填写要求	默认值	有效值	描述
rate	integer	必填		rate>0	指定的请求速率（以秒为单位），请求速率超过 rate 但没有超过（rate + burst）的请求会被延时处理。

名称	类型	填写要求	默认值	有效值	描述
burst	integer	必填		burst>0	请求速率超过（rate + burst）的请求会被直接拒绝。
key_type	string	可选	"var"	["var", "var_combination"]	要使用的用户指定 key 的类型。
key	string	必填		["remote_addr", "server_addr", "http_x_real_ip", "http_x_forwarded_for", "consumer_name"]	用来做请求计数的依据，当前接受的 key 有：remote_addr（客户端 IP 地址），server_addr（服务端 IP 地址），请求头中的 X-Forwarded-For 或 X-Real-IP，consumer_name（Consumer 的 username）。
rejected_code	integer	可选	503	[200,599]	当超过阈值的请求被拒绝时，返回的 HTTP 状态码。
rejected_msg	string	可选		非空	当超过阈值的请求被拒绝时，返回的响应体。
nodelay	boolean	可选	false		当设置为 true 时，请求速率超过 rate 但没有超过（rate + burst）的请求不会加上延迟；当设置为 false，则会加上延迟。
allow_degradation	boolean	可选	false		当设置为 true 时，如果限速插件功能临时不可用，将会自动允许请求继续。

配置示例

limit-req 使用示例

```
rate: 2
burst: 3
rejected_code: 503
```

key_type: "var"
key: "remote_addr"
rejected_msg: "Requests are too frequent, please try again later."

根据该场景请求路由

curl http://example.com/test

当请求速率在2(rate)次/秒内时，请求正常返回。

当请求速率超过2(rate)次/秒，但在5(rate+burst)次/秒内时，请求将被延迟处理，延迟时间根据漏桶算法。

当请求速率超过5(rate+burst)次/秒时，请求被限制，返回响应如下。

HTTP/1.1 503 Service Temporarily Unavailable
.....

{"error_msg":"Requests are too frequent, please try again later"}

配置模板

```
# [必填]指定的请求速率（以秒为单位），请求速率超过rate但没有超过（rate+burst）的请求会被延时处理, rate需大于0.
rate: 1
# [必填]请求速率超过（rate+burst）的请求会被直接拒绝, burst需大于等于0.
burst: 2
# [可选]请求超过阈值被拒绝时，返回的 HTTP 状态码。默认503，有效范围[200, 599].
#rejected_code: 503
# [可选]key 的类型
#key_type: "var"
# [必填]用来做请求计数的依据.当前接受的 key 有：remote_addr，server_addr, 请求头中的 X-Forwarded-For 或 X-Real-IP，consumer_name。
key: "remote_addr"
# [可选]当超过阈值的请求被拒绝时，返回的HTTP状态码
#rejected_msg: "Requests are too frequent, please try again later."
# [可选]当设置为 true 时，请求速率超过rate但没有超过（rate+burst）的请求不会加上延迟。默认 false
#nodelay: false
# [可选]当设置为 true 时，如果限速插件功能临时不可用，将会自动允许请求继续。默认 false
#allow_degradation: false
```

limit-conn插件

功能说明

limit-conn 插件用于限制客户端对单个服务的并发请求数。当客户端对路由的并发请求数达到限制时，可以返回自定义的状态码和响应信息

配置字段

名称	类型	填写要求	默认值	有效值	描述
conn	integer	必填		conn > 0	允许的最大并发请求数。超过 conn 的限制、但是低于 conn + burst 的请求，将被延迟处理。
burst	integer	必填		burst >= 0	每秒允许被延迟处理的额外并发请求数。
default_conn_delay	number	必填		default_conn_delay > 0	默认的典型连接（或请求）的处理延迟时间。
only_use_default_delay	boolean	可选	false		延迟时间的严格模式。当设置为 true 时，将会严格按照设置的 default_conn_delay 时间来进行延迟处理。
key_type	string	可选	"var"	["var", "var_combination"]	key 的类型。
key	string	可选			用来做请求计数的依据。如果 key_type 为 "var"，那么 key 会被当作变量名称，如 remote_addr 和 consumer_n 如果 key_type 为 "var_combinat 那么 key 会当作变量组合，如 $remote_addr $consumer_name；如果 key 的值为空，$remote_addr 会被作为默认 key。
rejected_code	integer	可选	503	[200,...,599]	当请求数超过 conn + burst 阈值时，返回的 HTTP 状态码。
rejected_msg	string	可选		非空	当请求数超过 conn + burst 阈值时，返回的信息。

名称	类型	填写要求	默认值	有效值	描述
allow_degradation	boolean	可选	false		当设置为 true 时，启用插件降级并自动允许请求继续。

配置示例

limit-conn 使用示例

```
conn: 1
burst: 0
default_conn_delay: 0.1
rejected_code: 503
key_type: "var"
key: "remote_addr"
```

根据该配置场景，请求以下路由。

```
curl -i http://example.com/index.html?sleep=20 &
curl -i http://example.com/index.html?sleep=20
```

在这条路由资源上，网关将只允许一个连接；当有更多连接进入时，网关会直接返回 503 HTTP 状态码，拒绝连接。

```
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>openresty</center>
</body>
</html>
```

配置模板

```
# [必填]最大并发连接请求数
conn: 1
# [必填]允许被延迟处理的并发请求数
burst: 0
# [必填]默认的典型连接（或请求）的处理延迟时间
default_conn_delay: 0.1
# [可选]延迟时间的严格模式。 如果设置为true的话，将会严格按照设置的时间来进行延迟
#only_use_default_delay: false
# [可选]key 的类型
#key_type: "var"
# [必填]用来做请求计数的依据
key: "remote_addr"
# [可选]请求超过 conn + burst 阈值时，返回的 HTTP 状态码
#rejected_code: 429
```

[可选]当设置rejected_msg时，非空。默认可不填
#rejected_msg: "Requests are too frequent, please try again later."
[可选]当插件功能临时不可用时是否允许请求继续。当值设置为 true 时则自动允许请求继续，默认值是 false
#allow_degradation: false

limit-count插件

功能说明

limit-count 插件使用固定时间窗口算法，主要用于限制单个客户端在指定的时间范围内对服务的总请求数，并且会在 HTTP 响应头中返回剩余可以请求的个数.

配置字段

名称	类型	填写要求	默认值	有效值	描述
count	integer	必填		count> 0	每个客户端在指定时间窗口内的总请求数量阈值。
time_window	integer	必填		time_window> 0	时间窗口的大小（以秒为单位）。超过该属性定义的时间，则会重新开始计数。
rejected_code	integer	可选	503	[200,...,599]	当请求超过阈值被拒绝时，返回的 HTTP 状态码。
key_type	string	可选	"var"	["var", "var_combination", "constant"]	key 的类型。

名称	类型	填写要求	默认值	有效值	描述
key	string	可选	"remote_addr"		用来做请求计数的依据。如果 key_type 为 constant，那么 key 会被当作常量；如果 key_type 为 var，那么 key 会被当作变量；如果 key_type 为 var_combinati 那么 key 会被当作变量组合，如 $remote_addr $consumer_name，插件会同时受 $remote_addr 和 $consumer 个变量的约束；如果 key 的值为空，$remote_addr 会被作为默认 key。
rejected_msg	string	可选		非空	当请求超过阈值被拒绝时，返回的响应体。
allow_degradation	boolean	可选	false		当插件功能临时不可用时（例如 Redis 超时），当设置为 true 时，则表示可以允许插件降级并进行继续请求的操作。
show_limit_quota_header	boolean	可选	true		当设置为 true 时，在响应头中显示 X-RateLimit-Limit（限制的总请求数）和 X-RateLimit-Remaining（剩余还可以发送的请求数）字段。

配置示例

limit-count 使用示例

```
count: 2
time_window: 60
rejected_code: 503
key_type: "var"
key: "remote_addr"
```

根据该配置场景，其限制了 60 秒内请求只能访问 2 次。请求以下路由

```
curl -i http://example.com/index.html
```

在执行测试命令的前两次都会正常访问。其中响应头中包含了 X-RateLimit-Limit 和 X-RateLimit-Remaining 和 X-RateLimit-Reset 字段，分别代表限制的总请求数和剩余还可以发送的请求数以及计数器剩余重置的秒数：

```
HTTP/1.1 200 OK
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

当第三次进行测试访问时，会收到包含 503 HTTP 状态码的响应头，目前在拒绝的情况下，也会返回相关的头，表示插件生效：

```
HTTP/1.1 503 Service Temporarily Unavailable
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

配置模板

基础配置案例

```
# [必填]时间窗口内的请求数量阈值。
count: 30
# [必填]时间窗口的大小（以秒为单位）
time_window: 60
# [可选]请求超过阈值被拒绝时，返回的 HTTP 状态码
#rejected_code: 429
# [可选]key 的类型
#key_type: "var"
# [可选]用来做请求计数的依据
#key: "remote_addr"
# [可选]当设置rejected_msg时，非空。默认可不填
# rejected_msg: "Requests are too frequent, please try again later."
# [可选]当限流插件功能临时不可用时（例如，Redis 超时）是否允许请求继续。默认 false
#allow_degradation: false
# [可选]是否在响应头中显示 X-RateLimit-Limit 和 X-RateLimit-Remaining （限制的总请求数和剩余还可以发送的请求数），默认 true
#show_limit_quota_header: true
```

消费者限流插件

功能说明

limit-count-by-client 插件使用固定时间窗口算法，主要用于限制消费者在单个客户端在指定的时间范围内对服务的总请求数，并且会在 HTTP 响应头中返回剩余可以请求的个数.

配置字段

名称	类型	填写要求	默认值	有效值	描述
count	integer	必填		count> 0	每个客户端在指定时间窗口内的总请求数量阈值。
time_window	integer	必填		time_window> 0	时间窗口的大小（以秒为单位）。超过该属性定义的时间，则会重新开始计数。
app_white_list	array[string]	可选			消费者白名单，白名单内的消费者不受该限流规则影响。
rejected_code	integer	可选	503	[200,...,599]	当请求超过阈值被拒绝时，返回的 HTTP 状态码。
rejected_msg	string	可选		非空	当请求超过阈值被拒绝时，返回的响应体。
error_interrupt	boolean	可选	false		在异常错误时，是否中断用户请求，false 不中断，true 中断，默认不中断。
rule_map	object	可选			消费者限速规则，约定对指定消费者的限速规则。

注意

- 消费者名称可到消费者详情页中查找。
- 当在作用范围内未开启认证授权时，不存在有效的消费者，此时`app_white_list`和`rule_map`的配置不会生效，将按最外层默认的限流策略生效。

配置示例

limit-count-by-client 使用示例

```
count: 2
time_window: 60
rejected_code: 503
app_white_list:
  - "consumer_name_1"
```

```
rule_map:
  consumer_name_2:
    count: 3
    time_window: 10
  consumer_name_3:
    count: 3
    time_window: 10
```

根据该配置场景，存在多条限流规则：

1. 消费者consumer_name_1不受限流影响
2. 消费者consumer_name_2与consumer_name_3在10秒内限流3
3. 其他消费者在60秒内限流2

```
curl -i http://example.com/index.html
```

在执行测试命令的前两次都会正常访问。其中响应头中包含了 X-RateLimit-Limit 和 X-RateLimit-Remaining 和 X-RateLimit-Reset 字段，分别代表限制的总请求数和剩余还可以发送的请求数以及计数器剩余重置的秒数：

```
HTTP/1.1 200 OK
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

当第三次进行测试访问时，会收到包含 503 HTTP 状态码的响应头，目前在拒绝的情况下，也会返回相关的头，表示插件生效：

```
HTTP/1.1 503 Service Temporarily Unavailable
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

配置模板

基础配置案例

```
# 时间窗口内的请求数量阈值
# [必填]count数值为当前网关实例集群的限流计数。
count: 30
# [必填]时间窗口的大小（以秒为单位）
time_window: 60
# [可选] 消费者白名单，字符串类型最多20个
app_white_list:
  - "app_name_1"
  - "app_name_2"
# [可选]请求超过阈值被拒绝时，返回的 HTTP 状态码
rejected_code: 429
```

```
# [可选]当设置rejected_msg时，非空。默认可不填
# rejected_msg: "Requests are too frequent, please try again later."
# [可选] 在异常错误时，是否中断用户请求，false 不中断，true 中断，默认不中断
error_interrupt: false
# [可选] APP（消费者）限速规则，约定对指定APP(消费者)的限速规则
rule_map:
  # 消费者名称
  app_1:
    # [可选] 当前消费者的限流计数，不传时取最外层默认的count值
    count: 3
    # [] 当前消费者的限流时间窗口大小（以秒为单位），不传时取最外层的默认time_window值
    time_window: 10
  # 消费者名称
  app_2:
    # [可选] 当前消费者的限流计数，不传时取最外层默认的count值
    count: 3
    # [可选] 当前消费者的限流时间窗口大小（以秒为单位），不传时取最外层的默认time_window值
    time_window: 10
```

结果缓存插件

功能说明

proxy-cache 插件提供了根据缓存键缓存响应的功能。该插件支持基于磁盘和基于内存的缓存选项，用于缓存 GET、POST 和 HEAD 请求。

可以根据请求 HTTP 方法、响应状态代码、请求头值等有条件地缓存响应。

配置字段

名称	类型	填写要求	默认值	有效值	描述
cache_strategy	string	可选	disk	["disk","memory"]	缓存策略。缓存在磁盘还是内存中。
cache_zone	string	可选	disk_cache_one	["disk_cache_one","memory_cache"]	与缓存策略一起使用的缓存区域。该值应与缓存策略相对应。例如，当使用内存缓存策略 memory 时，应该使用内存缓存区域 memory_cache。
cache_key	array[string]	可选	["$host", "$request_uri"]		缓存 key，可以使用变量。例如：["$host", "$uri", "-cache-id"]。

名称	类型	填写要求	默认值	有效值	描述
cache_bypass	array[string]	可选			当该属性的值不为空或者非 0 时则会跳过缓存检查，即不在缓存中查找数据，可以使用变量，例如：["$arg_bypass"]。
cache_method	array[string]	可选	["GET", "HEAD"]	["GET", "POST", "HEAD"]	根据请求 method 决定是否需要缓存。
cache_http_status	array[integer]	可选	[200, 301, 404]	[200,599]	根据 HTTP 响应码决定是否需要缓存。
hide_cache_headers	boolean	可选	false		当设置为 true 时将 Expires 和 Cache-Control 响应头返回给客户端。
cache_control	boolean	可选	false		当设置为 true 时遵守 HTTP 协议规范中的 Cache-Control 的行为。
no_cache	array[string]	可选			当此参数的值不为空或非 0 时将不会缓存数据，可以使用变量。
cache_ttl	integer	可选	300s	cache_ttl>=1	当选项 cache_control 未开启或开启以后服务端没有返回缓存控制头时，提供的默认缓存时间。

配置示例

在磁盘上缓存数据

磁盘缓存策略具有系统重启时数据持久性以及与内存缓存相比具有更大存储容量的优势。它适用于优先考虑耐用性且可以容忍稍大的缓存访问延迟的应用程序。

使用磁盘缓存策略时，缓存 TTL 由响应标头 Expires 或 Cache-Control 中的值确定。如果这些标头均不存在，或者 网关 由于上游不可用而返回 502 Bad Gateway 或 504 Gateway Timeout，则缓存 TTL 默认为 10 秒。

以下示例演示了如何在路由上使用 proxy-cache 插件将数据缓存在磁盘上：

```
cache_strategy: disk
cache_zone: disk_cache_one
```

在在线调试中请求对应的路由或者接口。

如果返回 200 HTTP 状态码，并且响应头中包含 Agw-Cache-Status 字段，则表示该插件已启用。如果你是第一次请求该路由，数据未缓存，那么 Agw-Cache-Status 字段应为 MISS。

```
HTTP/1.1 200 OK
...
Agw-Cache-Status: MISS
```

此时再次请求该路由，如果返回的响应头中 Agw-Cache-Status 字段变为 HIT，则表示数据已被缓存，插件生效。

```
HTTP/1.1 200 OK
...
Agw-Cache-Status: HIT
```

等待缓存在 TTL 之后过期，再次发送相同的请求。您应该看到带有以下标头的 HTTP/1.1 200 OK 响应，表明缓存已过期：

```
HTTP/1.1 200 OK
...
Agw-Cache-Status: EXPIRED
```

在内存中缓存数据

内存缓存策略具有低延迟访问缓存数据的优势，因为从 RAM 检索数据比从磁盘存储检索数据更快。它还适用于存储不需要长期保存的临时数据，从而可以高效缓存频繁更改的数据。

以下示例演示了如何在路由上使用 proxy-cache 插件在内存中缓存数据。

```
cache_strategy: memory
cache_zone: memory_cache
cache_ttl: 10
```

有条件地缓存响应

以下示例演示了如何配置 proxy-cache 插件以有条件地缓存响应。

使用 proxy-cache 插件创建路由并配置 no_cache 属性，这样如果 URL 参数 no_cache 和标头 no_cache 的值中至少有一个不为空且不等于 0，则不会缓存响应：

```
no_cache: ["$arg_no_cache", "$http_no_cache"]
```

向路由发送一些请求，其中 URL 参数的 no_cache 值表示绕过缓存：

```
curl -i "http://127.0.0.1:27151/anything?no_cache=1"
```

您应该收到所有请求的 HTTP/1.1 200 OK 响应，并且每次都观察到以下标头：

```
Agw-Cache-Status: EXPIRED
```

向路由发送一些其他请求，其中 URL 参数 no_cache 值为零：

```
curl -i "http://127.0.0.1:27151/anything?no_cache=0"
```

您应该收到所有请求的 HTTP/1.1 200 OK 响应，并开始看到缓存被命中：

Agw-Cache-Status: HIT

您还可以在 no_cache 标头中指定以下值：

curl -i "http://127.0.0.1:27151/anything" -H "no_cache: 1"

响应不应该被缓存：

Agw-Cache-Status: EXPIRED

有条件地从缓存中检索响应

以下示例演示了如何配置 proxy-cache 插件以有条件地从缓存中检索响应。

使用 proxy-cache 插件创建路由并配置 cache_bypass 属性，这样如果 URL 参数 bypass 和标头 bypass 的值中至少有一个不为空且不等于 0，则不会从缓存中检索响应：

cache_bypass: ["$arg_bypass", "$http_bypass"]

向路由发送一个请求，其中 URL 参数值为 bypass，表示绕过缓存：

curl -i "http://127.0.0.1:27151/anything?bypass=1"

您应该看到带有以下标头的 HTTP/1.1 200 OK 响应：

Agw-Cache-Status: BYPASS

向路由发送另一个请求，其中 URL 参数 bypass 值为零：

curl -i "http://127.0.0.1:27151/anything?bypass=0"

您应该看到带有以下标头的 HTTP/1.1 200 OK 响应：

Agw-Cache-Status: MISS

您还可以在 bypass 标头中指定以下值：

curl -i "http://127.0.0.1:27151/anything" -H "bypass: 1"

响应应该显示绕过缓存：

Agw-Cache-Status: BYPASS

配置模板

```
# [可选]缓存策略，缓存在磁盘还是内存中。disk或者memory，默认disk。
cache_strategy: disk
# [可选]cache_strategy为disk时，填disk_cache_one；为memory时，填memory_cache。
cache_zone: disk_cache_one
# [可选]缓存key，可以使用变量，类型为数组。默认为["$host","$request_uri"]
cache_key: ["$uri", "-cache-id"]
```

[可选]是否跳过缓存检索，即不在缓存中查找数据，可以使用变量，当此参数的值不为空时将会跳过缓存的检索，类型为数组。
#cache_bypass: ["$arg_bypass"]
[可选]根据请求method决定是否需要缓存,类型为数组。method可选["GET", "POST", "HEAD"]，默认为["GET", "HEAD"]
cache_method: ["GET", "POST", "HEAD"]
[可选]根据响应码决定是否需要缓存，,类型为数组。支持范围[200,599]，默认为[200, 301, 404]
cache_http_statu: [200, 301, 404]
[可选]是否将Expires和Cache-Control响应头返回给客户端，默认为false
#hide_cache_header: false
[可选]是否遵守HTTP协议规范中的Cache-Control的行为,默认为false
#cache_control: false
[可选]是否缓存数据，可以使用变量，当此参数的值不为空时将不会缓存数据,类型为数组。
#no_cache: ["$arg_no_cache"]
[可选]当选项cache_control未开启或开启以后服务端没有返回缓存控制头时，提供缓存时间，默认300秒
cache_ttl: 300

client-control插件

功能说明

client-control 插件通过设置客户端请求体大小上限来动态控制客户端的请求。当设置较大的限制时可能导致内存使用增加，需根据实际需求合理配置。

配置字段

名称	类型	填写要求	默认值	有效值	描述
max_body_size	integer	可选		[0,...]	动态设置 client_max_body_size 的大小

配置示例

client-control 使用示例

max_body_size: 1

根据该场景请求路由

curl http://example.com/test -d '123'

由于请求路由的请求体大小超过了所设置的客户端请求体大小上限，请求返回413.

HTTP/1.1 413 Request Entity Too Large
.....

