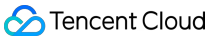
Service Registry and Governance

AI Gateway

Product Documentation

Copyright Notice

Trademark Notice



Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

AI Gateway

AI Gateway Overview

Last updated：2026-05-07 17:26:54

AI gateway is a new-generation gateway product launched by Tencent Cloud Intelligent Gateway for large models and intelligent scenarios. It focuses on solving core issues faced by enterprises when enterprises access, schedule, and manage multiple AI models, such as complex protocols, governance difficulties, uncontrollable costs, and high barriers to transforming existing businesses.
AI gateway serves as the traffic entry and governance hub for enterprise intelligent architectures, enabling enterprises to efficiently, securely, and economically integrate and utilize AI capabilities through unified protocol adaptation, intelligent routing scheduling, and comprehensive observability capabilities, thereby accelerating business innovation and intelligent transformation.

Product Features

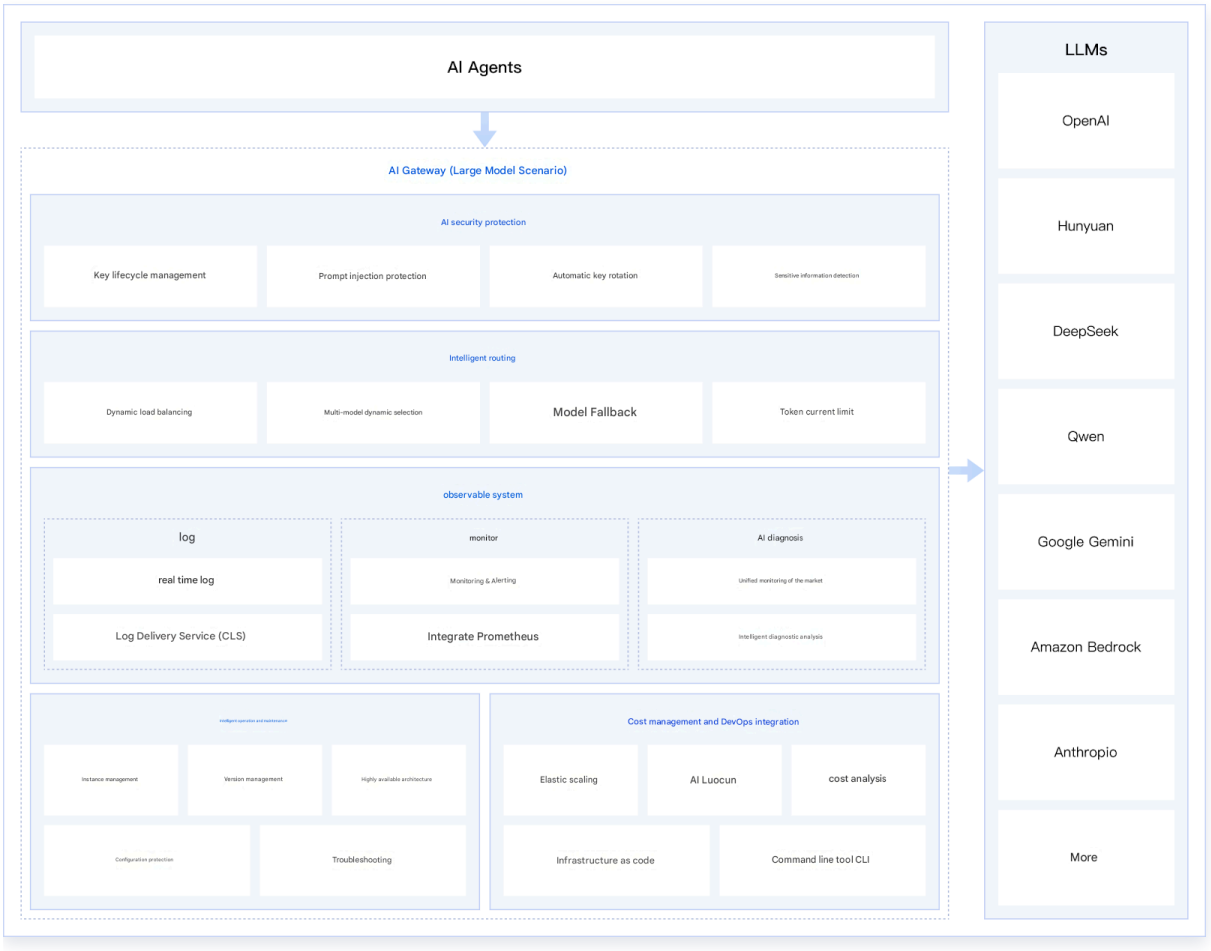
- **Intelligent Model Governance:** Unified access and intelligent scheduling of Tencent Cloud Hunyuan, open-source models, and third-party commercial models. Through load balancing (CLB), circuit breaking and degradation, and cost optimization policies, it achieves the optimal balance of performance, stability, and cost.
- **Rapid AI Transformation for Legacy Business Systems:** It features a powerful built-in protocol conversion engine that supports bidirectional conversion between AI ecosystem protocols such as MCP and OpenAI and traditional business protocols like HTTP/gRPC. This enables legacy business systems to quickly acquire AI capabilities, effectively protecting enterprises' existing IT investments.
- **Comprehensive End-to-End Security and Compliance:** Build a multi-layered security protection system spanning from access authentication and parameter filtering to Data Masking (DMask). Integrated with capabilities such as WAF and DDoS protection (Anti-DDoS), it ensures AI applications operate compliantly, securely, and reliably.
- **Enterprise-Grade High Availability Assurance**: Adopts a multi-AZ high-availability deployment architecture, supports automatic failover and elastic scaling of instances, ensuring service availability.