<html>
<head><title>413 Request Entity Too Large</title></head>
<body>
<h1>413 Request Entity Too Large</h1>
<hr> openresty

Powered by CGW. </body>
</html>

配置模板

#[可选]设置客户端请求体大小上限.有效范围[0,...]
#max_body_size: 1024

request-validation插件
功能说明

request-validation 插件用于提前验证向上游服务转发的请求。该插件使用 JSON Schema 机制进行数据验证，可以验证请求的 body 及 header 数据。

配置字段

名称	类型	填写要求	默认值	有效值	描述
header_schema	object	可选,至少配置 header_schema 和 body_schema 中任意一个，两者也可以同时使用。			header 数据的 schema 数据结构。
body_schema	object	可选,至少配置 header_schema 和 body_schema 中任意一个，两者也可以同时使用。			body 数据的 schema 数据结构。
rejected_code	integer	可选	400	[200,...,599]	当请求被拒绝时要返回的状态码。
rejected_msg	string	可选			当请求被拒绝时返回的信息。

配置示例

request-validation 使用示例

body_schema: { \"type\": \"object\", \"required\": [\"required_payload\"], \"properties\": {\"required_payload\": {\"type\": \"string\"},\"boolean_payload\": {\"type\": \"boolean\"} } }
rejected_code: 400
rejected_msg: "customize reject message"

请求以下路由

curl -i --header "Content-Type: application/json" \
> --request POST \
> --data '{"boolean-payload":true}' \
> http://example.com/test

请求中的body没有 request_payload 字段，请求被拒绝，返回结果为

HTTP/1.1 400 Bad Request

.....

customize reject message

常见的schema数据结构示例：

枚举（Enum）验证

```
{
  "body_schema": {
    "type": "object",
    "required": ["enum_payload"],
    "properties": {
      "enum_payload": {
        "type": "string",
        "enum": ["enum_string_1", "enum_string_2"],
        "default": "enum_string_1"
      }
    }
  }
}
```

布尔（Boolean）验证

```
{
  "body_schema": {
    "type": "object",
    "required": ["bool_payload"],
    "properties": {
      "bool_payload": {
        "type": "boolean",
        "default": true
      }
    }
  }
}
```

数字范围（Number or Integer）验证

```
{
  "body_schema": {
    "type": "object",
    "required": ["integer_payload"],
    "properties": {
      "integer_payload": {
        "type": "integer",
        "minimum": 1,
        "maximum": 65535
      }
    }
```

```
  }
}
```

字符串长度（String）验证

```
{
  "body_schema":{
    "type":"object",
    "required":["string_payload"],
    "properties":{
      "string_payload":{
        "type":"string",
        "minLength":1,
        "maxLength":32
      }
    }
  }
}
```

正则表达式（Regex）验证

```
{
  "body_schema":{
    "type":"object",
    "required":["regex_payload"],
    "properties":{
      "regex_payload":{
        "type":"string",
        "minLength":1,
        "maxLength":32,
        "pattern":"[[^[a-zA-Z0-9_]+$]]"
      }
    }
  }
}
```

数组（Array）验证

```
{
  "body_schema":{
    "type":"object",
    "required":["array_payload"],
    "properties":{
      "array_payload":{
        "type":"array",
        "minItems":1,
        "items":{
          "type":"integer",
          "minimum":200,
```

```
            "maximum": 599
          },
          "uniqueItems": true,
          "default": [200, 302]
        }
      }
    }
}
```

多字段组合（Combined）验证

```
{
  "body_schema": {
    "type": "object",
    "required": ["boolean_payload", "array_payload", "regex_payload"],
    "properties": {
      "boolean_payload": {
        "type": "boolean"
      },
      "array_payload": {
        "type": "array",
        "minItems": 1,
        "items": {
          "type": "integer",
          "minimum": 200,
          "maximum": 599
        },
        "uniqueItems": true,
        "default": [200, 302]
      },
      "regex_payload": {
        "type": "string",
        "minLength": 1,
        "maxLength": 32,
        "pattern": "[[^[a-zA-Z0-9_]+$]]"
      }
    }
  }
}
```

配置模板

```
#[必填]header_schema和body_schema至少二选一，可以同时使用
#header数据的schema数据结构
#header_schema:
#body数据的schema数据结构
#body_schema:
#[可选]当请求被拒绝时要返回的状态码。默认值400，有效范围[200,...,599]
```

#rejected_code: 400
#[可选]当请求被拒绝时返回的信息。有效长度为1-256
#rejected_msg: “The request param validation failed”

插件配置管理

启用插件配置

说明

插件的生效范围：

- **路由/接口级插件规则**：请求匹配到具体接口或路由时生效，优先级高于 API 规则。
- **API级插件规则**：请求匹配到某个 API 时生效，作用于其下所有接口。
- **实例级插件规则**：启用即生效网关全局，独立执行，且在接口和 API 规则前执行。

注意

如果在API和路由/接口上配置了同一条插件，由于接口/路由的优先级高于API，则只会执行路由/接口上的插件配置。

操作步骤1：在插件市场中启用

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。
3. 在插件市场页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击插件卡片上的"安装"，在弹出的安装插件框中选择需要使用此插件的网关实例，单击"确定"。
4. 在配置栏中，单击目标网关实例操作列下的"规则配置"，在规则配置页面选择生效范围。
   - 当选择路由/接口级插件规则时，单击"添加规则"，在添加路由/接口级别规则页面，打开"启用"状态，选择生效目标并配置插件规则，单击确定。
   - 当选择API级插件规则时，单击"添加规则"，在添加API级别规则页面，打开"启用"状态，选择生效目标并配置插件规则，单击确定。
   - 当选择实例级插件规则时，打开"启用"状态并配置插件规则，单击保存。

操作步骤2：在实例上启用

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择"实例"，并在顶部菜单栏选择地域。
3. 在实例页面，单击目标网关实例名称。
4. 在左侧导航栏，选择"插件"，并在顶部菜单栏选择地域。
5. 单击插件列表操作列中的"规则配置"，为所选插件配置规则并选择生效范围。

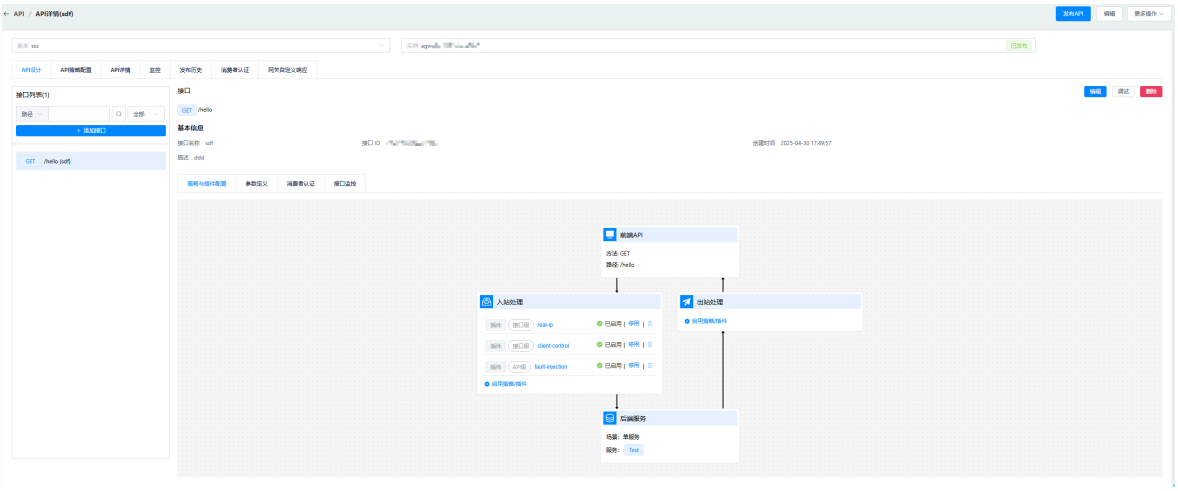
在API列表中配置插件规则

注意

当接口与API同时配置同类型插件规则时，接口的规则优先。

操作步骤

1. 登录云原生API网关控制台。
2. 在左侧导航栏，单击"API"，在顶部菜单栏选择地域。
3. 在API列表中选择要挂载的API，在接口列表中选择全部接口或要挂载的接口。
4. 在右侧的策略配置区块，单击入站处理或出站处理中的启用策略/插件。
5. 在弹出的启用策略/插件抽屉中，单击"添加插件"。
6. 在快捷导航处，选择要安装的插件类型或者搜索插件名称，单击"插件"卡片：
   - 如果插件未安装，在安装插件的弹出框中单击安装并配置，在启用插件的弹框中配置插件规则，并选择"启用"状态。
   - 如果插件已安装，在启用插件的弹框中，配置插件规则，并选择"启用"状态。
7. 单击"确定"，返回API的挂载列表，可以看到接口的插件挂载情况和启用状态。



告警管理

设置告警规则

前提：已开通云原生API网关实例、租户在当前资源池中已开通应用性能监控APM服务

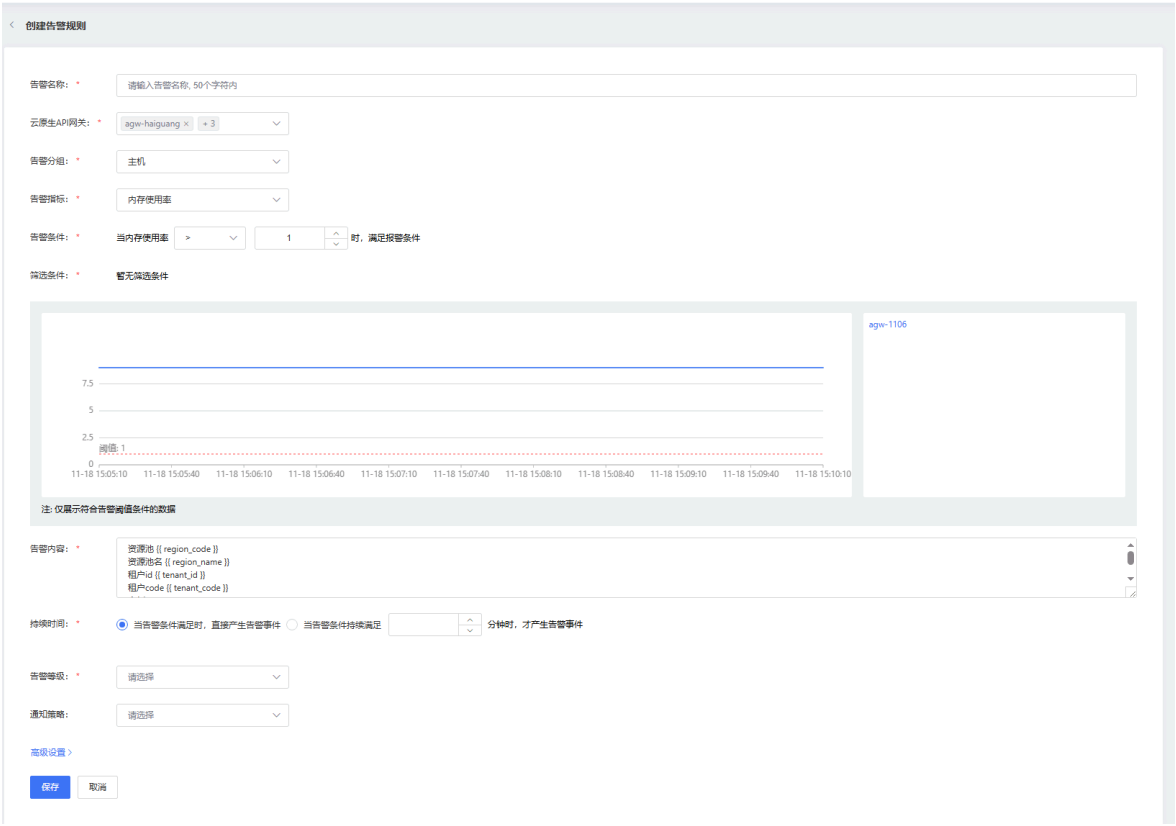
功能入口

1. 登录 云原生API网关控制台。

2. 在左侧导航栏，展开 "告警管理"菜单，点击“告警规则”。

展示当前租户下所有告警规则信息，支持基础的增删改查、起停、查看告警事件历史操作。

创建/编辑告警规则



- 告警名称：您可自定义告警名称。
- 云原生API网关：展示当前租户下所有可用实例，支持多选。
- 告警详情：包含告警分组、告警指标、告警条件、筛选条件、告警内容。
- 持续时间 ：支持设置延迟告警。

- 告警等级：一般、次要、重要、等级。
- 通知策略：可以选择通知策略菜单里设置的策略。
- 标签：自定义标签，用于筛选。

启动/停止

对于暂时不用的告警策略，您可以操作停止。

告警事件历史

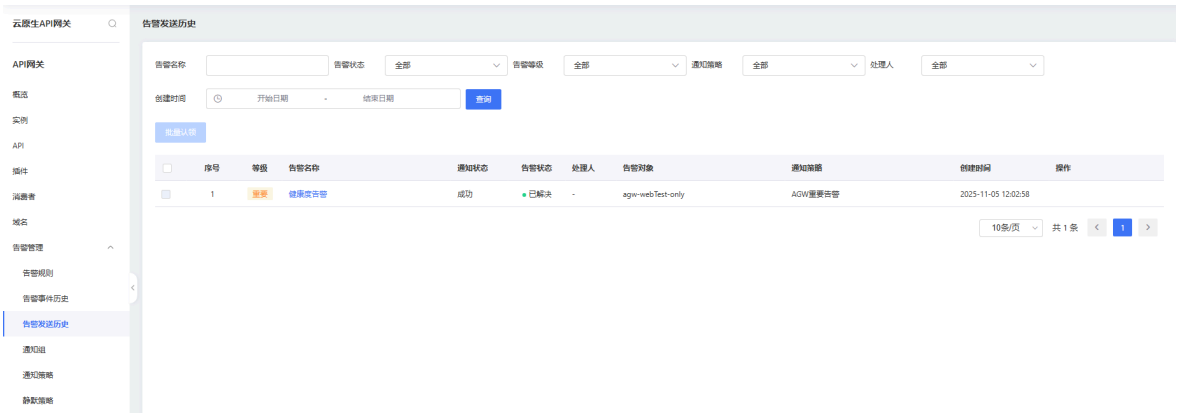
点击跳转告警事件历史菜单，显示该告警规则触发的实际告警记录。

查看告警发送历史

功能入口

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，展开 "告警管理"菜单，点击“告警发送历史”。

功能说明



展示当前租户下所有告警发送信息。

条件筛选

支持对告警名称、告警状态、告警等级、通知策略和创建时间进行筛选

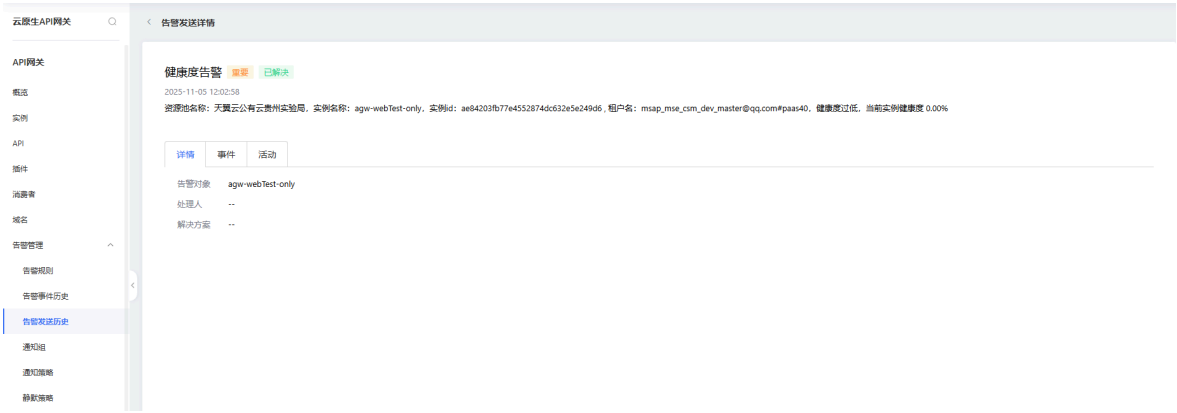
- 告警名称：显示告警规则名称。
- 告警状态：显示事件当前状态是待认领、已解决、处理中。
- 告警等级：显示告警的重要层级分布是一般、次要、重要、紧急。
- 通知策略：告警对应的通知策略。
- 处理人：告警的最新解决/认领人。
- 创建时间：告警产生的时间。

支持认领、解决、指定处理人操作

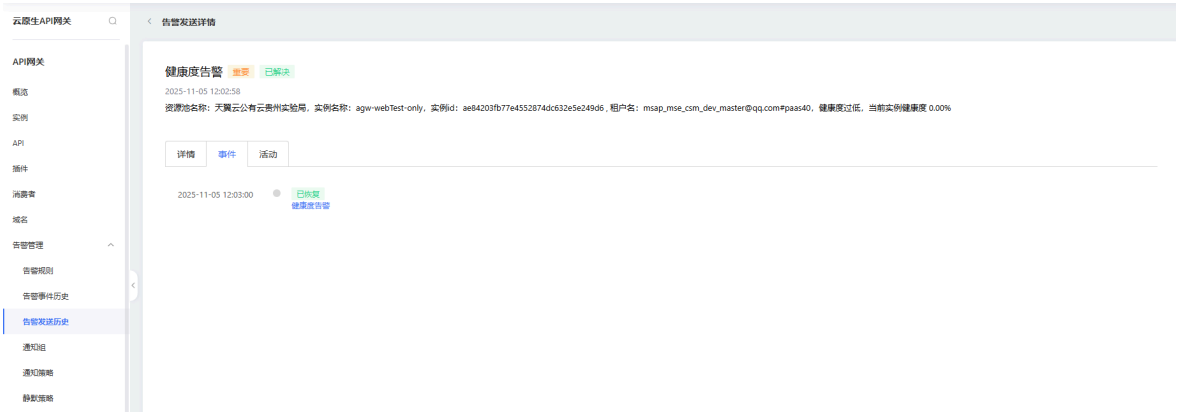
- 认领：当告警处于待认领状态时，可主动领取当前告警。
- 解决：当告警处于待认领/处理中状态时，点击解决可更新告警状态为已解决。
- 指定处理人：指定他人处理该告警。

支持查看告警详情

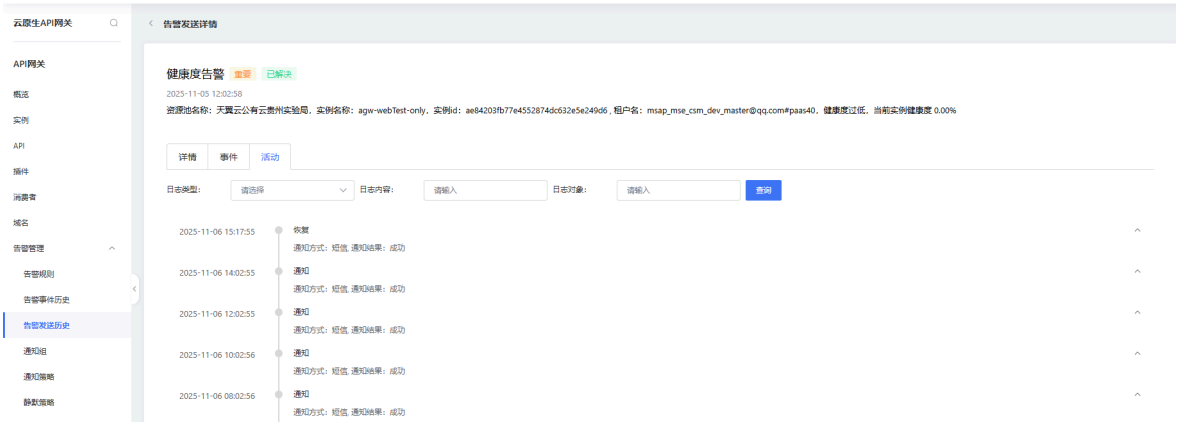
- 详情：显示告警发送的基础信息如告警对象、处理人、解决方案。



- 事件：显示触发告警的事件名称、状态和触发时间，点击可查看详情。



- 活动：显示包括认领、取消认领、指派处理人、告警关闭等在内的各种活动信息，支持筛选。

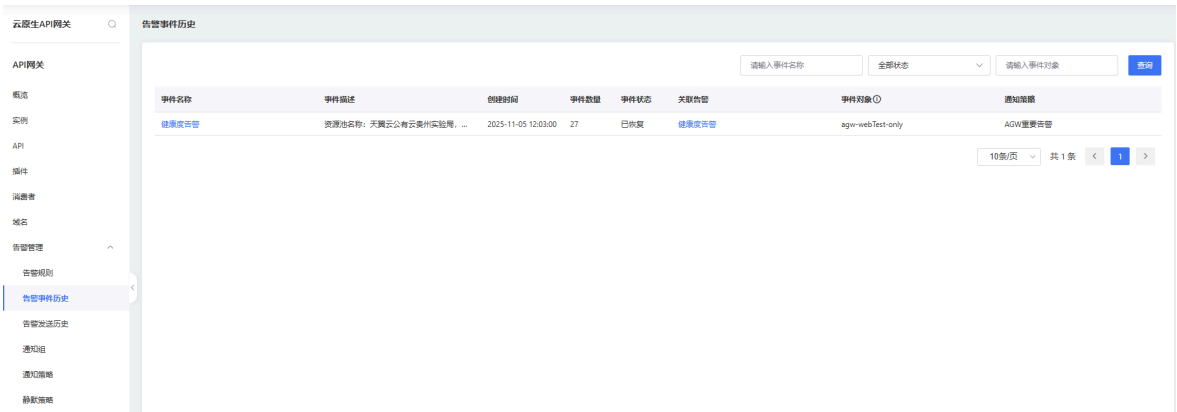


查看告警事件历史

功能入口

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，展开 "告警管理"菜单，点击“告警事件历史”。

功能说明



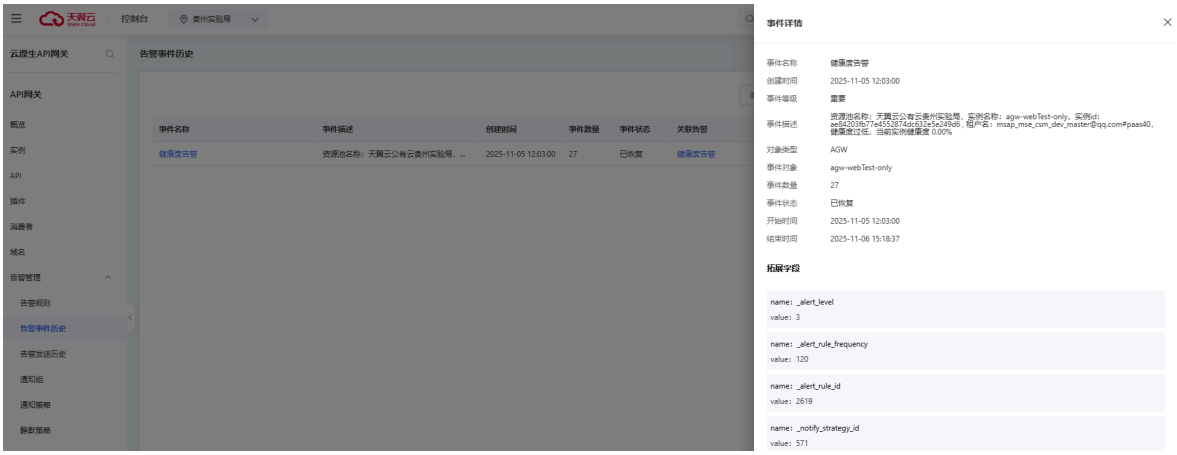
展示当前租户下所有告警事件信息。

条件筛序

支持对事件名称、事件状态、事件对象和对象类型进行筛选

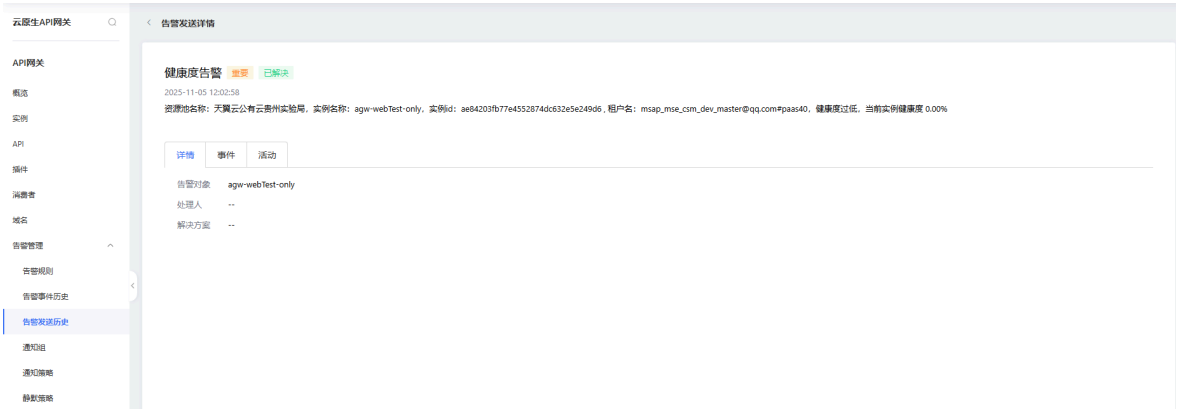
- 事件名称 ：显示告警规则名称。
- 事件状态 ：显示事件当前状态是告警中、已恢复、静默。
- 事件对象 ：监控对象，比如应用名称、集群名称等。
- 对象类型 ：告警事件对象的类型。

支持查看事件详情

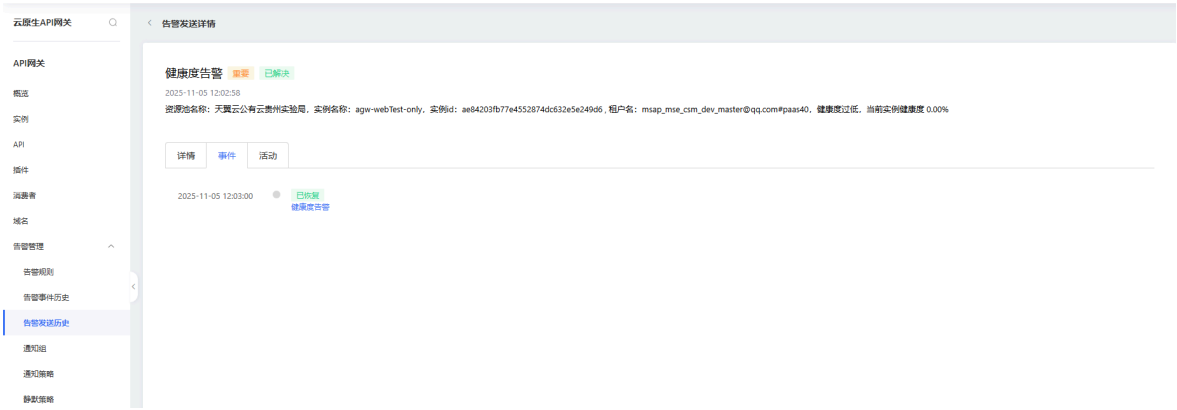


支持查看告警详情

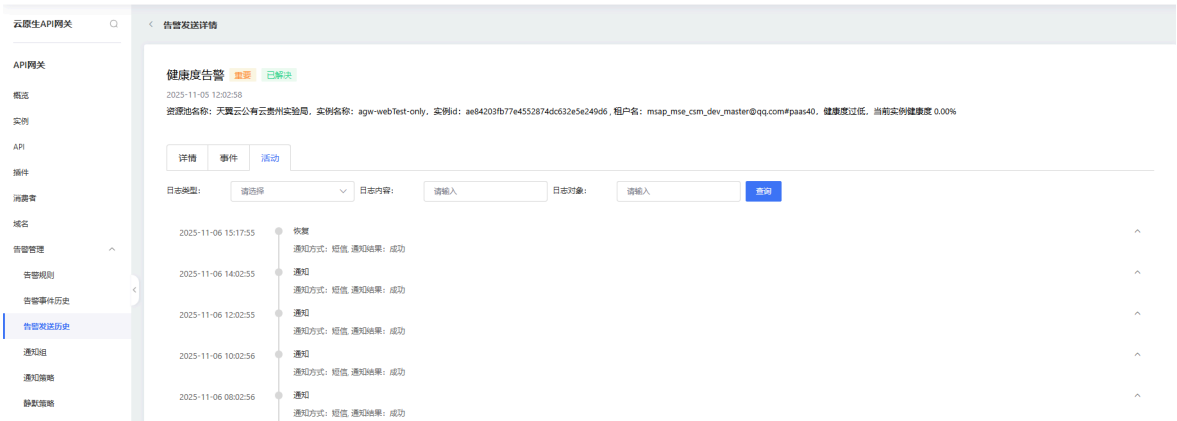
- 详情：显示告警发送的基础信息如告警对象、处理人、解决方案。



- 事件：显示触发告警的事件名称、状态和触发时间，点击可查看详情。



- 活动：显示包括认领、取消认领、指派处理人、告警关闭等在内的各种活动信息，支持筛选。



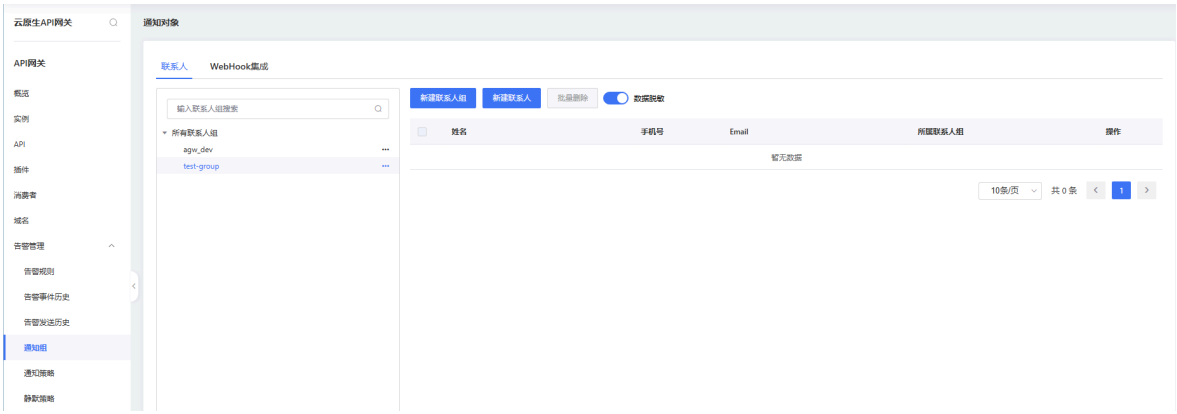
通知组

功能入口

选择目标资源池，并登录APM组件控制台。

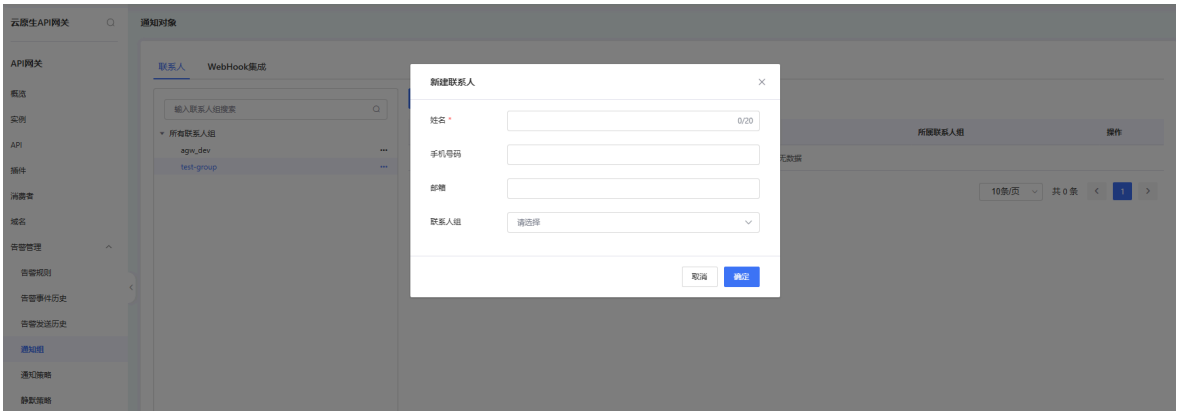
在左侧导航栏中选择「告警管理」-「通知组」。

功能说明



联系人（短信/邮箱）

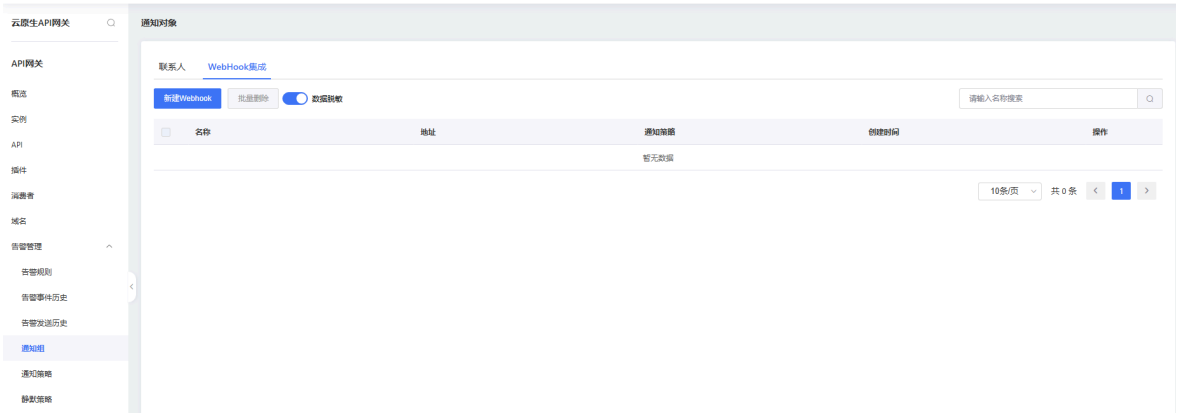
如需要通过短信/邮箱进行告警，可在此处维护联系人信息。支持对联系人信息进行增删改查，需要录入基础信息

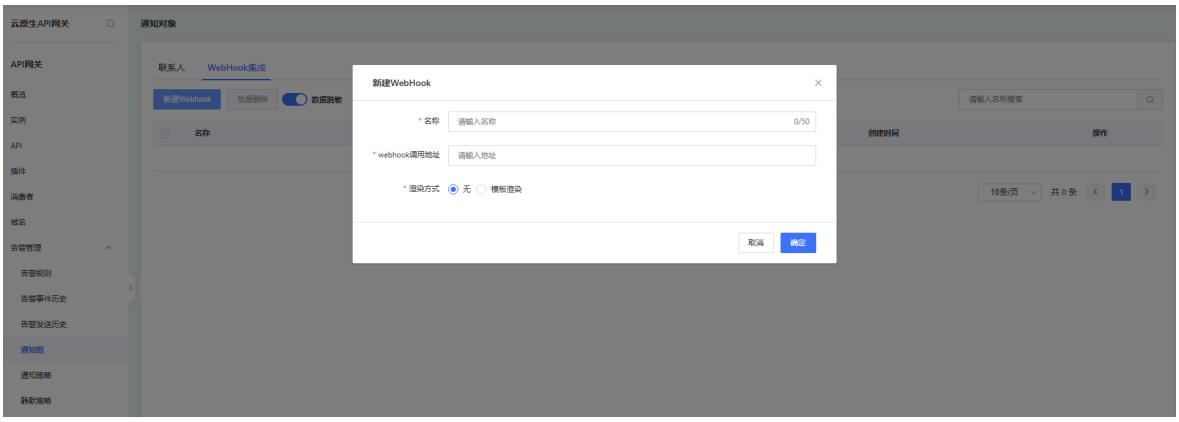


同时也支持对联系人进行分组，对联系人组进行增删改查。

WebHook集成

支持以WebHook的方式对第三方通知对象（钉钉、企业微信、飞书等）发送告警信息。支持增删改查WebHook信息。



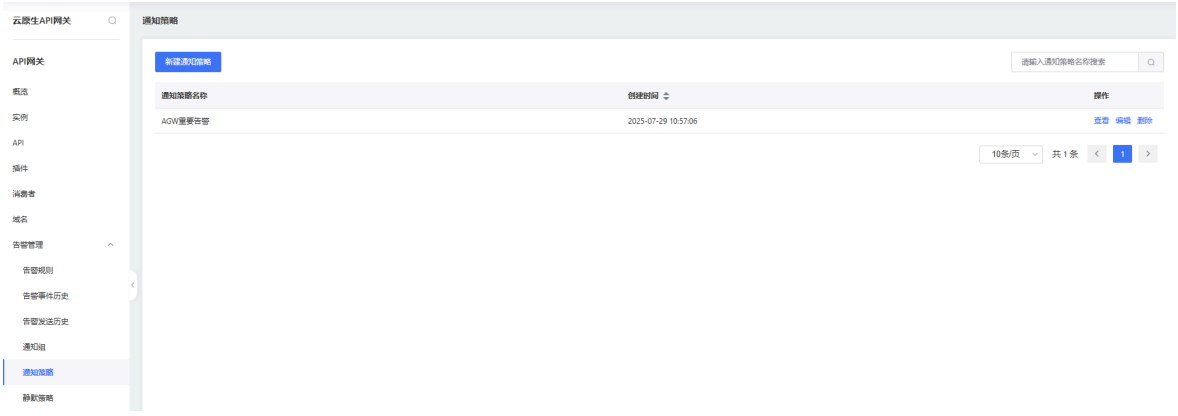


通知策略

功能入口

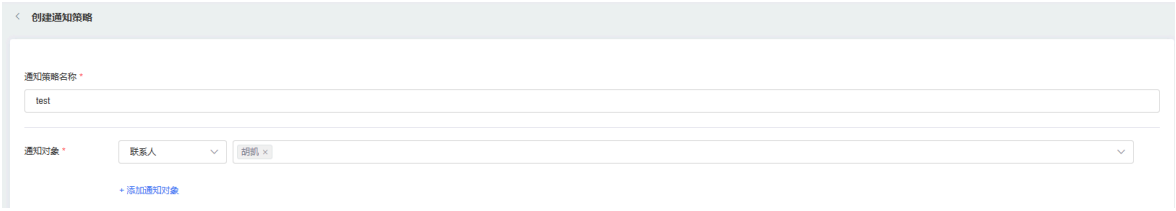
1. 登录 云原生API网关控制台。
2. 在左侧导航栏，展开 "告警管理"菜单，点击“通知策略”。

功能说明

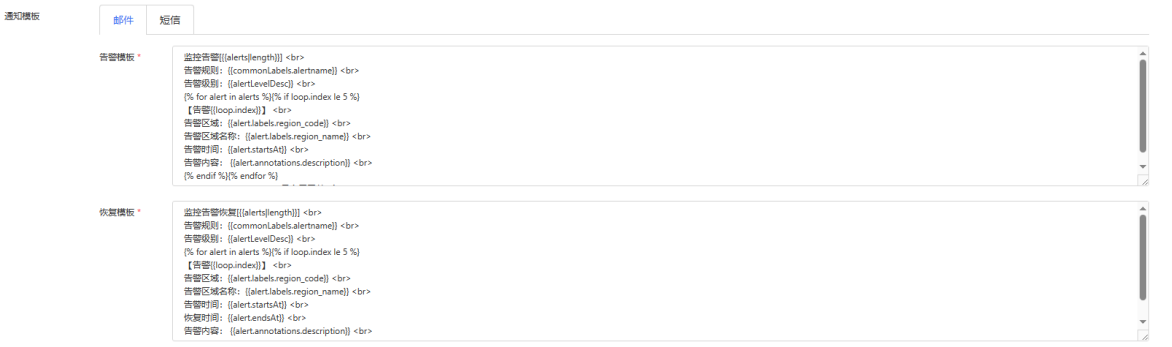


支持增删改查告警通知策略信息。

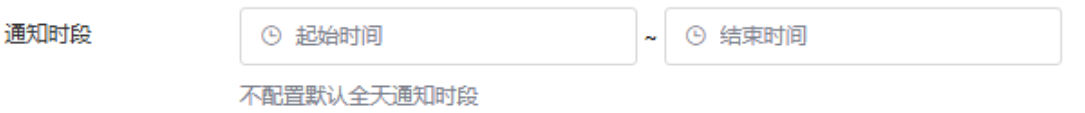
- 通知对象：可从通知组进行选择。



- 通知模板：可跟进不同通知渠道（邮件、短信）设置不同的通知模板。



- 通知时段：可以设置通知时段，默认是告警触发时通知。

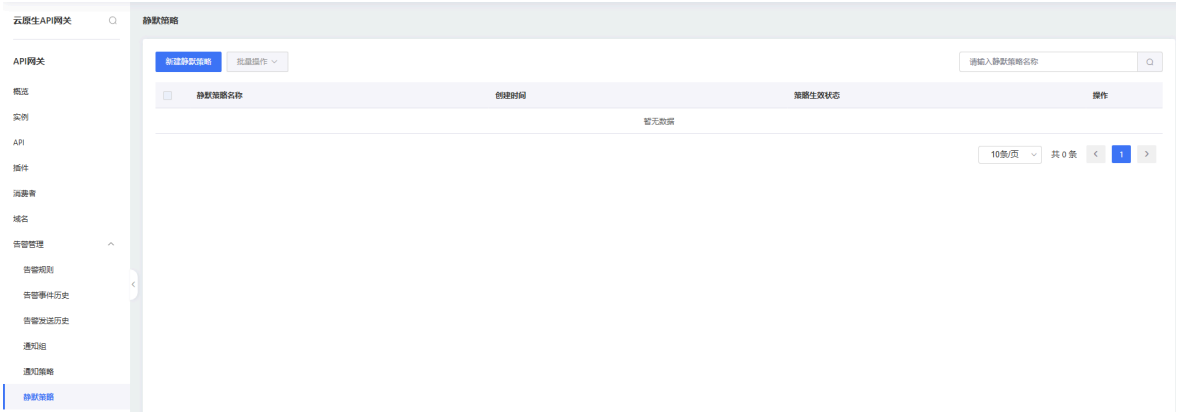


静默策略

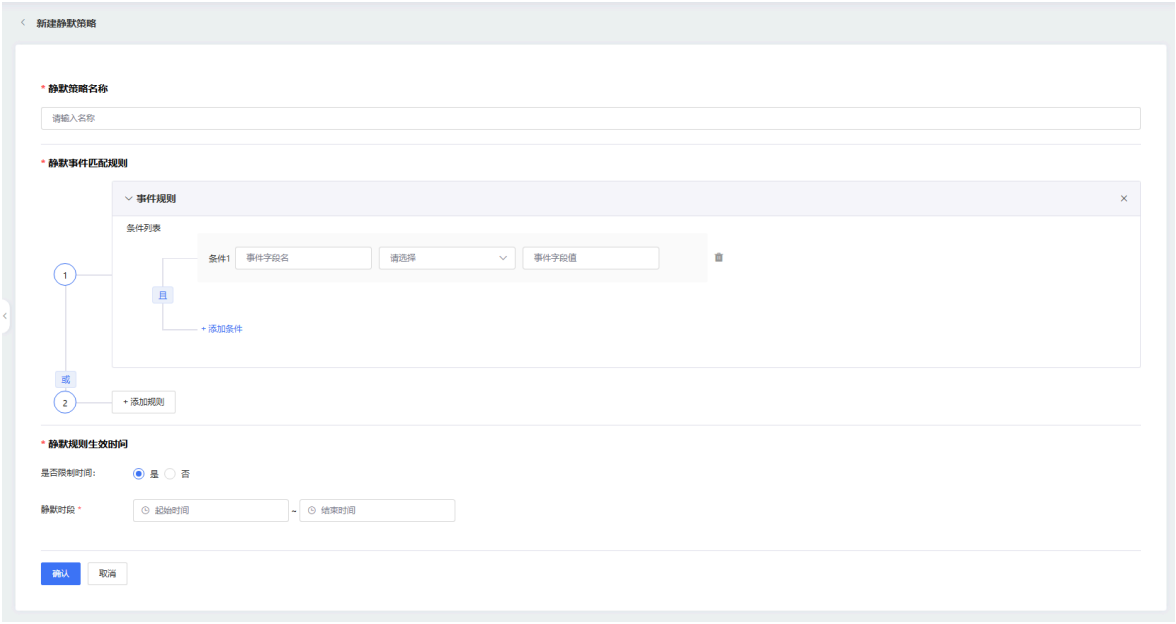
功能入口

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，展开 "告警管理"菜单，点击“静默策略”。

功能说明



支持增删改查静默策略。



- 事件规则：支持通过且或关系组合创建事件规则，使得在满足当前事件规则时，触发静默策略。
- 静默时段：设置静默规则的生效时间段。

AI网关

实例管理

创建网关实例

操作步骤

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，单击新建实例。
3. 跳转至订购页，选择相关配置（参见下方配置项说明），然后点击下一步。
4. 跳转至配置总览页，确认配置信息，点击提交订单。
5. 跳转至支付页，完成费用支付。
6. 返回AI网关控制台，左侧导航栏选择AI网关-实例，刷新列表查看创建的网关信息和状态。
7. 实例创建大约需要5~10分钟，当网关信息和创建时一致，且状态为运行中，则表示网关创建成功。