Business Scenarios

Scenario 1: Unified Governance and Intelligent Scheduling of Multiple Models

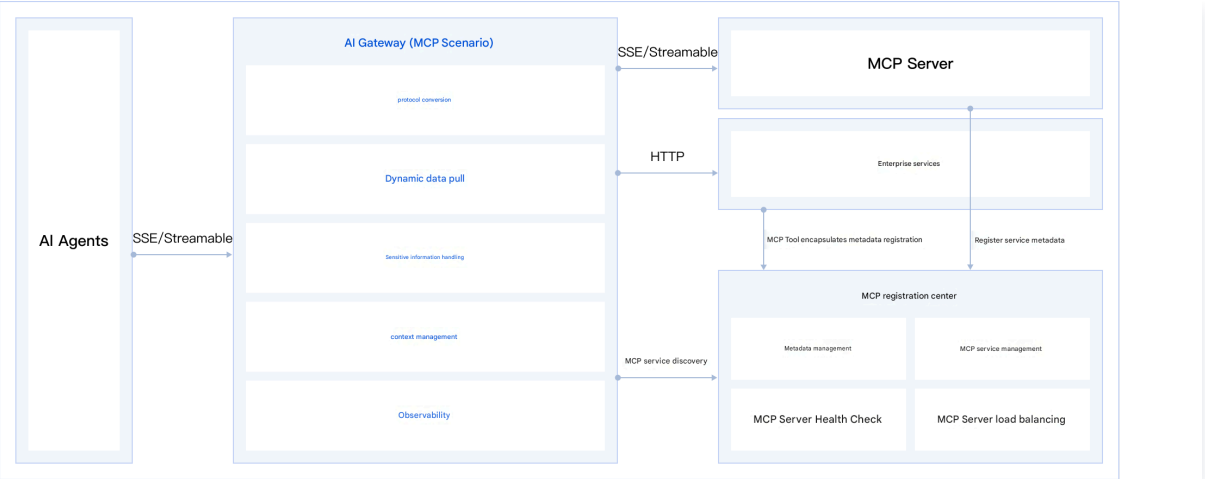
- **Core Issues:** Enterprises need to use multiple AI models simultaneously but lack a unified platform, leading to chaotic model usage, inefficient resource allocation, and uncontrollable costs.
- **Solution**: AI gateway serves as a unified entry point, enabling visual access and management of multiple models. Through intelligent routing policies, it automatically selects the optimal model based on factors

such as request content, model performance, and cost, and incorporates rate limiting and circuit breaking mechanisms to ensure service stability, achieving cost reduction and efficiency improvement.



Scenario 2: AI Transformation of Legacy Business Systems

- **Core Issues**: Traditional enterprises' legacy systems have outdated technology stacks; direct transformation requires significant investment and high risks, and they cannot adapt to modern AI application invocation protocols.
- **Solution**: The AI gateway automatically packages standard APIs provided by legacy business systems into standardized tools (MCP Tools) that AI applications can call, through its protocol conversion engine. This enables the "zero-code transformation" of business capabilities into AI capabilities. Developers can quickly build intelligent applications without needing to focus on underlying integration details.



Features

Feature	Note
Unified Protocol Access	100% compatible with the open-source gateway ecosystem, fully adapts to standard AI protocols such as MCP, OpenAI, and SSE, provides seamless conversion for traditional protocols like RESTful and gRPC, and enables a single gateway to handle all traffic.
Model Service Management	Provides full lifecycle management for model services, supports configuring keys and API endpoints for multiple model vendors, and offers model-level traffic control, Fallback disaster recovery, and fine-grained monitoring.
Intelligent Routing & Orchestration	Supports intelligent routing based on policies such as content semantics, cost, and performance. Can orchestrate and chain multiple model invocations or business APIs to complete complex tasks.
Fine-grained Traffic Governance	Provides capabilities such as rate limiting, circuit breaking, and degradation across multiple dimensions from consumers and APIs to models, ensuring the stability of backend services and model APIs.
Out-of-the-box Security Protection	Integrates security capabilities such as authentication and authorization, access control, sensitive information desensitization, and replay attack prevention, provides enterprise-grade security protection, and meets compliance requirements.
End-to-end Observability	Provides end-to-end tracing from user requests to model responses, monitors multi-dimensional metrics such as API call latency, Token consumption, and model costs, supports intelligent diagnosis and alarms, and facilitates Ops and cost optimization.

Fine-grained Permission Management	Implements secure isolation and convenient sharing of AI capabilities across different teams and projects through a multi-level permission model based on consumers and consumer groups, supporting platform-based operations.

Version Lifecycle Management

Last updated：2026-05-07 17:26:54

To ensure the stability and security of user services, this document introduces the lifecycle management mechanism for editions of AI Gateway.

Version Number Specification

AI Gateway uses a three-digit version number, that is, a.b.c, where each part of the version number has the following meaning:

- a.b: aligns with the first two digits of the version number in the Kong open source community.
- c: version number for feature development and bug fixes. Increment this version number when feature development or bug fixes occur.

Example: version number 3.9.1 indicates the feature version based on the open-source Kong 3.9 version.

Key Lifecycle Milestones

AI Gateway provides up to 27 months (18+9) of service support for each released product version. The support lifecycle is divided into the following three phases:

Milestone	Description	Action Required
GA (General Availability)	It means that the current version is fully deliverable to live network customers. During this period, the platform provides patches and service support for the version in this stage.	No additional attention is required. Use the product as normal.
EOM (End of Marketing)	It refers to the time when the creation of new instances for the current version is stopped across the entire network. The version generally goes to the EOM stage 18 months after it goes to the GA stage. During this period, the platform provides patches and service support for the version in this stage.	Develop an upgrade plan to upgrade to the latest stable version before the edition EOS.
EOS (End of Service & Support)	It refers to the time when service is stopped for the current version. At this point, the version is expired.	You must take immediate Ops action to upgrade your instance to the latest stable

	The version generally goes to the EOS stage 27 months after it goes to the GA stage. The product will no longer provide technical support other than version upgrades, nor will it commit to SLA.	version. Before this upgrade, your instance may be at high risk, facing potential system attacks or business stability risks. You must act as soon as possible.

patches and service support

Patch Scope

New features provided by the platform, functional defect fixes, community-contributed feature integrations, and security risk fixes.

Service Support Scope

- Instance creation: supports creating version instances in the GA phase.
- Upgrade and Ops support: provides the feature of version upgrades, and offers support for troubleshooting and failure recovery.
- After-sales support: provides Q&A, online guidance, troubleshooting, and debugging services. However, for instances of expired versions, the platform does not guarantee the quality or effectiveness of technical support, and the SLA may be affected due to non-compliance with best practices.

Expired Version Risk

- We cannot create new instances for expired versions.
- The platform no longer provides patch services for expired versions.
- The quality and effectiveness of technical support cannot be guaranteed.
- The platform reserves the right to forcibly upgrade instances of expired versions. Before performing a forced upgrade, we will send relevant notifications via SMS, email, in-site messages, and other means at least one month in advance.

To ensure the stability of your production services, we recommend that you promptly plan and upgrade your instance versions. For product upgrade operations, see Upgrading Gateway Versions.

Quick Start
Calling Hunyuan API Through AI Gateway

Last updated：2026-05-07 17:26:54

AI gateway calling the Hunyuan API manages access to multiple AI models through a unified interface, enabling efficient, secure, and scalable AI capability integration. It simplifies the Hunyuan API invocation process and provides one-stop services including traffic control, monitoring, and billing. This article will quickly guide you through the initial configuration to experience the complete process of invoking Hunyuan large model services via AI gateway. You will sequentially configure model keys, model services, and model APIs, create caller identities (consumers and consumer groups), and finally initiate successful invocations through the gateway.

Prerequisite

1. AI gateway instances have been created. For detailed operations, see Create AI Gateway.
2. have obtained the API invocation key for the Hunyuan large model.

Operation Overview

The core operational workflow is as below. You will sequentially complete the following six steps:

1. **Create Model Key**: Securely configure the API keys required to access Hunyuan in the gateway.
2. **Create Model Service**: Add Hunyuan as an AI model provider and associate it with the key created in the previous step.
3. **Create Model API**: creates an API that provides text generation capabilities externally and binds it to the service created in step 2.
4. **Create Consumer**: Create a caller representing your own application and add identity credentials (API Key) for it.
5. **Create Consumer Group and Grant Permissions**: Group consumers and grant the group access permissions to the model API created in step 3.
6. **Obtain Address and Initiate Invocation**: Find the API access address in the console and use the consumer's credentials to initiate the invocation request.

Next, complete each step as per the detailed instructions below.

Operation Steps

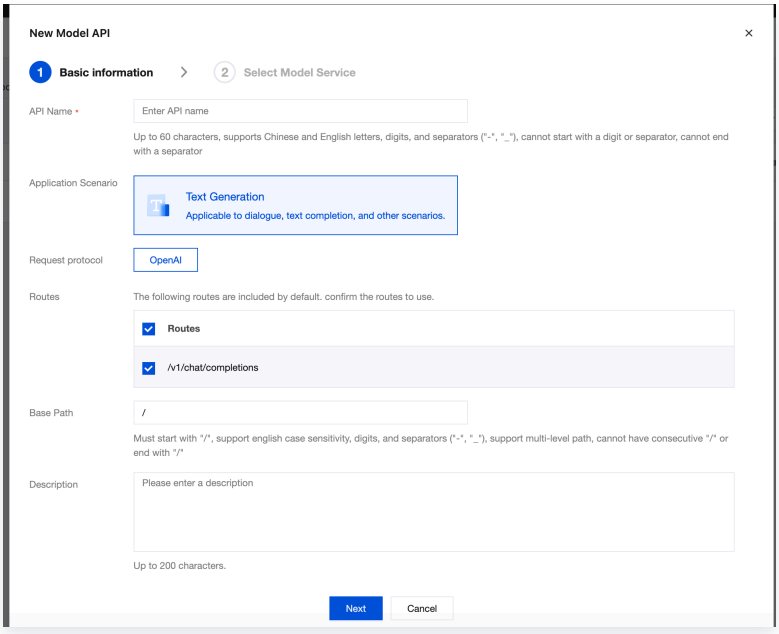
Step 1: Create Model Key

Model keys are used to securely store and manage credentials required to access third-party large model services.

1. Log in to the Microservices Platform Console, choose **Cloud-Native Intelligent Gateway** > **Instance List** in the left sidebar, select the instance you want to use, and go to the instance details.
2. In the left sidebar, click **Key Management**.
3. On the "Key Management" page, click **Create**.
4. In the "Create Key" window, configure the key information by referring to the following table.

Parameter	Filling Instructions
Key Type	Select **Model Key**.
Key Name	Custom name, such as hunyuan-key.
Generation Method	Select **Custom**.
Credential Content	Enter the API key obtained from Hunyuan official here.
Description	(Optional) Enter the description.

5. Click **OK** to complete key creation.



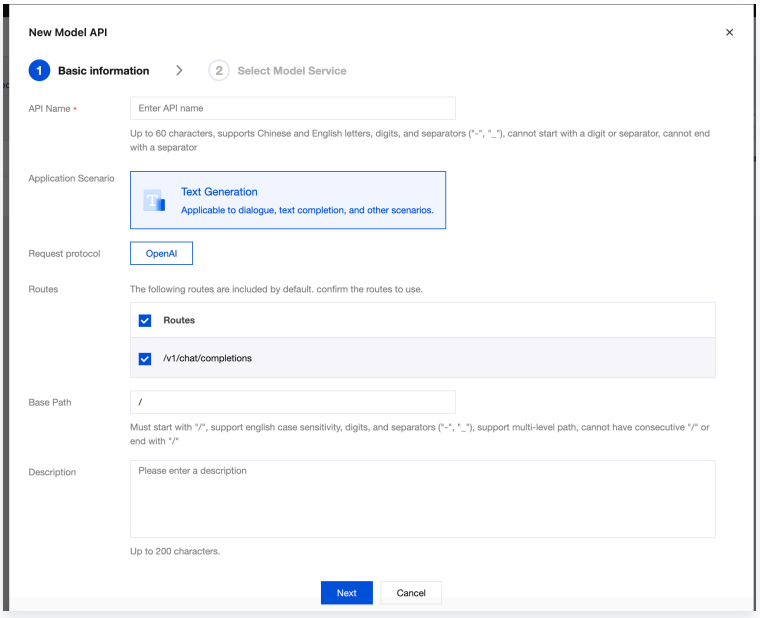
Step 2: Create Model Service

Model Service is used to encapsulate call configurations for a specific large model vendor (Hunyuan in this case).

1. In the left sidebar, click **Model Management** > **Model Service**.
2. On the "Model Service" page, click **Create**.
3. In the "Create Model Service" window, go to the **Basic Info** step and configure as follows.