实例配置说明

配置项	描述
计费模式	支持包年包月和按需计费方式，费用说明请参照购买指南->计费说明
购买时长	可以根据实际需求进行选择，支持1个月、2个月、3个月、4个月、5个月、6个月、1年
自动续期	您可以选择开启自动续期，避免AI网关到期后无法使用
自动续期购买时长	开启自动续期，可以选择续期时长，支持1个月、2个月、3个月、4个月、5个月、6个月、1年
网关类型	选择AI网关
部署方式	实例节点将按单可用区、多可用区部署方式，分布在单个或者多个可用区中。多可用区部署可增强实例的容灾能力
可用区	选择单可用区部署方式时，可用区可任选其一；选择多可用区部署时，必须选择至少3个可用区
CPU架构	部署实例的主机架构
主机类型	部署实例的主机类型
网关规格	参见产品简介-产品规格介绍
虚拟私有云	选择虚拟私有云，若您还没有虚拟私有云，请参照创建虚拟私有云
所在子网	择所在子网，若您还没有所在子网，请参照创建所在子网
启用IPv6	若子网已开启IPv6访问，在此处可选择启用IPv6,未启用IPv6时，将通过IPv4访问
安全组	选择安全组，若您还没有可用安全组，请参照创建安全组。
启用负载均衡	启用后创建一个负载均衡实例绑定到网关实例，可以选择负载均衡类型和负载均衡网络类型

配置项	描述
实例名称	自定义实例名称，不可重复；实例名称长度4~40个字符，大小写字母开头，只能包含大小写字母、数字及分隔符(-)，大小写字母或数字结尾
企业项目	网关实例关联的企业项目，可以到IAM控制台创建企业项目
指标监控	启用后，可在控制台观测分析中查看系统和API的流量、成功率、延迟等监控指标，若您还没有开通应用性能监控产品，可先点击提示链接前往开通
链路追踪	可选择采集百分比启用，启用后，可在控制台观测分析中查看API请求的链路追踪信息，您可通过委托授权的方式开通此服务
云日志服务	启用后，可在控制台观测分析中查看访问日志，您可通过委托授权的方式开通此服务

查看网关详情

操作步骤

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，点击目标实例名称或实例id，进入实例详情。
3. 跳转至实例概览页，查看实例信息、接入点、可观测信息、消费者认证等。

实例信息

基本信息

查看实例ID、名称、付费类型、创建时间、更新时间等信息。

运行信息

查看实例的业务状态、运行状态、实例规格等信息。

网络信息

查看实例的可访问端口、地区、可用区、VPC、子网、安全组等信息。

实例节点

查看实例节点列表，包括VPC的IPv4、IPv6、http端口、https端口、分布可用区等信息。

接入点

可查看当前实例绑定的弹性负载均衡ELB实例列表。

可观测信息

查看实例的链路追踪、日志投递配置项。

消费者认证

查看实例授权的消费者列表。

修改网关名称

概述

网关名称是网关实例的别名，不影响业务，允许用户灵活修改。

操作步骤

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，点击目标实例名称或实例id，进入实例详情。
3. 在实例信息-基础信息中，点击名称旁边的修改按钮。
4. 在弹窗中填写新的名称。实例名称长度4~40个字符，大小写字母开头，只能包含大小写字母、数字及分隔符(-)，大小写字母或数字结尾。点击确认提交修改操作。
5. 弹窗展示任务处理情况，任务完成表示修改成功。

添加接入点

概述

目前AI网关的接入点为弹性负载均衡ELB，实现网关多节点入站负载均衡和公网访问暴露，支持绑定一个或多个ELB作为网关入口，从而将访问流量自动分发到多台网关节点，实现更高水平的应用程序容错性能。

标准版及以上规格的AI 网关均采用高可用部署，具备多节点架构。通过接入天翼云弹性负载均衡产品 ELB，可实现对多个网关节点的流量分发、故障自动剔除等能力，并同时支持 HTTP 和 HTTPS 协议。 对于有公网访问需求的业务场景，可通过为弹性负载均衡绑定 EIP，实现网关服务的公网访问能力。

绑定ELB

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，进入实例列表页。
3. 在网关列表页单击需要查看的网关实例ID或者实例名称 。
4. 在左侧导航栏，单击概览，然后单击接入点。
5. 单击绑定ELB，在绑定ELB面板中配置ELB参数，单击左下角确认按钮。
6. 弹窗展示绑定任务执行情况，刚绑定时关联状态为进行中，系统会自动刷新，直至关联状态更新为绑定成功。
7. 在网关入口区域可以查看新绑定的ELB 。

ELB配置参数说明

参数	描述
类型	支持公网和私网
ELB	选择同VPC下的ELB实例，若您还没有ELB实例，请先创建ELB实例。
HTTP端口	配置HTTP端口
HTTPS端口	配置HTTPS端口

注意

80、443、8080、8443常见敏感端口需进行备案后才可配置。

接入点列表参数说明

参数	描述
ELB ID	ELB的ID，单击ELB ID可以查看ELB详情

参数	描述
入口地址（ip）	ELB的ip地址，即网关入口的访问地址，您可以根据实际需求添加域名
HTTP端口	HTTP端口
HTTPS端口	HTTPS端口
类型	公网或私网
关联状态	网关实例与ELB的关联状态，当关联状态为绑定成功时可以正常访问
关联时间	网关实例与ELB关联的时间
操作	您可以单击解绑ELB进行解绑操作

解绑ELB

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，进入实例列表页。
3. 在网关列表页单击需要查看的网关实例ID或者实例名称 。
4. 在左侧导航栏，单击概览，然后单击接入点。
5. 单击列表项的解绑ELB操作。
6. 弹窗展示解绑任务执行情况，刚绑定时关联状态为进行中，系统会自动刷新，直至关联状态更新为解绑成功。
7. 在网关入口区域可以查看最新的ELB列表。

配置可观测信息

概述

AI网关支持采集链路数据和请求日志。本文介绍配置可观测信息的操作步骤。

配置链路追踪

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，进入实例列表页。
3. 在网关列表页单击需要查看的网关实例ID或者实例名称。
4. 在左侧导航栏，单击概览，然后单击可观测信息。
5. 单击链路追踪的编辑，随后打开启用链路追踪开关，输入链路采样百分比，然后单击保存。

配置日志投递

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，进入实例列表页。
3. 在网关列表页单击需要查看的网关实例ID或者实例名称 。
4. 在左侧导航栏，单击概览，然后单击可观测信息。
5. 单击日志投递的编辑，随后打开启用访问日志采集开关，然后单击保存。

配置实例级别消费者认证

概述

消费者配置说明参见消费者管理，实例级消费者在API级、路由/接口级消费者认证之前生效。本文介绍实例级消费者的操作步骤。

注意

1. 如果在实例上开启消费者认证，则将对该实例下的所有路由/接口都生效，请谨慎开启。
2. 尽量避免在实例级、API级、路由/接口级上使用同一种认证方式。

操作步骤

1. 登录AI网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择AI网关-实例，进入实例列表页。
3. 在网关列表页单击需要查看的网关实例ID或者实例名称 。
4. 在左侧导航栏，单击概览，然后单击消费者认证。
5. 在配置信息栏单击编辑，选择启用或停用消费者认证。若启用，可选择Key-Auth、JWT等认证方式，可选择从不同位置获取token。
6. 在消费者列表栏，可以为当前实例授权、解除授权多个消费者。

Model API管理

管理Model API

创建Model API

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"。
4. 单击左上角按钮 "创建Model API", 并在弹出的窗口中选择具体的使用场景并单击“创建”按钮。
5. 在弹出的创建的配置页中，配置如下配置项，并单击确定。

配置项	描述
API名称	自定义Model API名称
协议	选择协议，当前各个场景支持的协议： 文本生成：OpenAI 兼容 图片生成：OpenAI 兼容 文本排序（Rerank）：百炼文本排序、vLLM 向量化（Embedding）：OpenAI 兼容
路由	按需选择路由，默认全选，路由选项与协议相关联，选择协议后，自动带出路由列表
BasePath	自定义API的基本路径，转发至后端服务时默认会移除BasePath
域名	按需选择访问域名

描述	Model API的描述信息
场景	单模型服务：选择一个LLM服务，并可以按需选择透传或指定模型名称 多模型服务：可选择多个LLM服务，并支持流量比例配置
后端服务	选择后端服务
超时	设置网关请求后端服务的超时时间，单位ms，当值为0时默认超时时间为30秒

更新Model API

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"。
4. 单击目标Model API操作列的"编辑"，在编辑Model API面板中修改BasePath、域名、描述、场景、后端服务、超时字段信息，字段含义详见创建Model API操作说明。

删除Agent API

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"。
4. 单击目标Model API操作列的"删除"，弹框中输入Model API名称，并单击确定。

路由管理

创建路由

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击路由列表，然后单击创建路由，在创建路由面板填入路由配置，并单击保存按钮。

配置	说明
路由名称	自定义路由名称
路由描述	添加路由描述
路径（Path）	选择路径，支持当前LLM协议下的路由
更多匹配规则	自定义匹配规则，创建多个相同路径路由时，支持配置方法、请求头、请求参数、Cookie，来区分不同路由
场景	单模型服务：选择一个LLM服务，并可以按需选择透传或指定模型名称 多模型服务：可选择多个LLM服务，并支持流量比例配置
后端服务	选择后端服务

更新路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。

3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击路由列表，单击目标路由，进入路由详情页。
5. 单击路由详情页右上角编辑按钮，在路由编辑面板中填入编辑内容，并单击保存。

删除路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击路由列表，单击目标路由，进入路由详情页。
5. 单击路由详情页右上角删除按钮，并在弹窗中单击确认。

发布路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击路由列表，单击目标路由，进入路由详情页。
5. 单击路由详情页右上角发布按钮，并在弹窗中单击确认。

下线路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击路由列表，单击目标路由，进入路由详情页。
5. 单击路由详情页右上角下线按钮，并在弹窗中单击确认。

配置策略与插件

限流

限流是针对高并发大模型服务的使用场景设计，保障系统的可用性。当前实现为单机限流，基于滑动时间窗口实现，可以配置时间窗口大小（秒）以及在一个时间窗口内限制的请求数。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击限流，填入限流策略配置并启用，单击保存按钮。

配置	说明
时间窗口	进行限流统计的时间窗口
限制请求	时间窗口内允许的最大请求次数，超出的请求将会被拒绝

跨域

跨域支持配置来源、方法、请求头、响应头、凭证等字段的配置。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理/出站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击跨域，填入跨域策略配置并启用，单击保存按钮。

配置项	说明
允许访问的来源	作用于 Access-Control-Allow-Origin 头部，标识跨域请求的来源（协议 + 域名 + 端口），如：scheme://host:port、https://foo.ctyun.com:8080，多个值使用','分割， '*'表示所有 Origin 均允许通过
允许的方法	作用于 Access-Control-Allow-Methods 头部，表示允许的访问方法
允许的请求头部	作用于 Access-Control-Allow-Headers 头部，允许跨域访问时请求方携带哪些 CORS 规范以外的 Header，多个值使用','分割，'*'来表示所有 Header 均允许通过
允许的响应头部	作用于 Access-Control-Expose-Headers 头部，允许浏览器和js 脚本访问的响应头部
允许携带凭证	作用于 Access-Control-Allow-Credentials 头部
预检的过期时间	作用于 Access-Control-Max-Age 头部
开启状态	开启时才生效

外部认证授权

该策略支持通过第三方外部服务进行身份认证与授权。当身份认证失败时，可以实现自定义错误或者重定向到认证页面的场景。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击外部认证授权，填入外部认证授权策略配置并启用，单击保存按钮。

配置	说明
开启状态	开启时配置才生效
服务地址	设置外部认证服务的地址（例如： https://localhost:9188）
请求方法	客户端向认证服务发送请求的方法。当设置为 POST 时，会将请求体转发给认证服务
转发到认证服务的请求头	设置需要由客户端转发给认证服务的请求头。如果没有设置，则只发送如 X-Forwarded-XXX 的请求头
转发给上游服务的请求头	认证通过时，由认证服务转发给上游服务的响应头。如果不设置则不转发任何响应头
转发给客户端的请求头	认证失败时，由认证服务向客户端发送的响应头。如果不设置则不转发任何响应头

验证 ssl 证书	当开启时，验证 SSL 证书，默认开启
认证服务请求超时时间	认证服务请求超时时间，单位毫秒（ms）
长连接超时时间	长连接超时时间，单位毫秒（ms）

熔断

该策略支持给上游服务配置熔断规则，发现上游服务不可用时，快速失败，等恢复了再慢慢试探，防止整个系统雪崩。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击熔断，填入熔断策略配置并启用，单击保存按钮。

配置	说明
开启状态	开启时配置才生效
上游服务健康状态码	上游服务处于健康状态时的HTTP状态码
上游服务连续正常请求次数	上游服务触发健康状态的连续正常请求次数
上游服务不健康状态码	上游服务处于不健康状态时的 HTTP状态码
触发异常请求次数	上游服务在一定时间内触发不健康状态的异常请求次数
熔断最大持续时间	上游服务熔断的最大持续时间，以秒为单位
不健康返回错误码	当上游服务处于不健康状态时返回的 HTTP错误码
不健康返回响应体信息	当上游服务处于不健康状态时返回的 HTTP响应体信息
不健康返回响应头信息	当上游服务处于不健康状态时返回的 HTTP响应头信息。该字段仅在配置了不健康返回响应体信息时才生效

黑白名单

Model API支持通过配置 IP 黑名单和白名单的方式限制客户端访问；黑白名单不能同时开启，只有一种能生效。

云原生网关默认读取请求中的 Remote_addr 字段值作为客户端 IP（即网络层 IP）；如果您的客户端访问出口存在七层代理，此时 Remote_addr 字段值为出口代理地址，可通过开启从 xff 头部获取 IP 配置选项，从 X-Forwarded-For 字段中获取客户端真实 IP。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击黑白名单，填入黑白名单策略配置并启用，单击保存按钮。

配置	说明
是否从 xff 头部获取IP	是否从 X-Forwarded-For 字段中获取客户端真实IP，若不开启，则默认从请求中的remote-addr中获取IP
黑名单	支持单/多个IP地址或类似 10.10.10.0/24的CIDR范围配置，每个条目一行，以回车分隔
白名单	支持单/多个IP地址或类似 10.10.10.0/24的CIDR范围配置，每个条目一行，以回车分隔
开启状态	开启时配置才生效

防重放

防止攻击者重复发送已截获的合法请求，避免重复操作或数据异常。开启后，请求头必须包含 x-ca-timestamp 和 x-ca-nonce 参数。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击防重放，填入防重放策略配置并启用，单击保存按钮。

配置	说明
时间窗口	时间窗口内不可重复请求，请求时间超过时间窗口为无效请求
开启状态	开启时配置才生效

Header设置

Header设置策略允许用户新增/修改/删除响应头。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到出站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击Header设置，填入Header设置策略配置并启用，单击保存按钮。

配置	说明
开启状态	开启时策略才生效
Header 类型	网关与后端交互时支持对请求和应答的头部做修改。Model API仅支持对响应头做修改

操作类型	支持新增、修改、删除操作 新增：若 header key 已存在，则在末尾追加 header value；否则新增 修改：若 header key 不存在，则新增 header kv；否则覆盖已有 header value 值 删除：若 header key 存在，则删除；否则忽略该header key
Header Key	指定Header Key
Header Value	新增/修改操作时指定Header Value

ProxyCookie设置

该策略支持对上游响应 Set-Cookie 头部重写，当前支持对 Set-Cookie 头部里的 Domain 和 Path 进行重写。

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到出站处理卡片，单击启用策略/插件，在启用策略/插件面板中单击ProxyCookie设置，填入ProxyCookie设置策略配置并启用，单击保存按钮。

配置	说明
proxy_cookie_domain 匹配规则	匹配上游应答 Set-Cookie 头部的 Domain 字段，支持正则匹配
proxy_cookie_domain 替换值	如果匹配，Set-Cookie 头部 Domain 字段将被替换成该配置值
proxy_cookie_path 匹配规则	匹配上游应答 Set-Cookie 头部的 Path 字段，支持正则匹配
proxy_cookie_path 替换值	如果匹配，Set-Cookie 头部 Path 字段将被替换成该配置值

添加插件

操作步骤：

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Model API"，然后单击目标API名称进入API详情页面。
4. 单击策略与插件，找到入站处理/出站处理卡片，单击启用策略/插件。
5. 在启用策略/插件面板中切换至添加插件tab页，找到目标插件卡片。

   # 如果插件未安装：单击插件卡片，选择需要安装的实例，并单击安装按钮。

   # 如果插件已安装：单击插件卡片，单击插件卡片，在插件编辑面板中选中启用状态，并单击保存。

消费者认证

配置Model API消费者认证

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。

3. 在左侧导航栏，选择"Model API"，单击Model API名称，进入Model API详情页，然后单击"消费者认证"页签。
4. 单击配置信息右侧的"编辑"，进行如下参数配置：

> 注意
>
> 开启消费者认证后，若没有配置授权关系，将无法访问当前API。

- 启用状态：开启或关闭消费者授权开关，默认关闭。
- 认证方式：目前支持API Key和JWT两种方式。
  - API Key：客户端访问时，需将凭证以指定的方式添加至请求中，网关收到请求后会验证API Key的合法性及权限。
  - JSON Web Token (JWT)： 用于在客户端和服务端之间以JSON对象的形式安全地传输信息，该信息可以被验证和信任。

5. 在消费者区域单击授权，选择消费者，单击"添加"。

Model API可观测

操作步骤

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"Model API"，单击Model API名称，进入Model API详情页。
4. 选择监控页签，可查看该MCP服务的QPS、请求成功率和平均延迟等监控指标，右上角可调整时间间隔。
   - QPS：每秒Model API请求和响应的数量统计。
   - 请求成功率：Model API请求的成功率。
   - 平均延迟：Model API请求的平均延迟时间。

MCP管理

托管MCP服务

功能介绍

MCP服务管理支持直接代理模式，适用于原生支持MCP协议的服务。该模式能够实现高效的流式通信和上下文保持，特别适合高并发、长连接的场景，例如 AI 推理、多模型协同等。

创建托管MCP服务

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"。
4. 单击按钮 "创建MCP服务"，AI网关目前提供了MCP服务直接代理类型。请配置以下基本信息，并点击"确定"。

配置项	描述
名称	自定义MCP服务名称
协议	目前仅支持MCP服务直接代理
描述	MCP服务的描述信息

配置项	描述
后端服务-服务名称	在服务列表的下拉框中选择目标MCP服务
后端服务-服务协议	当前为固定值MCP
后端服务-MCP Transport	支持SSE和Streamable HTTP两种传输协议，请根据目标服务的实际传输协议来选择
后端服务-路径	路径是实际的后端mcp服务的访问path。比如后端mcp server访问端点是xxx.com/sse，路径就填/sse；如果是xxx.com/test/sse，路径就填/test/sse
MCP接入点-域名	选择访问MCP服务使用的域名，支持多个域名
MCP接入点-路径	MCP接入点的路径是通过/mcp-servers、/MCP服务名称以及实际的后端MCP Server的访问path拼接成的。创建完成后，可以在MCP服务详情页查看完整的访问地址

编辑MCP服务

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，单击目标MCP服务操作列的"编辑"。或者进入MCP服务详情页，单击右上角的"编辑"。
4. 在编辑MCP服务面板中修改描述、服务名称、路径和域名，然后单击"确定"。

发布MCP服务

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入目标MCP服务详情页。
4. 当MCP服务状态为未发布或已发布(有修改)时，可以单击右上角的"发布"。

下线MCP服务

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入目标MCP服务详情页。
4. 当MCP服务状态为已发布或已发布(有修改)，可以单击右上角的"下线"。

删除MCP服务

> 说明
>
> MCP服务需先下线才能删除。

1. 登录 云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，单击目标MCP服务操作列的"删除"。

消费者认证

配置MCP服务级别消费者认证

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。

2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入MCP服务详情页，切换到"消费者认证"页签。
4. 单击配置信息右侧的"编辑"，进行如下参数配置：

> 注意
>
> 开启消费者认证后，若没有配置授权关系，将无法访问当前API。

   - 启用状态：开启或关闭消费者授权开关，默认关闭。
   - 认证方式：目前支持API Key和JWT两种方式。
     - API Key：客户端访问时，需将凭证以指定的方式添加至请求中，网关收到请求后会验证API Key的合法性及权限。
     - JSON Web Token (JWT)： 用于在客户端和服务端之间以JSON对象的形式安全地传输信息，该信息可以被验证和信任。
5. 在消费者区域单击授权，选择消费者，单击"添加"。

调试MCP服务

通过控制台调试MCP服务

> 说明
>
> 网关实例需要绑定公网ELB才能支持MCP服务调试。

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入MCP服务详情页。
4. 当MCP服务状态为已发布和已发布(有修改)时，可以单击右上角"调试"，进入调试面板。
5. 选择连接地址，单击"连接"。控制台会自动跟MCP Server建立连接。

> 说明
>
> - 如果使用自定义域名，需要确保此域名在公网可解析，且此域名解析的地址公网可达。
> - 不支持内网域名和内网地址调试。

6. 如果开启了消费者认证，需要在认证校验栏选择授权消费者。
7. 成功建立连接后，会自动获取工具列表，选择需要使用的工具，填入工具参数即可在线调试。

配置策略和插件

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入MCP服务详情页，切换到"策略与插件"页签。
4. 在启用策略/插件面板中，选择策略或插件进行配置。

策略配置

限流

当前实现为单机限流，基于时间窗口实现，可以配置时间窗口大小（秒）以及在一个时间窗口内限制的请求数。

配置	说明
时间窗口	进行限流统计的时间窗口
限制请求	时间窗口内允许的最大请求次数，超出的请求将会被拒绝

Header设置

Header配置支持对请求和响应的头部做修改。

配置	说明
开启状态	开启时策略才生效
Header类型	网关与后端交互时支持对请求和应答的头部做修改
操作类型	支持新增、修改、删除操作新增：若header key已存在，则在末尾追加header value；否则新增修改：若header key不存在，则新增header kv；否则覆盖已有header value值删除：若header key存在，则删除；否则忽略该header key
Header Key	头部Key
Header Value	头部Value