Parameter	Filling Instructions
Service Name	Custom name, such as hunyuan-service.
Service Type	Fixed to **AI Model Service**.
Model Vendor	Select **Hunyuan**.
Model Protocol	Select **OpenAI-compatible**.
Service Address	The system will automatically fill in the official endpoint address for Hunyuan.
Model Key	Select the key (hunyuan-key) you created in Step 1 from the drop-down list.
Key Usage Policy	Select **Polling**.

4. Click **Next** to go to the **Select Model Policy** step.
   - Model selection method: Keep the default specified model.
   - Default model: Select a Hunyuan model from the drop-down list, such as HY 2.0 Instruct 20251111.
5. Click **OK** to complete the model service creation.



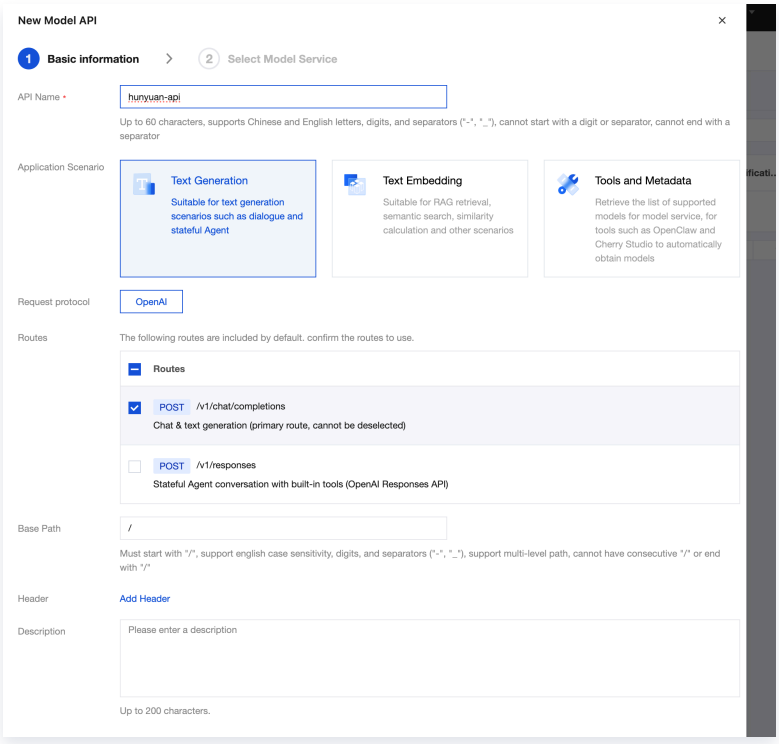
Step 3: Create Model API

Model API is the invocation endpoint exposed by the gateway. Clients access this API to utilize large model capabilities.

1. In the left sidebar, click **Model Management > Model API**.
2. On the "Model API" page, click **Create**.
3. In the "Create Model API" window, go to the **Basic Info** step and configure as per the table below.

Parameter	Filling Instructions
API Name	Custom name, such as hunyuan-api.
Usage scenario	Select **Text Generation**.
Request Protocol	Select **OpenAI**.
Route	/v1/chat/completions is selected by default. Please confirm.
Base Path	Enter a path prefix as the unique identifier for the API, such as /hunyuan. Clients will access it through this path.
Path Simplification	It is recommended to keep it enabled so that the backend receives a concise path.

4. Click **Next** to go to the **Select Model Service** step.
   - Service Type: Select **Single Model Service**.
   - Select service: Select the service you created in **Step 2** (hunyuan-service) from the drop-down list.
5. Click **OK** to complete the model API creation. The system will automatically generate an access route.



Step 4: Create Consumer

Consumers represent the client identities that call the API and require configuration of authentication credentials.

1. In the left sidebar, click **Consumer Management > Consumers**.
2. On the "Consumers" page, click **Create**.
3. In the "Create Consumer" window, configure the settings by referring to the following table.

Parameter	Filling Instructions
Consumer Name	Custom name, such as my-app.
Consumer Group	This can be left unselected for now and associated in the consumer group later.
Select a key	Click **Create Key** to generate an API Key for authenticating this consumer when the gateway is called. Note down the generated Key.

4. Click **OK** to complete the consumer creation.

Step 5: Create Consumer Group and Grant Permissions

Consumer Group is used to group consumers and uniformly grant them access permissions to the model API.

1. In the left sidebar, click **Consumer Management > Consumer Group**.
2. On the "Consumer Group" page, click **Create**.
3. Configure basic information for the consumer group, and in the "Consumers" option, associate the consumer (my-app) created in **Step 4**.
4. After creation, locate the group, switch to the "Authorized Model APIs" tab, and click the **Add Authorization** button.
5. On the authorization page, grant access to the model API (hunyuan-api) created in **Step 3** to this consumer group.

Step 6: Obtain the Access Address and Initiate the Invocation

After all configurations are completed, you can invoke the large model via the gateway address.

1. Obtain the gateway entry address: Go to **Basic Information > Instance Information**, and in the **Network Configuration** tab, view the "public network CLB address" or "private network Private Link address".
2. Obtain the route access address:
   - Go to **Model Management > Model API**, and click the name of the API you created in Step 3.
   - Click the **Route Management** tab and copy the "Request Path". The complete access address format is: `protocol://gateway entry address/request path`.
3. Initiate the invocation: Use the curl command or any HTTP client to refer to the following example and initiate the request. Replace <access address> with the full URL obtained in the previous step, replace <

API_KEY> with the API Key created for the consumer in Step 4, and replace <MODEL_NAME> with the model selected in Step 2.

```
curl -i -X POST <access address> \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer < API _KEY>" \
  -d '{
    "model": "<MODEL_NAME>",
    "messages": [
      {"role": "user", "content": "Hello, please introduce yourself."}
    ],
    "stream": false
  }'
```

4. Check the results: If all configurations are correct, you will receive a JSON-formatted response from the Hunyuan large model.

Operation Guide
Gateway Management
Create New AI Gateway

Last updated：2026-05-07 17:26:54

Scenarios

This article describes how to create an AI gateway instance through the Microservices Platform console.