跨域

AI网关支持跨域资源共享（CORS）。

CORS配置说明如下：

配置	说明
允许访问的来源	作用于Access-Control-Allow-Origin头部，格式如：scheme://host:port，比如:https://foo.ctyun.com:8080，多个值使用','分割，'*'表示所有Origin均允许通过
允许的方法	作用于Access-Control-Allow-Methods头部，表示允许的访问方法
允许的请求头部	作用于Access-Control-Allow-Headers头部，允许跨域访问时请求方携带哪些CORS规范以外的Header，多个值使用','分割，'*'来表示所有Header均允许通过
允许的响应头部	作用于Access-Control-Expose-Headers头部，允许浏览器和js脚本访问的响应头部
允许携带凭证	作用于Access-Control-Allow-Credentials头部
预检的过期时间	作用于Access-Control-Max-Age头部
开启状态	开启时才生效

Query参数设置

该配置支持对请求参数进行修改。

配置	说明
开启状态	开启时策略才生效
操作类型	支持新增、修改、删除操作新增：若请求参数已存在，则在末尾追加；否则新增修改：若请求参数不存在，则新增该参数；否则覆盖已有参数值删除：若请求参数存在，则删除；否则忽略该参数
参数名	请求参数key
参数值	请求参数Value

熔断设置

该配置支持在触发上游服务不健康状态时进行熔断，从而保护上游业务服务。

配置	说明
开启状态	开启时配置才生效
上游服务健康状态码	上游服务处于健康状态时的HTTP状态码
上游服务连续正常请求次数	上游服务触发健康状态的连续正常请求次数
上游服务不健康状态码	上游服务处于不健康状态时的HTTP状态码
触发异常请求次数	上游服务在一定时间内触发不健康状态的异常请求次数
熔断最大持续时间	上游服务熔断的最大持续时间，以秒为单位
不健康返回错误码	当上游服务处于不健康状态时返回的HTTP错误码
不健康返回响应体信息	当上游服务处于不健康状态时返回的HTTP响应体信息
不健康返回响应头信息	当上游服务处于不健康状态时返回的HTTP响应头信息。该字段仅在配置了不健康返回响应体信息时才生效

黑白名单

AI网关支持通过配置IP黑名单和白名单的方式限制客户端访问网关；黑白名单不能同时开启，同时只有一种能生效。

AI网关默认读取请求中的Remote_addr字段值作为客户端IP（即网络层IP）；如果您的客户端访问出口存在七层代理，此时Remote_addr字段值为出口代理地址，可通过开启从xff头部获取IP配置选项，从X-Forwarded-For字段中获取客户端真实IP。

配置	说明
是否从xff头部获取IP	是否从X-Forwarded-For字段中获取客户端真实IP
黑名单	黑名单IP配置
白名单	白名单IP配置

插件配置

1. 单击"添加插件"页签。
2. 在快捷导航处，选择要安装的插件类型或者搜索插件名称，单击插件卡片：
   - 如果插件未安装，在安装插件的弹出框中单击安装，在启用插件的弹框中配置插件规则，并选择启用状态。
   - 如果插件已安装，在启用插件的弹框中，配置插件规则，并选择启用状态。
3. 单击确定，返回"策略与插件"页面，可以查看API的插件启用状态。

MCP可观测

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例" 菜单，进入实例列表页，选择目标实例进入详情页。
3. 在左侧导航栏，选择"MCP管理-MCP服务"，进入MCP服务详情页。
4. 选择监控页签，可查看该MCP服务的QPS、请求成功率和平均延迟等监控指标，右上角可调整时间间隔。
   - QPS：每秒MCP服务请求和响应的数量统计。
   - 请求成功率：MCP服务请求的成功率。
   - 平均延迟：MCP服务请求的平均延迟时间。

Agent API管理

管理Agent API

创建Agent API

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"。
4. 单击左上角按钮 "创建Agent API"， 配置以下基本信息，并单击确定。

配置项	描述
API名称	自定义Agent API名称
域名	选择您计划创建Agent API的域名
BasePath	自定义API的基本路径，BasePath会转发给后端服务
协议	目前支持阿里云百炼和自定义协议 • 阿里云百炼应用：阿里云百炼推出的大模型应用，智能体、工作流与智能体编排，可扩展了大模型的应用范围 • 自定义：适用于您自定义的其他Agent服务
描述	Agent API的描述信息

编辑Agent API

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。

2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例概览。
3. 在左侧导航栏，选择"Agent API"。
4. 单击目标Agent API操作列的"编辑"，在编辑Agent API面板中修改域名、BasePath和描述，然后单击确定。

删除Agent API

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例概览。
3. 在左侧导航栏，选择"Agent API"。
4. 单击目标Agent API操作列的"删除"，弹框中单击 确定。

路由管理

创建路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API页面。
4. 路由列表TAB页单击 "创建路由"，在创建路由面板，进行如下配置，并单击保存。

配置项	描述
路由名称	自定义路由名称
路由描述	路由的描述信息

路径	设置匹配路由请求的Path参数
	• 相同匹配规则时Path越长优先级越高
	• 不同匹配规则时等于 **>** 前缀是
	  • 等于：即完全匹配。例如，Path等于/user。
	  • 前缀是：以前缀作为匹配条件。例如，Path以/user开头。

更多匹配规则	如需创建多个相同路径的路由，需要对方法（Method）、请求头（Header）、请求参数（Query）和Cookie进行配置，以区分不同路由
场景	支持Agent服务场景和基础场景 • Agent服务场景：只能选择Agent服务类型的服务。支持单Agent服务和多Agent服务 • 基础场景：可选择除Agent服务类型和LLM服务类型以外的其他类型服务。支持单服务和多服务
后端服务	在对应场景下的服务列表中选择服务

编辑路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API页面。
4. 在路由列表页面选择目标路由，单击页面右侧 "编辑"按钮，完成基本信息和后端服务配置修改。单击保存即可完成路由修改。

发布路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。

2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API页面。
4. 在路由列表页面选择目标未发布路由，单击页面右侧"发布"按钮，弹框中单击确定，即可对已创建的路由进行发布。

下线路由

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API页面。
4. 在路由列表页面选择目标已发布路由，单击页面右侧"下线"按钮，弹框中单击确定，即可完成路由下线。

删除路由

说明

路由需先下线再删除。

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API页面。
4. 在路由列表页面选择目标路由，单击页面右侧"删除"按钮，弹框中单击确定，即可完成路由删除。

消费者认证

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API概览。
4. 选择"消费者认证"页签，单击配置信息右侧的编辑，进行如下参数配置：

注意

开启消费者认证后，若没有配置授权关系，将无法访问当前API。

- 启用状态：开启或关闭消费者授权开关，默认关闭。
- 认证方式：目前支持API Key和JWT两种方式。
  - API Key：客户端访问时，需将凭证以指定的方式添加至请求中，网关收到请求后会验证API Key的合法性及权限。
  - JSON Web Token (JWT)： 用于在客户端和服务端之间以JSON对象的形式安全地传输信息，该信息可以被验证和信任。

5. 在消费者区域单击授权，选择消费者，单击 添加。

配置策略与插件

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例概览。

3. 在左侧导航栏，选择"Agent API"，进入目标API概览。
4. 选择策略与插件页签，单击 "启用策略/插件"。
5. 在启用策略/插件面板中，选择策略或插件进行配置。

策略配置

限流

当前实现为单机限流，基于时间窗口实现，可以配置时间窗口大小（秒）以及在一个时间窗口内限制的请求数。

配置	说明
时间窗口	进行限流统计的时间窗口
限制请求	时间窗口内允许的最大请求次数，超出的请求将会被拒绝

跨域设置

云原生网关支持路由级别的跨域资源共享（CORS）。

CORS配置说明如下：

配置项	说明
允许访问的来源	作用于Access-Control-Allow-Origin头部，格式如：scheme://host:port，比如:https://foo.ctyun.com:8080，多个值使用','分割，'*'表示所有Origin均允许通过
允许的方法	作用于Access-Control-Allow-Methods头部，表示允许的访问方法
允许的请求头部	作用于Access-Control-Allow-Headers头部，允许跨域访问时请求方携带哪些CORS规范以外的Header，多个值使用','分割，'*'来表示所有Header均允许通过
允许的响应头部	作用于Access-Control-Expose-Headers头部，允许浏览器和js脚本访问的响应头部
允许携带凭证	作用于Access-Control-Allow-Credentials头部
预检的过期时间	作用于Access-Control-Max-Age头部
开启状态	开启时才生效

外部认证授权

该配置支持通过第三方外部服务进行身份认证与授权。当身份认证失败时，可以实现自定义错误或者重定向到认证页面的场景。

配置项说明：

配置	说明
开启状态	开启时配置才生效
服务地址	设置外部认证服务的地址（例如：https://localhost:9188）
请求方法	客户端向认证服务发送请求的方法。当设置为POST时，会将请求体转发给认证服务

配置	说明
转发到认证服务的请求头	设置需要由客户端转发给认证服务的请求头。如果没有设置，则只发送如X-Forwarded-XXX的请求头
转发给上游服务的请求头	认证通过时，由认证服务转发给上游服务的响应头。如果不设置则不转发任何响应头
转发给客户端的请求头	认证失败时，由认证服务向客户端发送的响应头。如果不设置则不转发任何响应头
验证ssl证书	当开启时，验证SSL证书，默认开启
认证服务请求超时时间	认证服务请求超时时间
长连接超时时间	长连接超时时间

熔断设置

该配置支持在触发上游服务不健康状态时进行熔断，从而保护上游业务服务。

配置项说明：

配置	说明
开启状态	开启时配置才生效
上游服务健康状态码	上游服务处于健康状态时的HTTP状态码
上游服务连续正常请求次数	上游服务触发健康状态的连续正常请求次数
上游服务不健康状态码	上游服务处于不健康状态时的HTTP状态码
触发异常请求次数	上游服务在一定时间内触发不健康状态的异常请求次数
熔断最大持续时间	上游服务熔断的最大持续时间，以秒为单位
不健康返回错误码	当上游服务处于不健康状态时返回的HTTP错误码
不健康返回响应体信息	当上游服务处于不健康状态时返回的HTTP响应体信息
不健康返回响应头信息	当上游服务处于不健康状态时返回的HTTP响应头信息，该字段仅在配置了不健康返回响应体信息时才生效

黑白名单

云原生网关支持通过配置IP黑名单和白名单的方式限制客户端访问网关；黑白名单不能同时开启，同时只有一种能生效。

云原生网关默认读取请求中的Remote_addr字段值作为客户端IP（即网络层IP）；如果您的客户端访问出口存在七层代理，此时Remote_addr字段值为出口代理地址，可通过开启从xff头部获取IP配置选项，从X-Forwarded-For字段中获取客户端真实IP。

配置	说明
是否从xff头部获取IP	是否从X-Forwarded-For字段中获取客户端真实IP
黑名单	黑名单IP配置
白名单	白名单IP配置

防重放

防止攻击者重复发送已截获的合法请求，避免重复操作或数据异常。

开启后，请求头必须包含x-ca-timestamp和x-ca-nonce参数。

配置	说明
时间窗口	时间窗口内不可重复请求，请求时间超过时间窗口为无效请求

插件配置

1. 单击添加插件页签。
2. 在快捷导航处，选择要安装的插件类型或者搜索插件名称，单击插件卡片：
   - 如果插件未安装，在安装插件的弹出框中单击安装，在启用插件的弹框中配置插件规则，并选择启用状态。
   - 如果插件已安装，在启用插件的弹框中，配置插件规则，并选择启用状态。
3. 单击确定，返回"策略与插件"页面，可以查看API的插件启用状态。

Agent API可观测

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择资源池。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例概览。
3. 在左侧导航栏，选择"Agent API"，进入目标API概览。
4. 选择监控页签，可查看该API的QPS、请求成功率和平均延迟等监控指标，右上角可调整时间间隔。

- QPS：每秒Agent API请求和响应的数量统计。
- 请求成功率：Agent API请求的成功率。
- 平均延迟：Agent API请求的平均延迟时间。

服务管理

服务

创建服务

概述

您需要完成将已有的服务创建为路由备选服务，网关在进行转发时会根据路由规则将请求转发至备选服务。

创建服务

AI网关支持创建LLM服务、Agent服务、容器服务、MSE Nacos、固定地址、函数计算和DNS域名。

LLM服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。

3. 配置基本信息。

- 在弹窗中先选择服务来源为“LLM服务”，随后编辑其余配置。
- 服务名称： 自定义服务名称。
- 大模型供应商： 支持息壤、DeepSeek、OpenAI兼容（OpenAI Compatible）、百炼。

注意：本AI服务背后的大模型技术由第三方提供，并非由我们直接运营。在使用前，请务必根据自身需求独立判断该服务的适用性与可靠性，并严格遵守所有适用的法律法规及本平台的服务条款。如因您违反上述要求而引发任何问题或损失，我们将不承担任何责任。

服务地址(base_url)：大模型服务的BaseURL。

- API-KEY：访问大模型需要的API-KEY凭证。API-KEY的获取请咨询对应服务供应商。

4. 配置完成后单击确定，完成创建。

Agent服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。
3. 配置基本信息。

在弹窗中先选择服务来源为“Agent服务”，随后编辑其余配置。

- 服务名称： 自定义服务名称。
- 服务供应商： 支持选择百炼、自定义。

注意：本Agent服务背后的智能体能力由第三方提供，并非由我们直接运营。在使用前，请务必根据自身需求独立判断该服务的适用性与可靠性，并严格遵守所有适用的法律法规及本平台的服务条款。如因您违反上述要求而引发任何问题或损失，我们将不承担任何责任。

- 服务地址(base_url)：Agent服务的BaseURL。
- API-KEY：访问Agent服务需要的API-KEY凭证。API-KEY的获取请咨询对应服务供应商。

4. 配置完成后单击确定，完成创建。

容器服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击服务来源。
3. 单击创建来源，在创建来源面板中，配置以下参数：

- 来源类型：选择容器服务，通过K8s Server发现后端服务。
- 容器集群：选择后端服务所在的集群。

注意：只能添加同VPC内的容器集群，不支持跨VPC的服务来源。

4. 配置完成后单击确定，完成服务来源创建。

5. 单击服务管理页签，然后单击创建服务，配置以下参数：
   - 服务来源：选择容器服务。
   - 命名空间：选择服务所在的命名空间。
   - 服务列表：命名空间下的服务列表。
6. 配置完成后单击确定，完成服务创建。

MSE Nacos

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击服务来源。
3. 单击创建来源，在创建来源面板中，配置以下参数：
   - 来源类型：选择MSE Nacos，通过 MSE Nacos 注册中心发现后端服务。
   - 集群名称：选择后端服务所在的MSE Nacos集群。

   注意：只能添加同VPC内的MSE Nacos集群，不支持跨VPC的服务来源。

   # 拉取间隔：网关节点定时从来源实例同步服务注册信息的间隔，取值范围为15s~60s。
4. 配置完成后单击确定，完成服务来源创建。
5. 单击服务管理页签，然后单击创建服务，配置以下参数：
   - 服务来源：选择MSE Nacos集群。
   - 命名空间：选择服务所在的命名空间。
   - 服务列表：命名空间下的服务列表。
6. 配置完成后单击确定，完成服务创建。

固定地址

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。
3. 配置基本信息。
   - 在弹窗中先选择服务来源为“固定地址”，随后编辑其余配置。
   - 服务名称： 自定义服务名称。
   - 服务地址：单击添加， 输入后端节点主机地址和端口，支持同时添加多个服务地址。存在多个服务地址时，优先路由到优先级高的服务地址，若优先级相同，则按照权重按比例路由到服务地址。
   - 请求协议：转发至后端服务的请求协议，支持HTTP、HTTPS、GRPC、GRPCS和DUBBO。
4. 配置完成后单击确定，完成服务创建。

函数计算

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。

3. 配置基本信息。

   在弹窗中先选择服务来源为“函数计算”，随后编辑其余配置。

   - 函数：从函数列表中选择函数。
   - 版本/别名：选择函数的版本或别名。
   - 请求协议：函数的请求协议，支持HTTP。

4. 配置完成后单击确定，完成服务创建。

DNS域名

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击创建服务。
3. 配置基本信息。

   - 在弹窗中先选择服务来源为“DNS域名”，随后编辑其余配置。
   - 服务名称： 自定义服务名称。
   - 域名： 后端服务的域名地址。

   注意：如果指定的是外网地址，需要配置NAT网关。

   - 端口： 后端服务的端口。
   - 请求协议： 后端服务的请求协议，支持HTTP、HTTPS、GRPC、GRPCS和DUBBO。

管理服务

概述

完成服务创建后，可以在服务管理页面查看已创建的服务列表，同时也可以对服务进行查看、编辑和删除等操作。

查看服务

服务完成创建后，可以在服务管理页面查看服务。

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务。
3. 查看服务时，支持通过服务来源和服务名称筛选服务。
4. 单击服务名称，可跳转至服务概览页面查看服务的详细信息。

编辑服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务。
3. 单击需要变更的服务对应的编辑，在编辑服务面板，修改服务相关参数，然后单击确定。

删除服务

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务。
3. 单击需要变更的服务对应的删除。
4. 在删除服务面板，单击确定，完成删除服务。

服务来源

管理服务来源

概述

支持添加云容器引擎和注册配置中心作为服务来源，通过服务发现模块监听服务来源，动态感知后端服务。

创建云容器引擎服务来源

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击服务来源。
3. 单击创建来源，在创建来源面板中，配置以下参数：
   - 来源类型：选择容器服务，通过K8s Server发现后端服务。
   - 容器集群：选择后端服务所在的集群。

> 注意：只能添加同VPC内的容器集群，不支持跨VPC的服务来源。

创建注册配置中心服务来源

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，然后单击服务来源。
3. 单击创建来源，在创建来源面板中，配置以下参数：
   - 来源类型：选择MSE Nacos，通过 MSE Nacos 注册中心发现后端服务。
   - 集群名称：选择后端服务所在的MSE Nacos集群。

> 注意：只能添加同VPC内的MSE Nacos集群，不支持跨VPC的服务来源。

   - 拉取间隔：网关节点定时从来源实例同步服务注册信息的间隔，取值范围为15s~60s。

删除服务来源

1. 打开AI网关控制台实例页面，在顶部菜单栏选择目标实例所在地域，并单击目标实例ID。
2. 在左侧导航栏，单击服务，随后单击服务来源。
3. 单击需要变更的服务来源对应的删除。
4. 在删除服务面板，单击确定，完成删除服务来源。

消费者管理

创建消费者

操作步骤

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 单击 "创建消费者" ，在创建消费者面板中配置相关参数，然后单击 "保存" 。

配置项	描述
消费者名称	自定义消费者名称，要保证唯一性 命名规则：支持英文字母、数字和下划线，以英文字母或数字开头及结尾，4-32个字符
启用状态	消费者状态包括已启用和已停用。创建完成后可以手动启停 注意 消费者只有启用状态时才生效。
描述	对消费者进行描述
认证方式	当前AI网关中消费者支持JWT和KEY两种认证方式。创建认证时必须选择至少一种认证方式。删除消费者下的认证时也要保证至少存在一种认证方式。对于KEY认证的key，填写时需保证唯一，例如消费者1和消费者2下的KEY认证不可设置相同的key值。 **JWT认证**：JSON Web Token (JWT) 是一种用于在客户端和服务器之间以紧凑且自包含的JSON对象形式安全地传输信息。JWT通过数字签名确保信息的完整性和真实性，常用于在网关中验证用户身份并控制对资源的授权访问，从而实现无状态的身份验证和授权机制。 秘钥类型：可选择对称秘钥和非对称秘钥。 加密算法：对称秘钥加密算法可选择HS256和HS512。非对称秘钥加密算法可选择RS256和ES256。 秘钥：对称秘钥时需填写secret，如果开启base64secret则需按照base64编码格式填写。非对称秘钥时需填写RSA/ES公钥和私钥。 **KEY认证**：KEY认证是一种基础的认证机制，客户端在调用API时需将凭证（如API Key）以特定方式添加到请求中，网关接收到请求后会校验Key的有效性和权限范围。这种认证方式安全性较低，不建议用于涉及敏感操作的场景。 Bearer前缀：如果勾选，则访问时需携带"Bearer XXXXXXXXXX"的Key值；如果不勾选，则直接携带Key值访问。 Key：自定义Key凭证。

启用消费者

操作步骤

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 在消费者列表页操作栏中点击 "启用"，在弹出提示框中点击 "确认"。
4. 或进入目标消费者详情页，在基本信息栏的状态 点击 "启用"，在弹出提示框中点击 "确认"。

消费者授权管理

授权消费者

操作步骤1：在消费者管理侧授权

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 进入目标消费者详情页，选择消费者授权页签，单击"添加授权"按钮。

4. 在授权面板中，配置相关参数，单击"保存"。

配置项	描述
授权范围	AI网关支持四种维度的授权范围： • 实例：若授权消费者给实例，则作用于该实例下的所有API • Model API：若授权消费者给Model API，则作用于该API下的所有路由 • MCP服务：若授权消费者给MCP服务，则作用于该MCP服务请求 • Agent API：若授权消费者给Agent API，则作用于该API下的所有路由
有效期	选择此次消费者授权的有效期，如果不选择，则默认永久有效
授权实例	选择需要授权的实例
授权API	选择需要授权的API

操作步骤2：在资源端侧授权

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，进入"AI网关-实例"，找到不同资源端的消费者管理页面。
3. 点击"授权"按钮，选择要授权的消费者列表，单击"确定"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于Model API，在左侧导航栏，选择 "Model API"，进入"Model API"详情页>消费者认证页签。
- 对于MCP服务，在左侧导航栏，选择 "MCP管理-MCP服务"，进入"MCP服务"详情页>消费者认证页签。
- 对于Agent API，在左侧导航栏，选择 "Agent API"，进入"Agent API"详情页>消费者认证页签。

解除授权

操作步骤1：在消费者管理侧解除授权

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 进入目标消费者详情页，选择消费者授权页签。
4. 在已授权列表中，展开实例，支持批量勾选后点击"解除授权"，也支持对单个资源右侧操作栏点击"解除授权"。

操作步骤2：在资源端侧解除授权

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，进入"AI网关-实例"，找到不同资源端的消费者管理页面。
3. 已授权消费者列表右侧操作栏中点击"解除授权"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于Model API，在左侧导航栏，选择 "Model API"，进入"Model API"详情页>消费者认证页签。
- 对于MCP服务，在左侧导航栏，选择 "MCP管理-MCP服务"，进入"MCP服务"详情页>消费者认证页签。
- 对于Agent API，在左侧导航栏，选择 "Agent API"，进入"Agent API"详情页>消费者认证页签。

资源端开启消费者认证

注意

在资源端开启消费者认证后，需为当前API绑定消费者授权关系，否则该API无法访问。

操作步骤

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，进入"AI网关-实例"，找到不同资源端的消费者管理页面。
3. 点击编辑配置信息，填写如下配置信息，单击"确定"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。

说明

- 在实例上开启消费者认证后，则将对该实例下的所有API都生效，请谨慎开启。
- 实例上的认证是独立的，例如在实例上开启了JWT认证，在实例下的API开启KEY认证，则访问该路由时需要同时携带两种认证方式。如果在实例和API上开启的是同一种认证，则需要携带相同的消费者认证才能访问。

- 对于Model API，在左侧导航栏，选择 "Model API"，进入"Model API"详情页>消费者认证页签。
- 对于MCP服务，在左侧导航栏，选择 "MCP管理-MCP服务"，进入"MCP服务"详情页>消费者认证页签。
- 对于Agent API，在左侧导航栏，选择 "Agent API"，进入"Agent API"详情页>消费者认证页签。

配置项	描述
启用状态	开启后，认证鉴权生效，访问其下资源需要携带对应认证信息
认证方式	当前路由认证消费者时使用的认证方式。目前AI网关中支持JWT和KEY的认证方式
Token配置	访问携带Token相关配置信息
JWT Token	JWT Token 配置信息 隐藏认证信息：是否将认证信息透传到后端服务。 Header：设置从哪个header获取token，优先级最高默认值为'authorization'，使用方式 -H 'authorization: token值' Query：设置从哪个query string获取token，优先级低于header。默认值为'jwt'，使用方式 '?jwt=token值' Cookie：设置从哪个cookie获取token，优先级低于query。默认值为'jwt'，使用方式 '--cookie=jwt值'
KEY Token	KEY Token 配置信息。 隐藏认证信息：是否将认证信息透传到后端服务 Header：设置从哪个header获取token，优先级最高。默认值为'apiKey'，使用方式 -H 'apikey: token值' Query：设置从哪个query string获取token，优先级低于header。默认值为'apiKey'，使用方式 '?apikey=token值'

资源端关闭消费者认证

操作步骤

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，进入"AI网关-实例"，找到不同资源端的消费者管理页面。
3. 点击编辑配置信息，启用状态选择"关闭"。

- 对于实例，在左侧导航栏，选择 "实例"，进入实例概览页>消费者认证页签。
- 对于Model API，在左侧导航栏，选择 "Model API"，进入"Model API"详情页>消费者认证页签。
- 对于MCP服务，在左侧导航栏，选择 "MCP管理-MCP服务"，进入"MCP服务"详情页>消费者认证页签。
- 对于Agent API，在左侧导航栏，选择 "Agent API"，进入"Agent API"详情页>消费者认证页签。

停用或删除消费者

停用消费者

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 在消费者列表页操作栏中点击 "停用"，在弹出提示框中点击 "确认"。
4. 或进入目标消费者详情页，在基本信息栏的状态 点击 "停用"，在弹出提示框中点击 "确认"。

删除消费者

注意

删除消费者前请先停用消费者。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-消费者"。
3. 在消费者列表页面，单击目标消费者的操作列下的"删除"，然后在确认删除对话框中输入当前的消费者名称，然后单击"删除"。

域名管理

创建域名

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 单击 "创建域名"，在创建域名面板中配置相关参数，然后单击 "确定"。

参数	描述
协议	可选择HTTP或HTTPS协议，不同协议支持的端口为： • HTTP：支持27151端口 • HTTPS：支持27154端口

参数	描述
域名	支持完整域名和泛域名的形式，例如绑定*.ctyun.cn这个域名后，可通过a.ctyun.cn，1.ctyun.cn等来访问。 域名填写规则为：域名包含英文字母、数字和连接符(-)，连接符不能在每段首尾或连续出现，每段长度不得超过63个字符且最后一段为2-6个英文字符
选择协议为HTTPS协议时	
证书	选择HTTPS协议时需上传证书或选择已有证书 注意 所选择的证书域名需与所填写域名相匹配。 如证书域名为*.ctyun.cn，则所填写域名可以是*.ctyun.cn，a.ctyun.cn等。

绑定场景

- Model API可绑定域名
- MCP服务可绑定域名
- Agent API可绑定域名

删除域名

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 在域名列表页面，选择目标域名名称的操作列下单击"删除"，然后在确认删除对话框中输入当前的域名，然后单击"删除"。

说明

删除域名前请先确保没有在线MCP服务或Model API、Agent API正在使用该域名。

更换域名协议或证书

当发生证书到期、域名所有者有变更、网站需从HTTP升级到HTTPS等情况时，需要变更域名的证书和协议，来保障网站或平台的安全性与合规性。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 在域名列表页面，选择目标域名名称的操作列下单击"编辑"。
4. 更换协议：单击域名右侧的下拉列表选择协议，然后单击"确定"。
5. 更换证书：单击证书右侧的下拉列表选择证书，或上传新的证书，然后单击"确定"。

上传证书

当域名为HTTPS协议时，必须选择对应的证书文件。

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 在创建或编辑域名页面，选择HTTPS协议时，点击 "上传证书"。
4. 在上传证书面板中配置相关参数，然后单击 "上传 "。

参数	描述
证书名称	标记证书的名称
证书文件	证书文件，支持pem、cer，crt，key格式
私钥文件	私钥文件，支持pem、cer，crt，key格式。需与证书文件相匹配

编辑证书

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 在创建或编辑域名页面，选择HTTPS协议时，在证书列表中，选择目标证书项的操作列下单击"修改"。
4. 在修改证书面板中重新上传证书文件和私钥文件，然后单击 "上传"。

删除证书

操作步骤

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "域名"菜单。
3. 在创建或编辑域名页面，选择HTTPS协议时，在证书列表中，选择目标证书项的操作列下单击"删除"，然后在确认删除对话框中单击"删除"。

说明

删除证书前请先确保没有域名正在使用该证书。

插件市场

管理插件

安装插件

安装插件是指将AI网关插件市场中的插件安装到具体的网关实例的过程。有两种方式可以安装插件：

操作步骤1：在插件市场安装

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-插件"。
3. 在插件页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击插件卡片上的"安装"，在弹出的安装插件框中选择需要使用此插件的网关实例，单击"确定"。

操作步骤2：在网关实例中安装

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例详情。
3. 在左侧导航栏，选择"插件"。
4. 单击"安装插件"按钮，在安装插件页面的快捷导航处，选择要安装的插件类型或者搜索插件名称，单击插件卡片，在弹出的安装插件框中，单击"安装"。

卸载插件

当您想要将插件彻底从网关上删除时，您可以选择卸载插件。有两种方式可以卸载插件：

注意

卸载插件时，如果存在启用的插件规则，请先停用插件再卸载；如果插件未启用，卸载插件会将配置的插件规则一并删除。

操作步骤1：在插件市场卸载

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-插件"。
3. 在插件市场页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击要卸载的插件卡片。
4. 单击配置栏，在要卸载此插件的网关实例操作栏中，单击"卸载"。
5. 在弹出框中，点击"确认"按钮，页面提示卸载插件成功。

操作步骤2：在网关实例中卸载

1. 登录云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例详情。
3. 在左侧导航栏，选择"插件"。
4. 在插件列表中，单击所要卸载插件操作列中的"卸载"。
5. 在弹出框中单击"确认"按钮，页面提示卸载插件成功。

平台插件

传输协议

fault-injection插件

功能说明

fault-injection 插件是故障注入插件，用于模拟服务故障。它可以在特定条件下人为引入延迟、返回错误状态码或自定义响应，从而帮助开发者和测试人员测试系统的容错能力和服务在异常情况下的表现。该插件可以和其他插件一起使用，并在其他插件执行前被执行。

配置字段

名称	类型	填写要求	默认值	有效值	描述
abort	object	abort与delay至少配置一个			abort 属性将直接返回给客户端指定的响应码并且终止其他插件的执行。

名称	类型	填写要求	默认值	有效值	描述
delay	object	abort与delay至少配置一个			delay 属性将延迟某个请求，并且还会执行配置的其他插件。

子项abort中每一项的配置字段说明如下。

名称	类型	填写要求	默认值	有效值	描述
http_status	integer	必填		[200, 599]	返回给客户端的 HTTP 状态码
body	string	可选			返回给客户端的响应数据。支持使用 NGINX 变量，如 client addr: $remote_addr
headers	object	可选			返回给客户端的响应头，可以包含 NGINX 变量，如 $remote_addr
percentage	integer	可选		[0, 100]	将被中断的请求占比
vars	array[]	可选			执行故障注入的规则，当规则匹配通过后才会执行故障注。vars 是一个表达式的列表，来自 lua-resty-expr。

子项delay中每一项的配置字段说明如下。

名称	类型	填写要求	默认值	有效值	描述
duration	number	必填			延迟时间，可以指定小数
percentage	integer	可选		[0, 100]	将被延迟的请求占比
vars	array[]	可选			执行请求延迟的规则，当规则匹配通过后才会执行故障注。vars 是一个表达式的列表，来自 lua-resty-expr。

注：vars 是由 lua-resty-expr 的表达式组成的列表，它可以灵活的实现规则之间的 AND/OR 关系，示例如下：

```
[
  [
    ["arg_name","==","jack"],
    ["arg_age","==",18]
  ],
  [
    ["arg_name2","==","allen"]
  ]
]
```

以上示例表示前两个表达式之间的关系是 AND，而前两个和第三个表达式之间的关系是 OR。

配置示例

场景1：故障注入

```
abort:
 http_status: 503
 body: "Fault Injection!"
```

根据该配置，路由请求时会被拦截，返回自定义的状态码和响应体。

场景2：请求延迟

```
delay:
 duration: 3
```

根据该配置，路由请求时会延迟3秒后再执行

场景3：带条件的故障注入和请求延迟

```
abort:
 http_status: 504
 body: "Fault Injection!"
 vars: [
          [
            ["arg_name","==","jack"]
          ]
       ]
delay:
 duration: 3
 vars: [
          [
            ["http_age","==","18"]
          ]
       ]
```

根据该配置，当请求参数中name值为jack的路由时会被拦截

```
curl http://example.com/test?name=jack
```

```
HTTP/1.1 503 Service Temporarily Unavailable
......
Fault Injection!
```

当请求header中age值为18的路由时会延迟3秒执行

```
time curl http://example.com/test -H 'age: 18'

HTTP/1.1 200
......
real 0m3.008s
user 0m0.003s
sys 0m0.003s
```

配置模板

```
#abort和delay属性至少要配置其中一个
#abort属性直接返回给客户端指定的响应码并且终止其他插件的执行
abort:
  #[必填]返回客户端的HTTP状态码，有效范围[200,599]
  http_status: 503
  #[可选]返回给客户端的响应数据
  #body: "Fault Injection!"
  #headers: {"X-Error-Code": "12345","X-Reason": "Invalid request"}
  #[可选]将被中断的请求占比，有效范围[0,100]
  #percentage: 100
  #[可选]执行故障注入的规则，当规则匹配通过后才会执行故障
  #vars: [[[ "arg_name","==","jack" ]]]
#delay属性将延迟某个请求，并且会执行配置的其他插件
delay:
  #[必填]延迟时间，可以指定小数
  duration: 3
  #[可选]将被延迟的请求占比，有效范围[0,100]
  #percentage: 100
  #[可选]执行请求延迟的规则，当规则匹配通过后才会执行故障
  #vars: [[[ "http_age","==","18" ]]]
```

安全防护

uri-blocker 插件

功能说明

uri-blocker 插件通过指定一系列 block_rules 来拦截用户请求，实现了基于URI屏蔽HTTP请求，并且自定义返回码和响应体，可以用于防护部分资源不对外部暴露。

配置字段

名称	类型	填写要求	默认值	有效值	描述
block_rules	array[string]	必填			正则过滤数组。它们都是正则规则，如果当前请求 URI 命中其中任何一个，则将响应代码设置为 rejected_code 以退出当前用户请求。数组中的正则规则需保证唯一性。例如：["root.exe", "root.m+"]。
rejected_code	integer	可选	403	[200, 599]	当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 状态代码。
rejected_msg	string	可选		非空	当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 响应体。
case_insensitive	boolean	可选	false		是否忽略大小写。当设置为 true 时，在匹配请求 URI 时将忽略大小写。

配置示例

uri-blocker 使用示例

```
block_rules:
 - root.m+
 - root.exe
rejected_code: 405
rejected_msg: “This uri is not allowed to be visited”
case_insensitive: true
```

根据该场景请求路由

```
curl http://example.com/root.exe
curl http://example.com/Root.exe
```

当前请求的URI命中了配置中的正则表达式，且在匹配时忽略大小写。请求将返回应答如下

```
HTTP/1.1 405 Not Allowed
```

.....

{"error_msg":"This uri is not allowed to be visited"}

配置模板

```
#[必填]正则过滤数组，不可重复。如果当前请求 URI 命中其中任何一个，则将响应代码设置为 rejected_code 以退出当前用户请求
block_rules:
 - test.com+
#[可选]当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 状态代码。有效值[200,599], 默认值403
#rejected_code: 403
#[可选]当请求 URI 命中 block_rules 中的任何一个时，将返回的 HTTP 响应体
#rejected_msg: “This URI is not allowed to be accessed ”
#[可选]匹配URI时是否忽略大小写.默认false
#case_insensitive: false
```

流量管控

limit-count插件

功能说明

limit-count 插件使用固定时间窗口算法，主要用于限制单个客户端在指定的时间范围内对服务的总请求数，并且会在 HTTP 响应头中返回剩余可以请求的个数.注意：limit-count 插件的速率限制策略默认是针对单个实例节点，即按照实例节点上的请求数来计数，当存在节点达到数量阈值后请求会被拒绝。如果想对实例集群限速，则policy需指定为redis，并填写redis相关配置信息。

配置字段

名称	类型	填写要求	默认值	有效值	描述
count	integer	必填		count> 0	每个客户端在指定时间窗口内的总请求数量阈值。
time_window	integer	必填		time_window> 0	时间窗口的大小（以秒为单位）。超过该属性定义的时间，则会重新开始计数。
rejected_code	integer	可选	429	[200,599]	当请求超过阈值被拒绝时，返回的 HTTP 状态码。
key_type	string	可选	"var"	["var", "var_combination", "constant"]	key 的类型。

名称	类型	填写要求	默认值	有效值	描述
key	string	可选	"remote_addr"		用来做请求计数的依据。如果 key_type 为 var，则 key 将被解释为变量。变量不需要以美元符号（$）为前缀。如果 key_type 为 var_combination，那么 key 会被当作变量组合，所有变量都应该以美元符号($) 为前缀。如 $remote_addr $consumer_name，插件会同时受 $remote_addr 和 $consumer_name 两个变量的约束；如果 key_type 为 constant，那么 key 会被当作常量。如果 key 的值为空，$remote_addr 会被作为默认 key。
rejected_msg	string	可选		非空	当请求超过阈值被拒绝时，返回的响应体。
show_limit_quota_header	boolean	可选	true		当设置为 true 时，在响应头中显示 X-RateLimit-Limit（限制的总请求数）和 X-RateLimit-Remaining（剩余还可以发送的请求数）字段。
group	string	可选		非空	对多个API设置相同 group，则路由可以共享相同的速率限制计数器。注意同一个group下的插件配置值需相同。

名称	类型	填写要求	默认值	有效值	描述
policy	string	可选	"local"	["local","redis"]	速率限制计数器的策略。如果是 local，则计数器存储在本地内存中，按照单个实例节点计数。如果是 redis，则计数器存储在 Redis 实例上，按照实例集群计数。
redis_host	string	当 policy 为 redis 时必填			Redis 节点的地址。
redis_port	integer	当 policy 为 redis 时必填	6379	[1,65535]	Redis 节点的端口。
redis_username	string	可选			如果使用 Redis ACL，则为 Redis 的用户名。如果使用旧式身份验证方法 requirepass，则仅配置 redis_password。
redis_password	string	可选			Redis 节点的密码。

配置示例

limit-count 使用示例

```
count: 2
time_window: 60
rejected_code: 429
```

根据该配置场景，其限制了 60 秒内请求只能访问 2 次。请求以下路由

```
curl -i http://example.com/index.html
```

在执行测试命令的前两次都会正常访问。其中响应头中包含了 X-RateLimit-Limit 和 X-RateLimit-Remaining 和 X-RateLimit-Reset 字段，分别代表限制的总请求数和剩余还可以发送的请求数以及计数器剩余重置的秒数：

```
HTTP/1.1 200 OK
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

当第三次进行测试访问时，会收到包含 503 HTTP 状态码的响应头，目前在拒绝的情况下，也会返回相关的头，表示插件生效：

```
HTTP/1.1 503 Service Temporarily Unavailable
......
X-RateLimit-Limit: 2
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 58
```

配置模板

基础配置案例

```
# [必填]时间窗口内的请求数量阈值。
count: 30
# [必填]时间窗口的大小（以秒为单位）
time_window: 60
# [可选]请求超过阈值被拒绝时，返回的 HTTP 状态码
#rejected_code: 429
# [可选]当设置rejected_msg时，非空。默认可不填
# rejected_msg: "Requests are too frequent, please try again later."
```

client-control插件

功能说明

client-control 插件通过设置客户端请求体大小上限来动态控制客户端的请求。当设置较大的限制时可能导致内存使用增加，需根据实际需求合理配置。

配置字段

名称	类型	填写要求	默认值	有效值	描述
max_body_size	integer	可选		[0,...]	动态设置 client_max_body_size 的大小

配置示例

client-control 使用示例

max_body_size: 1

根据该场景请求路由

curl http://example.com/test -d '123'

由于请求路由的请求体大小超过了所设置的客户端请求体大小上限，请求返回413.

```
HTTP/1.1 413 Request Entity Too Large
......
```

配置模板

```
#[可选]设置客户端请求体大小上限.有效范围[0,...]
```

#max_body_size: 1024

AI插件

AI提示词校验插件

功能说明

该插件通过检查和验证输入的提示消息来保护您的AI端点。它根据用户定义的允许和拒绝模板检查请求的内容，以确保仅处理批准的输入。根据其配置，该插件可以仅检查最新消息或整个对话历史记录，并且可以设置为检查来自所有角色或仅来自最终用户的提示。当同时配置允许和拒绝模板时，插件首先确保至少匹配一个允许的模板，如果没有匹配，则会返回“Request does not match allow patterns“错误来拒绝请求。如果找到允许的模板，就会检查是否出现任何被拒绝的模板，如果检测到，则会返回“Request contains prohibited content”错误来拒绝请求。

配置字段

名称	类型	填写要求	默认值	描述
match_all_roles	boolean	可选	false	如果设置为true，插件将检查所有角色的提示消息。否则，它仅在其角色为 user 时验证。
match_all_conversation_history	boolean	可选	false	如果设置为true，对话历史记录中的所有消息都会检查。如果为 false，则仅检查最后一条消息的内容。
allow_patterns	array[string]	可选		允许的匹配规则列表。提供时，提示词必须与至少一个规则匹配才能被视为有效。
deny_patterns	array[string]	可选		禁止的匹配规则列表。如果这些模式中的任何一个与提示词内容匹配，则请求将被拒绝。

配置示例

使用示例：

```
allow_patterns:
 - goodword
deny_patterns:
 - badword
```

根据该场景请求路由：

```
curl http://127.0.0.1:27151/v1/chat/completions -i -XPOST -H 'Content-Type: application/json' -d
'{
```

```
  "model": "gpt-4",
  "messages": [{ "role": "user", "content": "badword request" }]
}'
```

则这个请求会返回400异常：

```
HTTP/1.1 400 Bad Request
......