Operation Steps

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. Click **Create** to go to the Cloud Native Intelligent Gateway creation page.

Parameter	Required	Description
Gateway Type	Yes	Supported types: **AI Gateway** and **Cloud Native Gateway**.
Billing Modes	Yes	**Yearly/Monthly subscription** and **Pay-as-you-go** are supported. If your gateway instance needs to be used for more than one month, you are advised to use the **prepaid (yearly/monthly subscription)** mode. For details about the price, see Pricing.
Region.	Yes	Select the region closest to your deployed business.
Product Edition	Yes	Only the Standard Edition is supported.
Name	Yes	Enter the name of the current AI gateway. The maximum length is 60 characters. Uppercase and lowercase letters in Chinese and English, hyphens (-), and underscores (_) are supported.
Description	No	Enter the description of the current AI gateway.
Resource Tag	No	Used for resource management by type. For detailed usage methods, see Tag Management.
Node	Yes	Select the Tencent Cloud VPC network where the current AI gateway is

Network		deployed.
Node specification	Yes	An AI gateway cluster consists of multiple nodes. Select the specification for each node here.
Node Quantity	Yes	An AI gateway cluster consists of multiple nodes. Select the number of nodes here. • When the number of nodes is 2, only Random AZ can be selected. • When the number of nodes is greater than or equal to 3, Random AZ and Specified AZ can be selected. (At least two AZs need to be selected, and up to three AZs can be selected.)
Deployment Architecture	Yes	With dual-AZ or multi-AZ deployment in the same city, a high-availability registered gateway is provided, which supports same-city multi-active by default.
Public Load Balancer	No	Enabling public network access will charge fees. For details about the price, see Public Network Traffic Fees. • Billing Mode: **By traffic usage** and **By bandwidth** are supported. • Maximum bandwidth: Supports a range of 1 – 2048 Mbps. • AZ Type: **Single-AZ** and **Multi-AZ** are supported. When Multi-AZ is selected, you need to select a primary AZ and a secondary AZ. When the primary AZ is faulty, Cloud Load Balancer (CLB) can automatically switch to the secondary AZ within a short time and restore services to ensure gateway availability.
Log Collection	No	• Enable real-time log service: Cloud Native API Gateway provides real-time logging and simple search capabilities free of charge by default. To persistently store logs for troubleshooting, audit, and other scenarios, you are advised to enable Cloud Log Service (CLS). • Enable CLS: Gateway log shipping is enabled for you, and log analysis is provided. Before selecting this option, confirm that you have activated CLS. The service is provided by CLS and incurs fees. For detailed billing items, see Billing Details.

3. Click **Buy Now** and wait patiently until the instance is created.

Result Verification

Return to the gateway instance list page and view information and status of the created gateway instance. If the displayed gateway instance information is the same as that specified during creation and the status is Running, the gateway instance is created successfully.

Must-Knows

- Creating an AI Gateway requires the account to have the ApiGateWay_QCSRole role, and the role must include the `QcloudAccessForApiGateWayRoleInCloudNativeAPIGateway` policy. If you do not have the relevant role and policy, please complete the authorization in the permission pop-up window that appears during the creation process.
- The AI Gateway logging feature relies on Tencent Cloud Log Service (CLS). Ensure that you have activated CLS. Otherwise, go to the CLS console to activate the service before creating an AI Gateway.

Upgrading the Gateway Edition

Last updated：2026-05-07 17:26:54

Overview

If the current gateway version has entered the decommissioning phase, to ensure the stable operation of the gateway instance, it is recommended that you upgrade your gateway to a stable version via the console.

> **Note:**
> To ensure a successful upgrade, please do not perform any modification operations on the gateway instance before or during the upgrade process. During the upgrade, write access to the Gateway Console will be disabled. Please plan your upgrade schedule accordingly.

Version upgrade

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. If the gateway version column shows **Upgrade Available**, under **Actions** > **More**, click **Upgrade Gateway Version**. On the **Instance Information** > **Basic Information** card, find the product version and click **Upgrade**.
3. Click **OK** to enter the version upgrade process.
4. An upgrade environment check will be performed before the upgrade. You must ensure that all check items pass before proceeding with the upgrade. If any check fails, you can fix the issue and then click **Recheck** to re-run the environment check. The check items are as follows:

Check Item	Pass Condition	Remediation Action
Instance State	The status is Running.	Wait for the instance modification to complete, and then perform a recheck after its status becomes Running.
Backend Service State	No service has a backend pointing to the gateway admin address.	Delete the service or modify the service address, and then perform a recheck.
AS State	All AS rules are disabled.	Disable the enabled AS rules, and then perform a recheck.
Number of	Sufficient subnet IP addresses	Go to the VPC, release the subnet IP addresses, and then perform a recheck.

Subnet IP Addresses		
Node Weight	Consistent node weights	Go to **Instance Details > Basic Information > Deployment Architecture**, manually adjust the node weights, and then perform a recheck after the node weights are made consistent.

5. Confirm the instance ID and instance name, select the target version for upgrade, read and select the gateway upgrade instructions, click **Confirm** to start the upgrade. The entire upgrade process takes time depending on the current number of nodes in the instance. It may take longer if there are many nodes. You can check the task status in the status bar in the gateway list.

View Upgrade Task Progress

1. Click **Cloud Native Intelligent Gateway** > **Instance List**, find the target gateway, click **Status**, click **View Task Status**.
2. The upgrade adopts a canary policy, divided into two phases, **Version Upgrade** and **Resource Reclamation**. Version Upgrade is primarily used to configure component resources for the upgrade, during which nodes may be added. After the version upgrade is completed, resource reclamation is required to destroy excess nodes generated during the upgrade process. Therefore, you need to manually confirm and agree to perform resource reclamation, after which the gateway will automatically trigger the old resource reclamation process.

Cancel Upgrade

If business abnormalities are detected during the upgrade process, the upgrade can be revoked.

1. Click **Cloud Native Intelligent Gateway** > **Instance List**, find the target gateway, click **Status**, click **View Task Status**.
2. Click **Cancel Upgrade**. The gateway will initiate the upgrade cancellation task to roll back traffic and ensure production stability.