{"message":"Request doesn't match allow patterns"}
```

配置模板

基础配置案例

```
# [可选]默认只检查user的提示词，如果设置为true，插件将检查所有角色的提示消息
#match_all_roles: false
# [可选]默认只检查最后一条提示词，如果设置为true，对话历史记录中的所有消息都会检查
#match_all_conversation_history: false
# [可选]允许的匹配规则列表
allow_patterns:
  - goodword
# [可选]禁止的匹配规则列表
deny_patterns:
  - badword
```

插件配置管理

启用插件配置

说明

插件的生效范围：

- **API级插件规则**：请求匹配到某个 API 时生效，作用于其下所有路由。
- **实例级插件规则**：启用即生效网关全局，独立执行，且在API 规则前执行。

注意

AI类型插件只能作用于Model API级别

操作步骤1：在插件市场中启用

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-插件"。
3. 在插件市场页面的快捷导航栏处，选择插件类型或者搜索插件名称，单击插件卡片上的"配置"。
4. 单击目标网关实例操作列下的"规则配置"，在规则配置页面选择生效范围。
   - 当选择API级（包括Model API/MCP服务/Agent API级别）插件规则时，单击"添加规则"，打开"启用"状态，配置插件规则，单击保存。
   - 当选择实例级插件规则时，打开"启用"状态并配置插件规则，单击保存。

操作步骤2：在实例上启用

1. 登录 云原生API网关控制台，并在顶部菜单栏选择地域。
2. 在左侧导航栏，选择 "AI网关-实例"，进入目标实例详情页面。
3. 在左侧导航栏，选择"插件"，单击插件列表操作列中的"规则配置"，为所选插件配置规则并选择生效范围，单击保存。

观测分析

开启网关日志采集

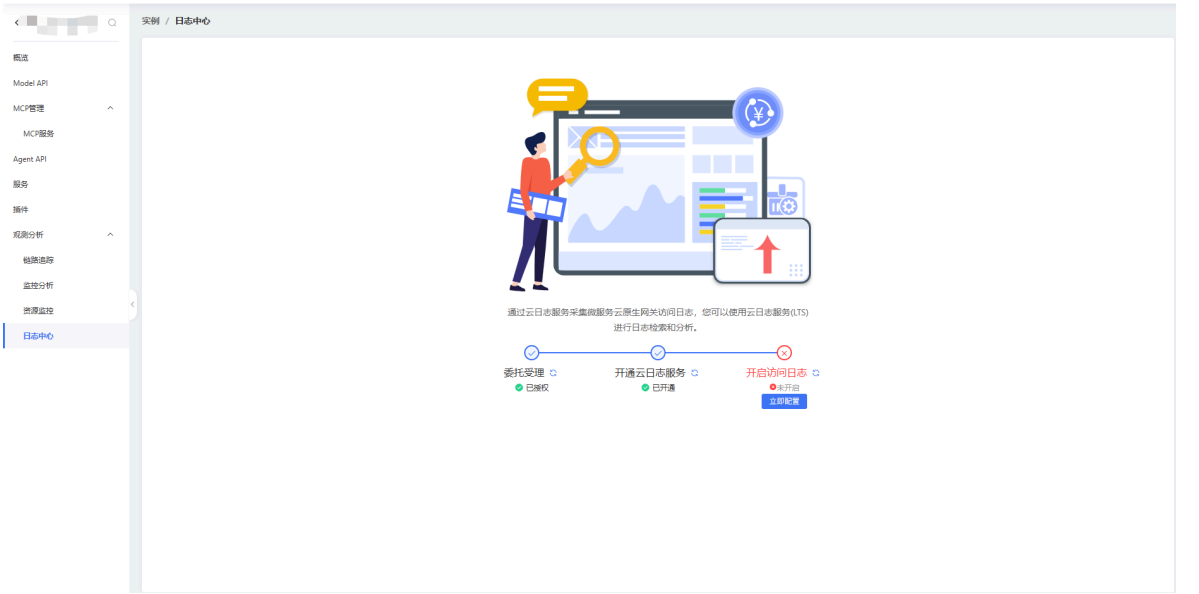
AI网关对接天翼云日志服务（LTS）实现了访问日志采集、上报和查询能力。开启日志采集后，您可以通过分析AI网关的访问日志了解客户端用户行为，以便排查问题。

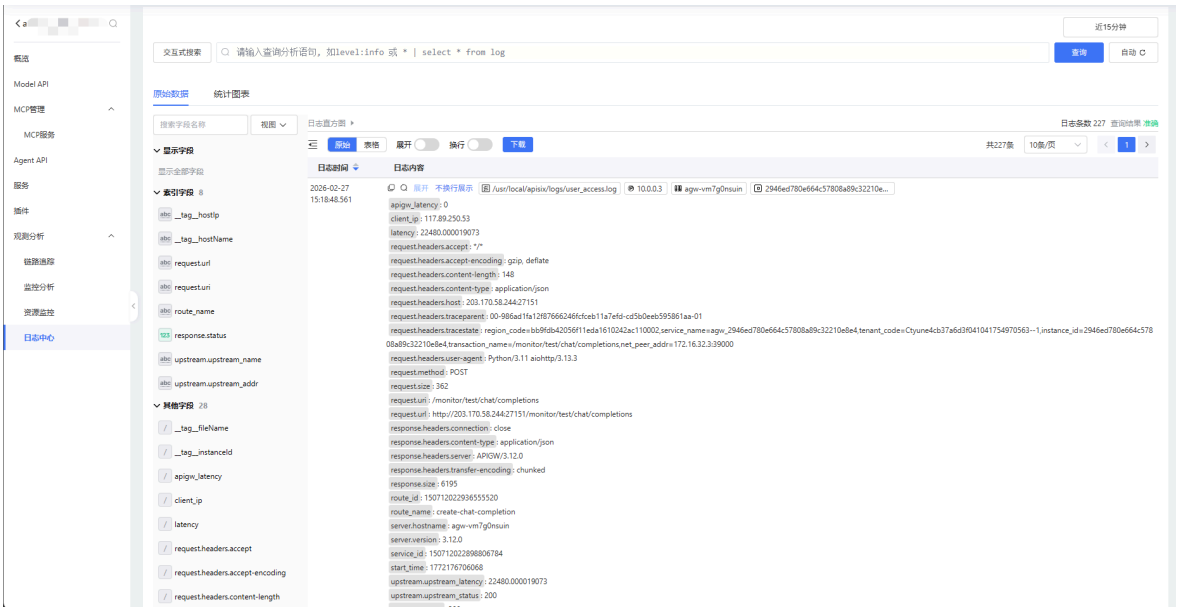
操作步骤

1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏” AI网关” 下，点击 "实例"菜单。
3. 在实例列表页，点击实例ID或者实例名称进入实例详情页，在左侧导航栏中，选择 "观测分析"。
4. 选择 " 日志中心"，有三个步骤进行开启：
   - 检查是否开启委托授权，如未开启，需要点击 "立即创建"，创建名为 CtyunAssumeRoleForAgw 的委托授权。
   - 检查是否开通日志服务，如未开通，需要点击 "立即开通"，开通日志服务。
   - 检查是否接入日志，如未接入，需要点击 "开启"，接入日志服务。

说明

- 日志和链路追踪的启用需要您授权AI网关访问云日志服务和应用性能监控服务的权限，允许使用相关资源。
- 日志和链路追踪服务将根据您的使用量进行收费。





访问日志格式说明

网关访问日志示例


```
{
  "server": {
    "version": "3.12.0",
    "hostname": "agw-vm7g0nsuin"
  },
  "response": {
```

```
    "size": 6195,
    "status": 200,
    "headers": {
      "connection": "close",
      "server": "APIGW/3.12.0",
      "content-type": "application/json",
      "transfer-encoding": "chunked"
    }
  },
  "request": {
    "method": "POST",
    "querystring": {},
    "size": 362,
    "url": "http://203.170.xx.xx:27151/monitor/test/chat/completions",
    "headers": {
      "user-agent": "Python/3.11 aiohttp/3.13.3",
      "accept": "*/*",
      "content-type": "application/json",
      "traceparent": "00-986ad1fa12f87666246fcfceb11a7efd-cd5b0eeb595861aa-01",
      "tracestate": "region_code=xx,service_name=agw_2946ed78
0e664c57808a89c32210e8e4,tenant_code=Ctyune4cb37a6d3f041041754970563--
1,instance_id=2946ed78
0e664c57808a89c32210e8e4,transaction_name=/monitor/test/chat/completions,net_peer_addr=172.16.x.x:39
      "host": "203.170.xx.xx:27151",
      "content-length": "148",
      "accept-encoding": "gzip, deflate"
    },
    "uri": "/monitor/test/chat/completions"
  },
  "start_time": 1772176706068,
  "route_id": "150712022936555520",
  "route_name": "create-chat-completion",
  "upstream": {
    "upstream_latency": 22480.000019073,
    "upstream_status": 200,
    "upstream_name": ""
  },
  "latency": 22480.000019073,
  "service_id": "150712022898806784",
  "apigw_latency": 0,
  "client_ip": "117.89.xx.xx"
}
```


日志索引字段说明如下

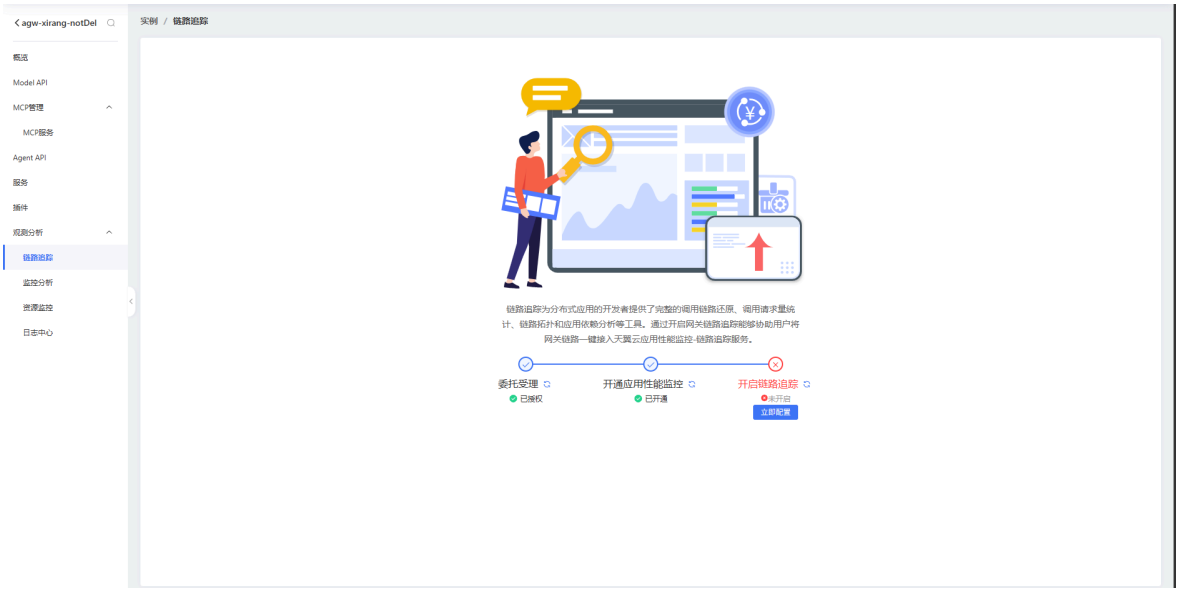
字段	说明
__tag__hostIp	数据来源主机 IP
__tag__hostName	数据来源主机名称
request.url	请求 url
request.uri	请求 uri
route_name	路由名称
response.status	响应结果状态
upstream.upstream_name	上游服务名称
upstream.upstream_addr	上游服务地址

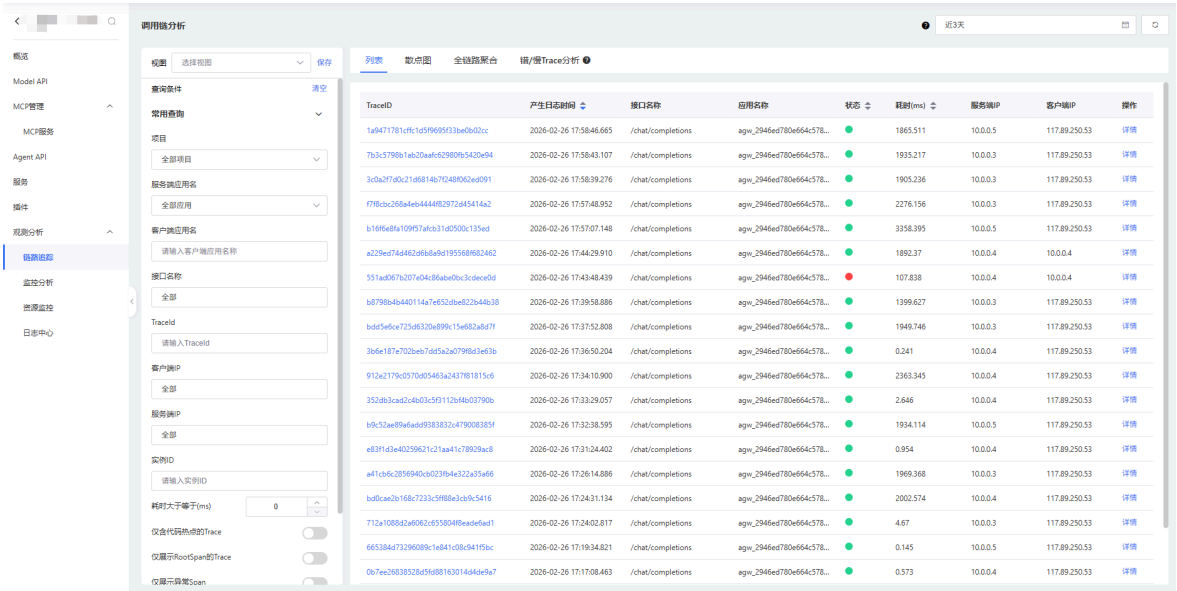
开启网关链路追踪

在开启了链路追踪并配置采样率大于0，网关会根据采样率配置上报链路追踪数据，链路追踪基于 traceid 将调用链上下游串联起来，帮助您分析和诊断分布式应用架构下的性能瓶颈，提高微服务时代下的开发诊断效率。

操作步骤

1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏”AI网关”下，点击 "实例"菜单。
3. 在实例列表页，点击实例ID或者实例名称进入实例详情页，在左侧导航栏中，选择 "观测分析"。
4. 选择 "链路追踪"，有三个步骤进行开启：
   - 检查是否开启委托授权，如未开启，需要点击 "立即创建"，创建名为CtyunAssumeRoleForAgw的委托授权。
   - 检查是否开通应用监控性能服务，如未开通，需要点击 "立即开通"，开通应用监控性能服务。
   - 检查是否接入链路追踪，如未接入，需要点击 "开启"，接入链路追踪。

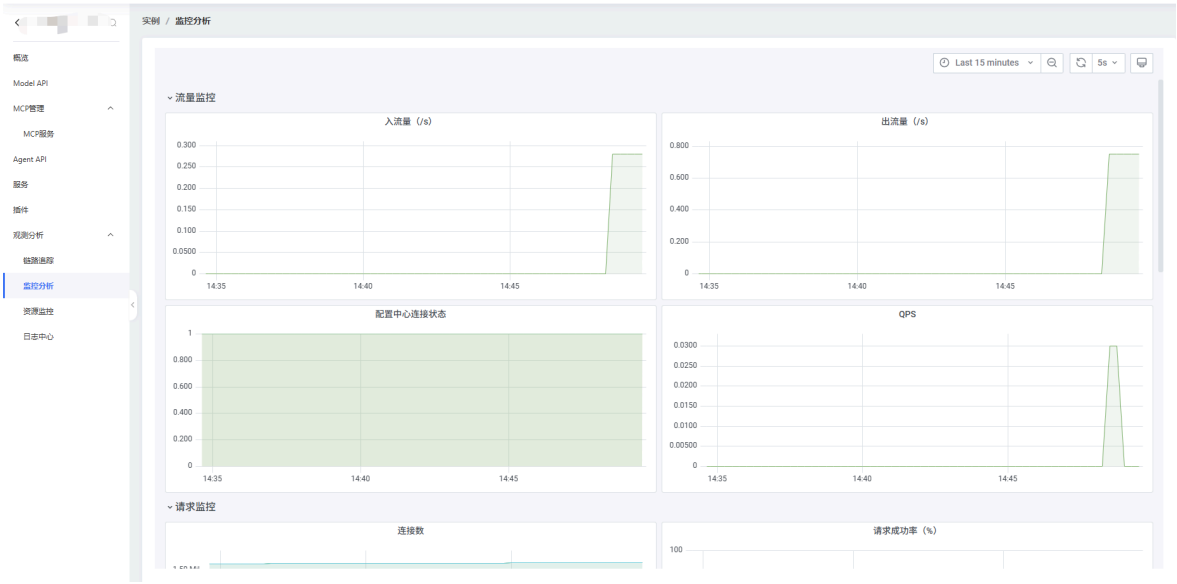




查看网关监控数据

操作步骤

1. 登录云原生API网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏“AI网关”下，点击 "实例"菜单。
3. 在实例列表页面，点击实例ID或者实例名称进入实例详情页。
4. 进入实例详情页，在左侧导航栏中，选择 "观测分析"，点击“监控分析”。
5. 在监控分析页面，可看到该实例最近时刻的入流量、出流量、QPS，请求成功率和平均延迟等指标信息。



查看网关资源监控数据

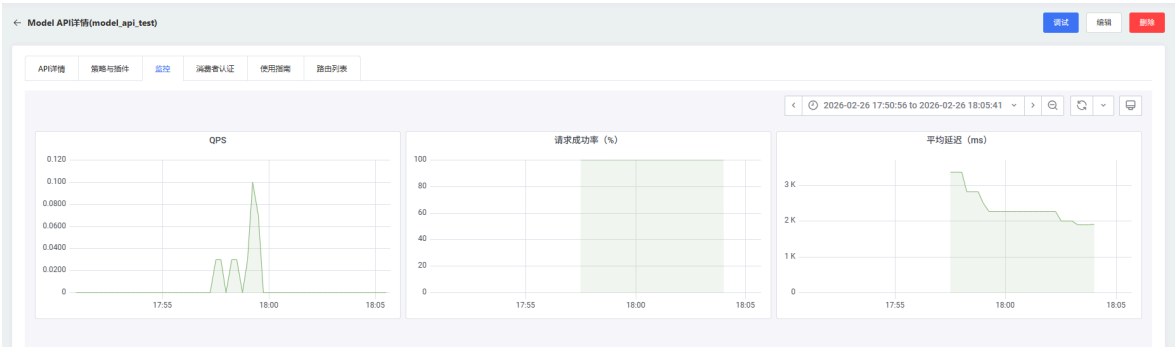
操作步骤

1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏“AI网关”下，点击 "实例"菜单。
3. 在实例列表页面，点击实例ID或者实例名称进入实例详情页。
4. 进入实例详情页，在左侧导航栏中，选择 "观测分析"，点击“资源监控”。
5. 在资源监控页面，可看到该实例每个节点最近时刻的 CPU 使用率，内存使用率，磁盘和网络等指标信息。

查看 Model API 监控数据

操作步骤

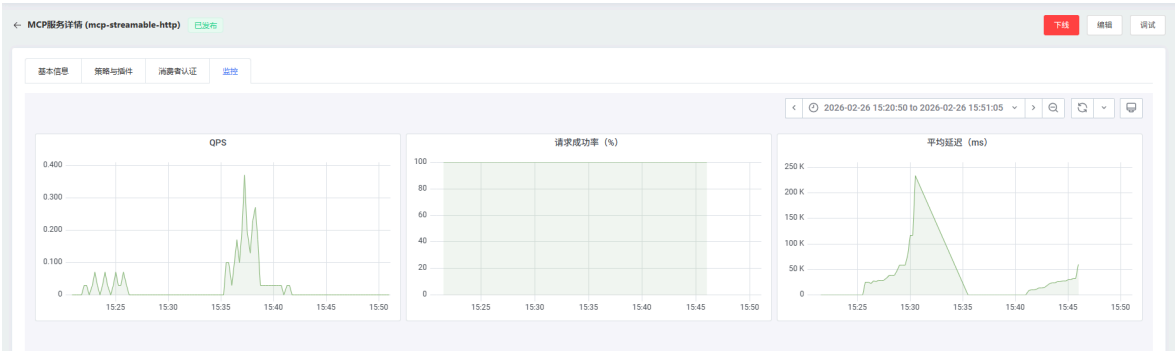
1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏“AI网关”下，点击 "实例"菜单。
3. 在实例列表页面，点击实例ID或者实例名称进入实例详情页。
4. 进入实例详情页，在左侧导航栏中，选择 "Model API"，点击列表中的“API名称”，进入API详情页。
5. 点击“监控”Tab页，可看到该API最近时刻的 QPS，请求成功率（%），平均延迟（ms）等指标信息。



查看 MCP 服务监控数据

操作步骤

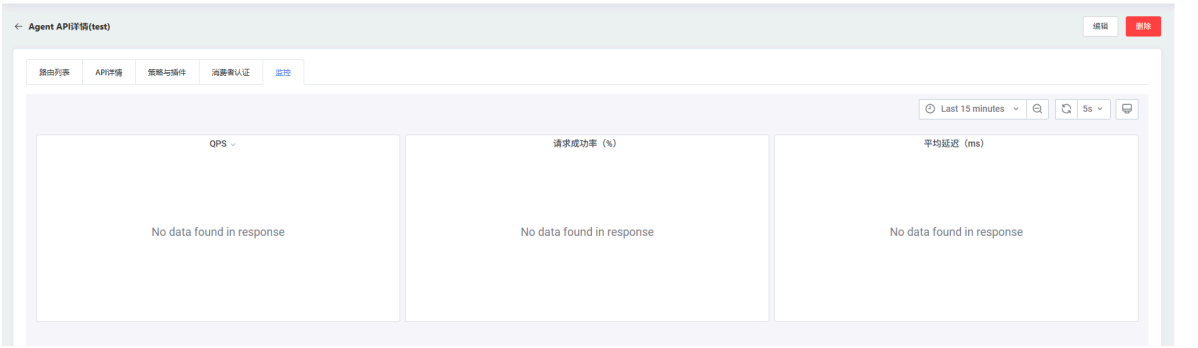
1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏“AI网关”下，点击 "实例"菜单。
3. 在实例列表页面，点击实例ID或者实例名称进入实例详情页。
4. 进入实例详情页，在左侧导航栏中，选择 "MCP 管理"，选择“MCP服务”，点击列表中的“服务名称”，进入MCP服务详情页。
5. 点击“监控”Tab页，可看到该MCP服务最近时刻的 QPS，请求成功率（%），平均延迟（ms）等指标信息。



查看 Agent API 监控数据

操作步骤

1. 登录云原生 API 网关控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏“AI网关”下，点击 "实例"菜单。
3. 在实例列表页面，点击实例ID或者实例名称进入实例详情页。
4. 进入实例详情页，在左侧导航栏中，选择 "Agent API"，点击列表中的“API名称”，进入API详情页。
5. 点击“监控”Tab页，可看到该API最近时刻的 QPS，请求成功率（%），平均延迟（ms）等指标信息。


Agent API详情(test)
编辑
删除
路由列表
API详情
策略与插件
消费者认证
监控
Last 15 minutes
5s
QPS
No data found in response
请求成功率（%）
No data found in response
平均延迟（ms）
No data found in response

通过HTTP API访问MSE Nacos中的服务

前提条件

1. 已创建MSE Nacos服务，具体操作请参见 创建Nacos引擎。
2. 微服务demo已注册至MSE Nacos，具体操作可参考 注册配置中心快速接入示例。
3. 已经创建云原生API网关实例，具体操作请参见 创建网关实例。

方案概览

1. 创建HTTP API：配置API的基本信息。
2. 创建路由：
   - 配置路由规则：定义API与MSE Nacos服务的映射关系，确保请求能够准确地被路由到正确的目标服务。
   - 所属实例：选择对应的云原生网关实例，确保路由配置在实际部署时能够正确应用。
   - 关联服务：将服务实例与路由规则关联，实现服务的动态发现与调用，提升系统的灵活性和扩展性。
3. 路由调试：通过模拟请求和分析响应，验证云原生API网关与MSE Nacos服务的集成是否符合预期，确保服务调用的高效性和稳定性。

步骤一：创建HTTP API

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 在API页面单击"创建API">类型选择"HTTP API">单击"创建"，在创建HTTP API页面填写API名称和描述。

步骤二：创建路由

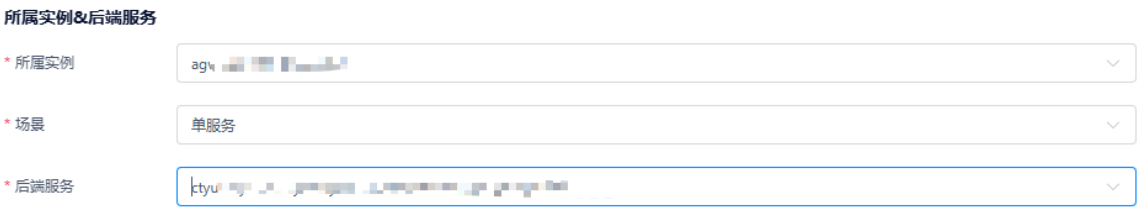
单击目标API名称，进入API详情页，单击创建路由，在创建路由面板，配置相关参数。

1.配置路由基本信息

必填路由名称和路径，可为该路由绑定域名。

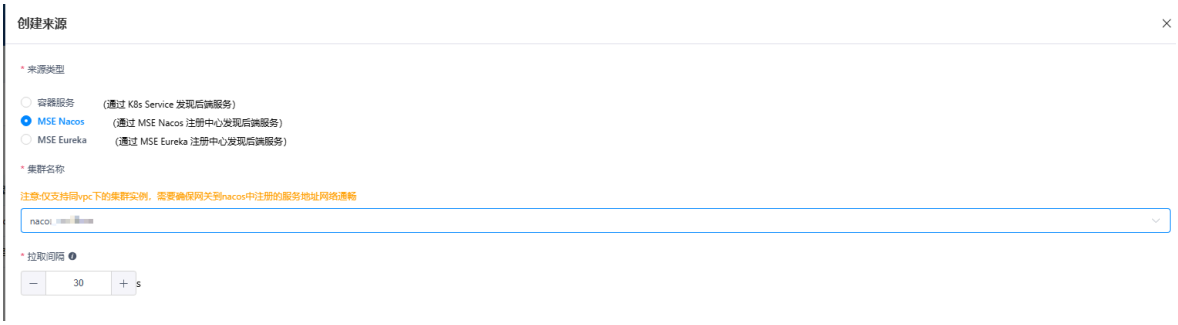
2.所属实例&后端服务

选择对应的云原生API网关实例，使用场景选择单服务。

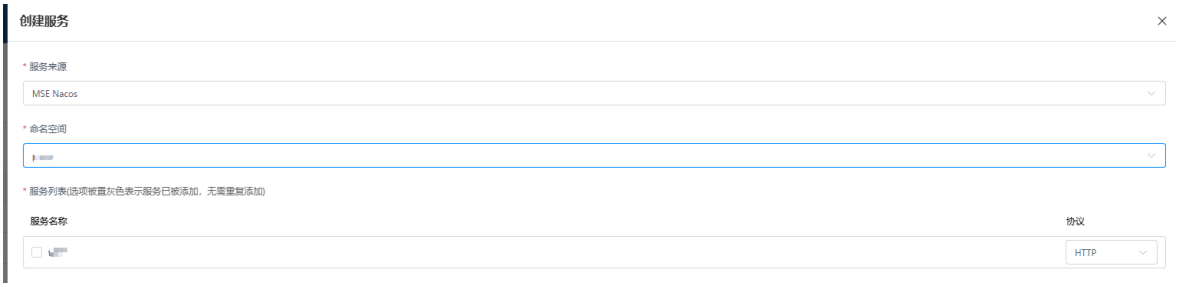


3.关联服务

1. 创建服务来源：进入 所属实例，选择"服务">"服务来源"，点击"创建来源"，来源类型选择"MSE Nacos"，并选择"集群"。



2. 创建服务：选择"服务">"服务管理"，点击"创建服务"，服务来源选择 "MSE Nacos"，选择对应的命名空间和服务。



c.保存并发布路由。

步骤三：路由调试

1. 路由发布成功后，单击目标路由操作列的"调试"。
2. 在调试面板中输入相关接口参数，并单击"发送请求"，在右侧即可看到服务接口的返回结果。

通过HTTP API访问CCE应用里注册的K8s Service

前提条件

1. 已具备云容器引擎CCE实例，参见 创建一个CCE应用集群。
2. 部署微服务demo到云容器引擎CCE实例，参见 创建工作负载及服务 或者 使用容器镜像服务发布容器应用。
3. 已经创建云原生API网关实例，具体操作请参见 创建网关实例。

方案概览

1. 创建HTTP API：配置API的基本信息。
2. 创建路由：
   - 配置路由规则：定义API与CCSE服务的映射关系，确保请求能够准确地被路由到正确的目标服务。
   - 所属实例：选择对应的云原生网关实例，确保路由配置在实际部署时能够正确应用。
   - 关联服务：将服务实例与路由规则关联，实现服务的动态发现与调用，提升系统的灵活性和扩展性。
3. 路由调试：通过模拟请求和分析响应，验证云原生API网关与CCSE服务的集成是否符合预期，确保服务调用的高效性和稳定性。

步骤一：创建HTTP API

1. 登录云原生API网关控制台。
2. 在左侧导航栏，选择 "API"，并在顶部菜单栏选择地域。
3. 在API页面单击"创建API">类型选择"HTTP API">单击"创建"，在创建HTTP API页面填写API名称和描述。

步骤二：创建路由

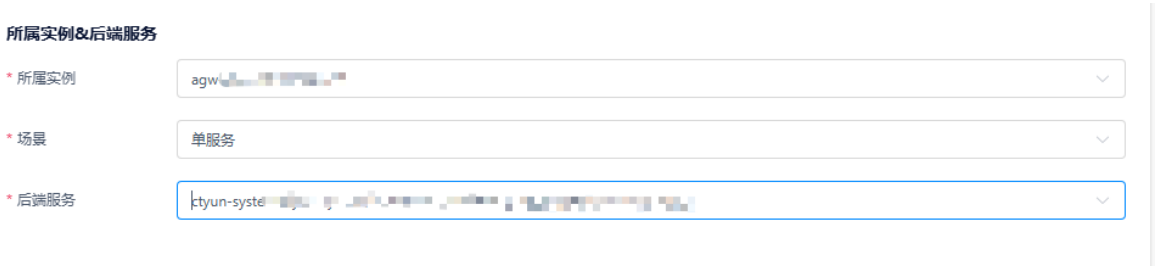
单击目标API名称，进入API详情页，单击创建路由，在创建路由面板，配置相关参数。

1.配置路由基本信息

必填路由名称和路径，可为该路由绑定域名。

2.所属实例&后端服务

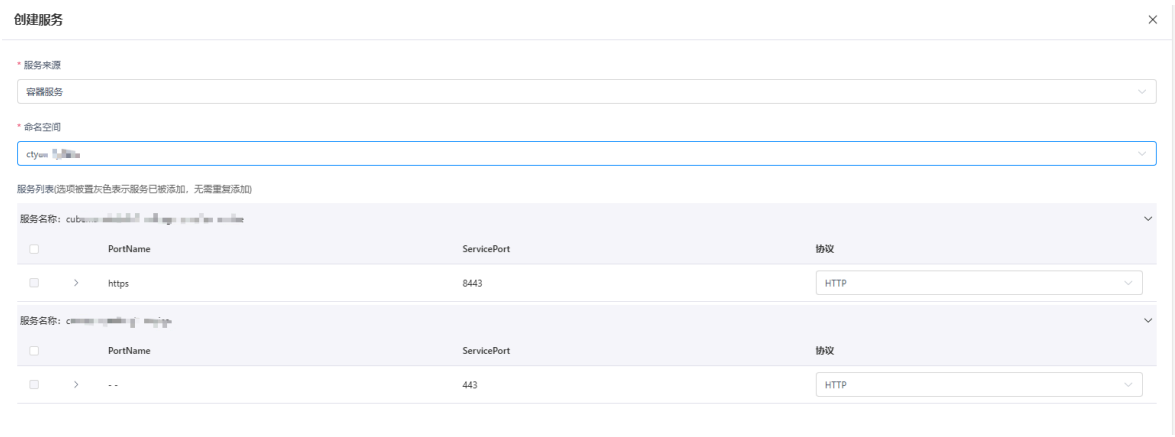
选择对应的云原生API网关实例，使用场景选择单服务。



3.关联服务

1. **创建服务来源：**进入 所属实例，选择"服务">"服务来源"，点击"创建来源"，来源类型选择"容器服务"，并选择"容器集群"。 如果需要监听K8s Ingress，则填写监听的命名空间标签选择器。

2. **创建服务**：选择"服务">"服务管理"，点击"创建服务"，服务来源选择 "容器服务"，选择对应的命名空间和服务。



c.保存并发布路由。

步骤三：路由调试

1. 路由发布成功后，单击目标路由操作列的"调试"。
2. 在调试面板中输入相关接口参数，并单击"发送请求"，在右侧即可看到服务接口的返回结果。

使用云原生API网关实现蓝绿、金丝雀发布及AB实验

前提条件

- 了解蓝绿部署、A/B测试以及金丝雀发布机制。详细信息，请参见 服务发布策略。
- 已具备云容器引擎CCE实例，参见 创建一个CCE应用集群。
- 已完成云原生API网关创建，具体操作，请参见 创建网关实例。

部署服务

容器内部署服务

首先在容器内部署两个版本的服务，两个版本的Deployment分别挂到reviews-v1和reviews-v2服务下：