Viewing Gateway Details

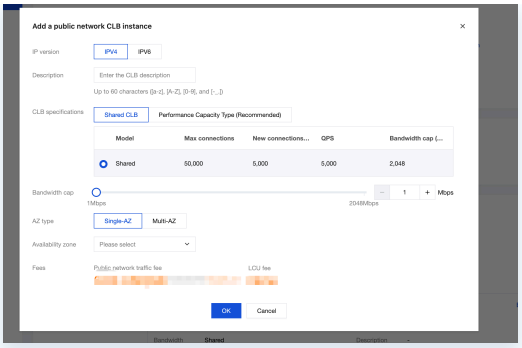
Last updated：2026-05-07 17:26:54

This document describes how to view the basic information and access address of a gateway instance after it is created.

Operation Steps

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the newly created instance to go to the gateway instance basic information page and view the basic information of the instance.
   - **Instance Information:** You can perform the following operations in the Basic Information section:
     - Click **Edit** in the upper right corner of the Basic Information module to modify the instance name and description.
     - Click the **edit** icon next to Public Network Billing Mode to modify the public network billing mode.
     - Click the **edit** icon next to Tag to modify the bound tag information of the gateway instance.
   - **Node Configuration:**
     - Node Specification: the specification of each node in the gateway instance.
     - Node Quantity: the number of nodes in the gateway instance.
   - **Network Configuration**:
     - Access Port: the access port of the AI gateway.
     - Management port on the private network: the management port of the AI gateway. This port does not require authentication and can only be accessed through Private Link on the private network.
     - Private Link on the private network: the private network address of the AI gateway. Clients can access the gateway nodes of the AI gateway via the private network node address.
     - Public Load Balancer(CLB): public load balancer instance configured for the gateway instance. You can modify the public network bandwidth and configure access policies. You are advised to set access security policies for public network access.

       Note:
       You can select the CLB specification for Private Link and Public Load Balancer. Click **Add**, set the CLB specifications, and click **OK**.



3. On the instance details page, click the **Deployment Architecture** tab to display the status of gateway nodes and AZ information.

   Click to switch views to the deployment architecture diagram, which displays the access layer (CLB) and the deployment status of gateway nodes.

Upgrading the Gateway Specifications

Last updated：2026-05-07 17:26:54

Overview

If the gateway instance specifications do not meet your business requirements, you can increase the number of gateway nodes and node specifications in the console.

> **Note**
> If you need to reduce the number of gateway nodes and node specifications, please submit a ticket.

Prerequisites

The gateway instance is in the **Running**/**Upgrade failed** state.

Changing the Node Specifications

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, select the "ID" of the gateway instance to be configured and go to its basic information page, where you can view the instance's basic information.
3. In the **Node Configuration** module, click **Modify** at the node specification to go to the Node Modification Page.
4. On the specification change page, select the target node specifications and click **Confirm**. In the confirmation pop-up window, select **Confirm**. The instance upgrade takes approximately 3 – 5 minutes to complete. You can check the task status in the status bar in the gateway list.

Changing the Number of Nodes

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the newly created instance to go to the gateway instance basic information page and view the basic information of the instance.
3. In the **Node Configuration** module, click **Modify** at the number of nodes to go to the Node Modification Page.
4. On the modification page, select the target number of nodes and click **Confirm**. In the confirmation pop-up window, select **Confirm**. The instance upgrade takes approximately 3 – 5 minutes to complete. You can check the task status in the status bar in the gateway list.

Deleting a Gateway Instance

Last updated：2026-05-07 17:26:54

When you no longer need to use an AI Gateway instance, you can delete it to release resources and stop billing. Deletion operations fall into two scenarios: manual deletion and automatic release upon expiration/overdue payment. When manually deleting an instance, you will be provided with risk confirmation and release options; please exercise caution.
This document will guide you through the manual deletion process and explain the rules for automatic resource release by the system after expiration or overdue payment.

Operation Steps

Deleting a Gateway Instance Manually

If you confirm that you no longer need a running AI Gateway instance, you can log in to the console to manually delete it.

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the Instance List page, locate the target instance and click Delete in the Actions column.
3. The system will display a "Confirm Deletion" confirmation window. Please carefully read and confirm the relevant information and options.

- Release resources option
  - Immediate release: The system will permanently terminate all related configurations and data. This action is irreversible; please exercise caution.
  - Release after 2 hours: The system will first move the instance to the recycle bin and retain it for 2 hours (billing will continue during this period). If you need to restore the instance within 2 hours, you can attempt the operation, but it may fail due to insufficient resources. After 2 hours, the system will automatically and permanently release the related configurations and data. This action is irreversible.

4. After confirming the above information and selecting the checkbox, click the **Delete** button at the bottom of the window. To cancel the operation, click **Cancel**.

> ⚠ Note:
> - Only instances in the **Running/Failed** state can be terminated.
> - After the instance is released, all its configurations, routes, monitoring data, and so on, will be cleared. **This operation is irreversible; please proceed with caution.**

Deleting a Gateway Instance Automatically Upon Expiration or in Arrears

If your AI Gateway instance expires without renewal or your account is in arrears, the system will automatically handle it according to the following rules:

1. Resource retention period (up to 7 days): After an instance expires or enters arrears, it will enter the "Isolated" state in the console and be retained for up to 7 calendar days. During this period, the instance cannot provide production services, but saved data and configurations will not be deleted. For details, see Arrears Description.
2. Renewal recovery: During the 7-day retention period, you can locate the instance in the "Instance List" in the console and restore service by selecting **Renew** in the Actions column. After successful renewal, the instance will return to the "Running" state.
3. Automatic release: If the instance is not renewed by the 7th day (inclusive) after expiration/arrears, the system will automatically release all related computing and storage resources at 00:00 on the 8th day (Beijing time). After resource release, all data in the instance will be permanently erased and unrecoverable.

Model Management
Model API

Last updated：2026-05-07 17:26:54

Scenarios

Model API is the unified interface exposed by the AI gateway. Clients use large model capabilities by invoking specific model APIs. The core working principle is: You create a model API and associate it with a backend model service. The gateway automatically generates corresponding access routes based on the configuration. Client requests enter the gateway by matching this route, and the gateway forwards them to the associated model service for processing.
You can create and manage model APIs here to define how clients access and which specific model service requests are routed to. This article describes how to add, edit, and delete model APIs for the AI Gateway, as well as manage their associated model services and automatically generated routes.