```
apiVersion: v1
kind: Service
metadata:
 name: reviews-v1
 labels:
  workloadKind: Deployment
  workloadName: reviews-v1
spec:
 ports:
  - port: 9080
```

```
    targetPort: 9080
    name: http
  selector:
   app: reviews
   version: v1
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: reviews-v1
spec:
  replicas: 1
  selector:
   matchLabels:
    name: reviews-v1
  template:
   metadata:
    labels:
     app: reviews
     version: v1
     name: reviews-v1
     source: CCSE
   spec:
    containers:
     - name: reviews
       image: 'registry-vpc-crs**************************************************'
       imagePullPolicy: IfNotPresent
       env:
        - name: LOG_DIR
          value: "/tmp/logs"
       volumeMounts:
        - name: tmp
          mountPath: /tmp
        - name: wlp-output
          mountPath: /opt/ibm/wlp/output
    volumes:
     - name: wlp-output
       emptyDir: {}
     - name: tmp
       emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: reviews-v2
  labels:
```

```
  workloadKind: Deployment
  workloadName: reviews-v2
spec:
 ports:
  - port: 9080
    targetPort: 9080
    name: http
 selector:
  app: reviews
  version: v2
---
apiVersion: apps/v1
kind: Deployment
metadata:
 name: reviews-v2
spec:
 replicas: 1
 selector:
  matchLabels:
    name: reviews-v2
 template:
  metadata:
    labels:
      app: reviews
      version: v2
      name: reviews-v2
      source: CCSE
  spec:
    containers:
      - name: reviews
        image: 'registry-vpc-crs-************************************************'
        imagePullPolicy: IfNotPresent
        env:
          - name: LOG_DIR
            value: "/tmp/logs"
        volumeMounts:
          - name: tmp
            mountPath: /tmp
          - name: wlp-output
            mountPath: /opt/ibm/wlp/output
    volumes:
      - name: wlp-output
        emptyDir: {}
      - name: tmp
        emptyDir: {}
```

最佳实践

添加服务到网关

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "实例" ，并在顶部菜单栏选择地域。
3. 选择目标实例后，在左侧导航栏，选择 "服务" > "服务来源"。
4. 点击 "创建来源"，来源类型选择 "容器服务"，选择对应的容器集群，然后单击 "确定"。
5. 进入 "服务管理" > "创建服务"，选择服务所在的命名空间。
6. 选择刚部署的reviews-v1和reviews-v2服务，添加为网关服务，然后单击 "确定"。

创建路由

1. 登录 云原生API网关控制台。
2. 在左侧导航栏，选择 "API" ，并在顶部菜单栏选择地域。
3. 单击 "创建API"，类型选择"HTTP API"即可。
4. 进入"API详情页" > "创建路由"，填写相应配置，所属实例&后端服务选择上述步骤所配置的服务。

蓝绿部署

蓝绿部署需要按照服务当前版本所占用的资源状况为服务新版本申请同样的资源规格，部署完毕之后将流量整体切换到服务新版本。

创建一条路由规则，先将版本v1和v2比例设置为50%、通过蓝绿发布将流量从v1整体切换至v2，仅需要修改之前创建的路由规则中的目标服务，将v2比例设置为100%，则可以做到请求全部到v2中。

A/B测试

A/B测试是基于用户请求的元信息将流量路由到新版本，也就是可以根据请求内容来动态路由。在本示例中，希望End-User头部为jason时访问reviews-v2版本。

创建两条路由规则：

- 匹配Path为/version的请求访问服务版本v1。
- 匹配Path为/version，且End-User头部为jason的请求访问服务版本v2。

金丝雀发布

金丝雀发布允许引流一小部分流量到新版本，待验证通过后，逐步增加流量，直至完全切换，期间可伴随着新版本的扩容和旧版本的缩容操作，达到资源利用率最大化。

创建一条路由规则，在目标服务中按照权重将流量转发至新旧版本。创建路由时选择多服务路由，选择路由目标为reviews-v1和reviews-v2，并配置流量比例，例如，v1权重设置为80%，v2权重设置为20%，这样在请求时会按照指定的比例访问到对应版本的服务。

服务发布策略

蓝绿部署

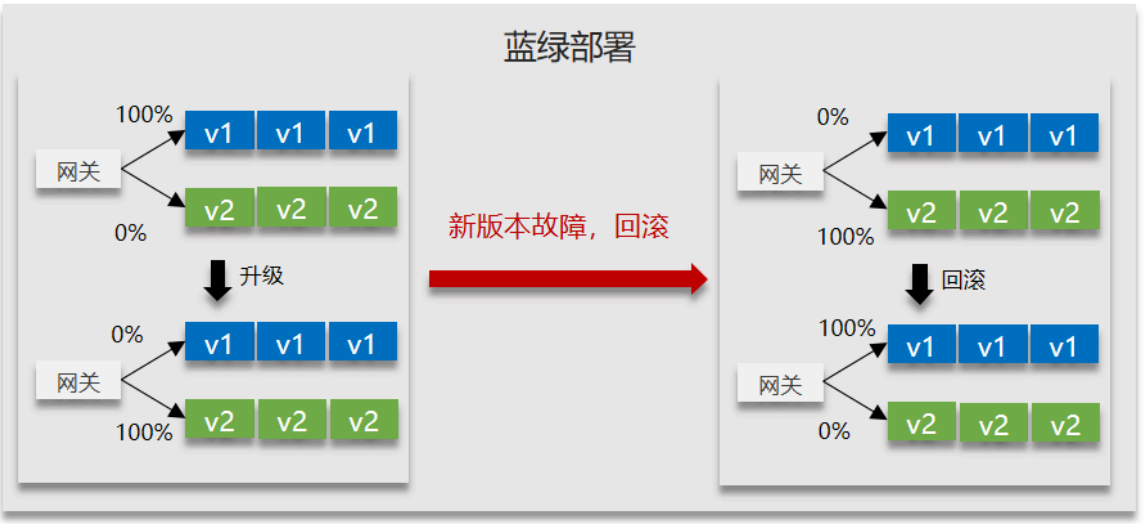
概念

蓝绿部署是一种通过冗余资源保障服务高可用的发布策略。它会同时维护两套完全相同的生产环境（蓝色旧版和绿色新版），平时仅由蓝色环境承载流量，绿色环境作为热备待命。发布时只需将流量无缝切换至绿色环境，旧版则转为备用状态。

这种设计既确保了发布过程零停机，又能在新版本故障时实现秒级回切，大幅降低业务风险。由于双环境资源对等，无需担心容量不足问题，但需要额外支付一倍的资源成本。

蓝绿部署特别适合对稳定性要求严苛的关键业务系统，通过牺牲部分资源成本换取绝对的发布安全性和快速回滚能力。

如下图所示，某服务旧版本为v1，对新版本v2进行冗余部署。版本升级时，将现有流量全部切换为新版本v2。当新版本v2存在问题或者发生故障时，可以快速切回旧版本v1。



优点

- 部署结构简单，运维方便。
- 服务升级过程操作简单，周期短。

缺点

- 资源冗余，需要部署两套生产环境。
- 新版本故障影响范围大。

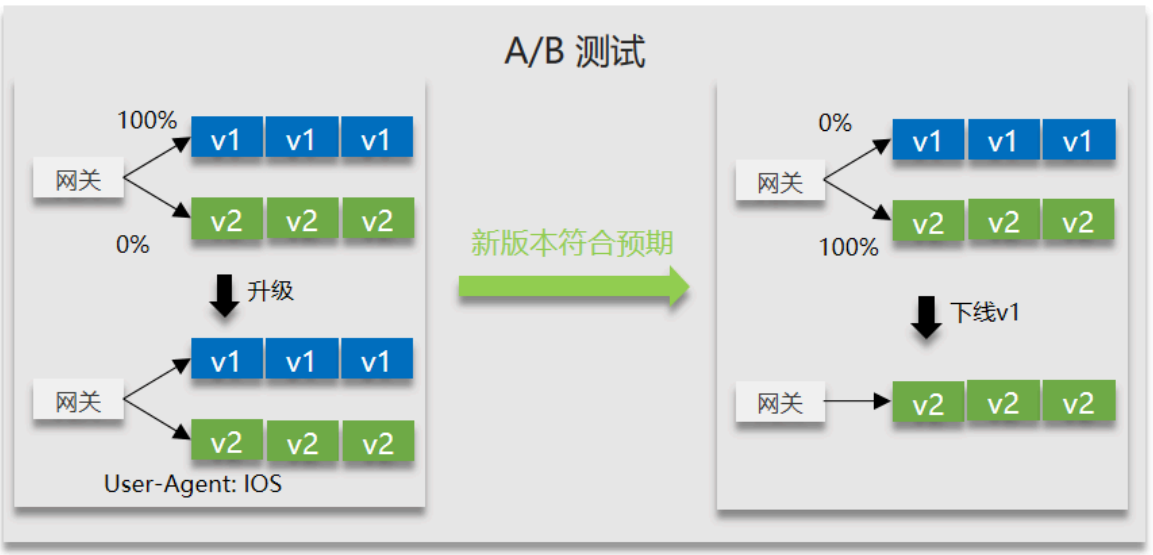
A/B测试

概念

A/B测试是一种通过对比实验优化产品决策的技术方法。它会将用户流量随机分配到两个或多个不同版本（A版和B版）的服务或界面中，在相同运行环境下收集各版本的用户行为数据。这是一种基于请求内容匹配的灰度发布策略，只有匹配特定规则的请求才会被引流到新版本，常见的做法包括基于HTTP Header和Cookie。

这种方案既能验证新功能/改版的实际效果，又能有效控制产品迭代风险。由于采用实时分流机制，整个过程无需停机切换，且可以随时终止实验恢复原状。

如下图所示，某服务当前版本为v1，现在新版本v2要上线。希望安卓用户可以尝鲜新功能，其他系统用户保持不变。通过在监控平台观察旧版本与新版本的成功率、RT对比，当新版本整体服务符合预期后，即可将所有请求切换到新版本v2，最后为了节省资源，可以逐步下线到旧版本v1。



优点

- 可以对特定的请求或者用户提供服务新版本，新版本故障影响范围小。
- 需要构建完备的监控平台，用于对比不同版本之间请求状态的差异。

缺点

- 仍然存在资源冗余，因为无法准确评估请求容量。
- 发布周期长。

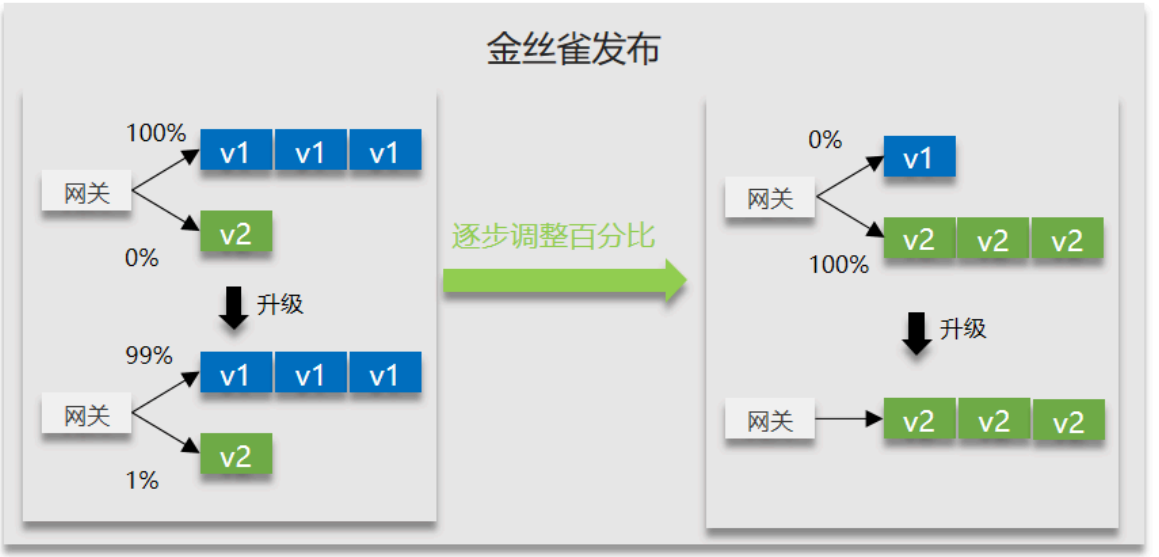
金丝雀发布

概念

金丝雀发布（Canary Release）是一种渐进式服务发布策略，通过小范围流量试用来验证新版本稳定性。其核心原理类似煤矿中用金丝雀探测危险——先部署少量新版本实例（如5%流量），其余流量仍由旧版本处理。通过

监控新版本的错误率、延迟等关键指标，确认无异常后再逐步扩大新版本流量比例（如30%→50%→100%），最终完成全量升级。若期间发现问题，则立即回滚至旧版本，将影响范围控制在最小。

如下图所示，某服务当前版本为v1，现在新版本v2要上线。为确保流量在服务升级过程中平稳无损，采用金丝雀发布方案，逐步将流量从老版本迁移至新版本。



优点

- 按比例将流量无差别地导向新版本，新版本故障影响范围小。
- 发布期间逐步对新版本扩容，同时对老版本缩容，资源利用率高。

缺点

- 流量无差别地导向新版本，可能会影响重要用户的体验。
- 发布周期长。