Operation Steps

Add Model API

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model API** tab. On the API list page, click **New**.
4. In the "New Model API" window, complete the configuration for the first step, "Basic Information".

Parameter	Required	Description
API Name	Yes	Enter a name for this API for identification. The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("-", "_"). It cannot start with a digit or a separator, and cannot end with a separator.
Scenarios	Yes	Select the purpose of this API. "Text Generation" is supported. The system will preconfigure relevant default routes based on the selected scenario.
Request Protocol	Yes	Select the protocol used by clients to call this API, for example, "OpenAI". This selection will affect how the preconfigured routes and gateway

		process the request/response format.
Route	Yes	Default routes are preconfigured automatically based on the selected "Usage Scenario" and "Request Protocol". Select the routes you need to enable for this API. Each selected route will be combined with the Base Path to generate an independent access path.
Base Path	No	Set a unified route prefix for this API. The complete path for a client request is `/{Base Path}/{Route path}`. For example, if you set the Base Path to `/qwen` and select the route `/v1/chat/completions`, the complete access path is `/qwen/v1/chat/completions`.
Path Simplification	No	After being enabled, the gateway automatically removes the Base Path prefix from the request path before forwarding the request to the backend model service. For example, if a client requests `/qwen/v1/chat/completions`, the backend service actually receives `/v1/chat/completions`. This helps decouple the client request path from the actual path of the backend service.
Description	No	Description of this API for subsequent management.

Note:
In this step, the Base Path you defined and the selected **route** will be combined to form the final access path for this API. The system will automatically preset one or more default routes based on the "Usage Scenario" and "Request Protocol" you selected. For example, when the "Text Generation" scenario and "OpenAI" protocol are selected, the system will preset the `/v1/chat/completions` route. After creation, the gateway will automatically generate a routing rule based on this full path.

5. After completing the basic information configuration, click **Next** to go to the "Select Model Service" step. In this step, you need to bind this API to a specific model service (which has been configured with policies such as vendor, key, model Fallback, and so on).

Parameter	Required	Description
Service type	Yes	Select "Single-Model Service" to indicate that this API is fixedly routed to a backend model service.
Select	Yes	Select an existing model service. You can also quickly create one by clicking the "New Service" link to jump to the model service page.

servi ce		

6. Click **OK** to complete the model API creation. At this point, the gateway will automatically generate a corresponding routing rule for each route you selected in the "Basic Information" step.

View and Edit Model API

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model API** tab.
4. Click the "ID/Name" of the API to go to its details page.
5. Under the "Basic Information" tab, you can view the complete configuration details of the API.
6. On the "Basic Information" tab of the details page, click **Edit** in the upper-right corner to modify its basic information configuration. After making changes, click **OK** to save your changes.

Manage Routes

A route is a rule that the gateway uses to distribute client requests to the corresponding model API. When a model API is created, the system has automatically generated routes based on the configuration.

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model API** tab.
4. Click the "ID/Name" of the API to go to its details page.
5. On the API details page, click the **Route Management** tab to view all routing rules automatically generated by the system for this API. This section presents the route ID, name, type, and complete matching path. The gateway determines which model API processes incoming requests by matching these routing rules.

Manage Associated Model Service

A Model API needs to be associated with a model service to actually work. You can manage its associated model service under the "Basic Information" tab on the API details page.

> **Note:**
> A model API can be bound to at most one model service.

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model API** tab.
4. Click the "ID/Name" of the API to go to its details page.
5. In the "Model Service" section of the "Basic Information" tab, the table lists currently associated model services, including their ID/Name and model provider.
   5.1 If you need to unbind the current model API from the model service, follow these steps:
      5.1.1 In the "Model Service" section table, locate the associated service and click **Unbind** on the right.
      5.1.2 The system will pop up an "Unbind" confirmation dialog box. This dialog will display detailed information about the model API and model service to be unbound. To prevent accidental operations, you must manually enter the name of the model service to be unbound in the text box for reconfirmation.
      5.1.3 After confirming that the information is correct and the service name entered is accurate, click **OK** to complete the unbinding. After unbinding is performed, the current model API will no longer be able to use this model service. To abort the operation, click **CANCEL**.
   5.2 If the model API is not bound to any service, or if you have unbound the existing association, you can rebind it to a model service.
      5.2.1 In the "Model Service" section, click the **Bind Model Service** button.
      5.2.2 In the pop-up selector, select a created model service from the list.
      5.2.3 Click **OK** to complete the binding. From then on, requests through this model API will be processed by the newly bound model service.

Delete Model API

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model API** tab.
4. On the **Model API** list page, locate the target API, click **Delete** under its operation column, and the system will perform a dependency check before deletion.
5. The system will pop up a dialog box for you to confirm the deletion and automatically check whether the API is bound to other resources (such as "Consumer Group" authorization).
   - If there are no dependencies: the pop-up window will directly display the API information, click **Confirm** to delete. When the API is deleted, all automatically generated routing rules will also be

deleted.

- If there are dependencies: the pop-up window will display "Resource Deletion Dependency Check Results", and prompt "There are unresolved dependencies", while listing specific dependencies.

6. If there are dependencies, you need to first remove all listed dependencies. After removing the dependencies, click the **Recheck** link in the pop-up window, and the system will perform the validation again.
7. When the validation passes and the dependency prompt disappears, click **Confirm** to finally delete the API. To cancel the deletion, click **Cancel**.

Model Services

Last updated：2026-05-07 17:26:54

Scenarios

You need to add large model services to the AI Gateway so that the gateway can proxy requests to the corresponding model providers, enabling unified access, routing, degradation, and key management. The AI Gateway supports adding model services from providers such as Hunyuan, Google Gemini, DeepSeek, Qwen, OpenAI, and so on. This document describes how to add, edit, and delete model services for the AI Gateway.

Operation Steps

Add Model Service

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Model Management**, then click the **Model Services** tab, and in the service list, click **New**.
4. In the "New Model Service" window, complete the configuration for the first step, "Basic Information".

Parameter	Required	Description
Service Name	Yes	Enter a service name. The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("-", "_"). It cannot start with a digit or a separator, and cannot end with a separator.
Service type	Yes	Fixed to "AI model service".
Model provider	Yes	Select a model vendor. Supported vendors include Hunyuan, Google-Gemini, DeepSeek, Qwen, and OpenAI.
Model protocol	Yes	Select the model protocol you need to use based on the protocols supported by the model vendor.
Service Address	Yes	Confirm the service address of the model service.

Model key	Yes	Select a pre-configured API key for this vendor, or click "New Key" to navigate to the key management page and add one. The gateway will use this key to call the corresponding model API.
Secret usage policy	No	Defines how the keys are used when multiple keys are configured. The default is round-robin, which can balance the load across multiple keys.
Description	No	Description of this service for subsequent management.

Note:
The big model capabilities provided by the AI model service are offered by third parties, not directly by the AI gateway. Users shall independently evaluate the service applicability and reliability, ensuring their usage complies with relevant laws and agreements. We shall not be held liable for any consequences arising from violations of regulations.

5. After completing the basic information, click Next to go to the "Select Model Policy" step.
   - Model Selection Method: This configuration determines how the gateway handles the model (model) parameter in client requests.

Specified Model

The gateway will ignore the model parameter in client requests and consistently use the model specified in the "Default Model" section below. This mode is suitable for cost control and high availability scenarios, facilitating unified routing and degradation handling.

- Default Model: When the "Model Selection Method" is "Specified Model", you must select a specific model name here.
- Model Fallback: When it is enabled, if a request to the "Default Model" fails, the gateway can automatically switch (Fallback) to other available models according to predefined rules, ensuring service high availability.
- Fallback Rules: After enabling Fallback, you need to select or configure the list of fallback models and switching rules here when the primary model is unavailable.

Passthrough Request Model

The gateway will directly use the model parameter from client requests and forward it to the vendor. This mode is suitable for scenarios requiring flexible client-side control over model selection, but ensure that clients pass the correct model name.

- Model Parameter Validation: When it is enabled, the gateway will validate whether the model parameter in client requests is within the allowed list.
- Allowed Model List: defines the allowlist of model names that the client is allowed to request.
- Handling Policy for Validation Failure: defines the policy for handling model validation failures, supporting "return a 404 error" or "fall back to the default model and degrade the service".

6. After the configuration is completed, click **Confirm** to create the model service.
7. After adding, the newly added service will appear in the service list. Click **Service ID/Name** to view detailed service information.

Edit Service

On the **Model Service** list page, locate the target service, click **Edit** under its operation column to modify the service configuration information, and after modification, click **Confirm** to save.

Deleting a Service

On the **Model Service** list page, locate the target service, click **Delete** under its operation column, and a dependency check will be performed before deletion.

1. The system will pop up a dialog box for you to confirm the deletion and automatically check whether the service is bound to other resources (such as "Model API").
2. Verify results:
   - If there are no dependencies: the pop-up window will directly display the service ID and name, and click **Confirm** to delete.
   - If there are dependencies: the pop-up window will display "Resource Deletion Dependency Check Results" below the service information, and prompt "There are unresolved dependencies", while listing specific dependencies.
3. If there are dependencies, you need to first remove all listed dependencies. After removing the dependencies, click the **Recheck** operation in the pop-up window, and the system will perform the validation again. When the validation passes and the dependency prompt disappears, click **Confirm** to finally delete the service. To cancel the deletion, click **Cancel**.

Certificate Management

Last updated：2026-05-07 17:26:54

Scenarios

This document describes how to manage SSL Certificates required by your gateway, including operations such as creating, deleting, and editing certificates.

Prerequisite

- You have completed Domains, real-name authentication, filing, and other processes. If you have not completed these, you can use Tencent Cloud's domain services. For details, see View Guide.
- You already have SSL Certificates. If you do not have a certificate, you can use the certificate service provided by Tencent Cloud. For details, see View Guide.

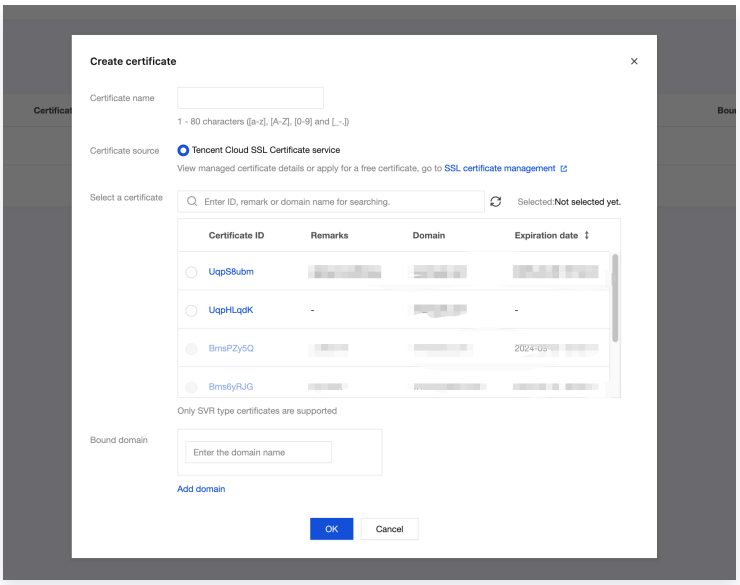
Creating a Certificate

Step 1: Configuring DNS Resolution

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the newly created instance to go to the gateway instance details page.
3. View **Network Configuration > public load balancer instances** to obtain the access IP address.
4. According to the Quick DNS Setup Guide, resolve your domain name to the IP address in the gateway's public network proxy.

Step 2: Creating a Certificate

1. In the left sidebar, select **Certificate Management** and click **Create**.
2. Enter the certificate name, select a certificate, and enter your domain name. **Only SVR certificates are supported.**



3. Click **Confirm** to complete certificate creation. Return to the certificate list to view the created certificate.

Step 3: Verifying Whether the Certificate Takes Effect

Access `https://domain name` to verify whether the certificate takes effect.

Updating the Certificate

Before the certificate expires, you can renew it.

1. On the certificate list page, select the certificate you want to update, and click **Update Certificate** in the operation bar.
2. In the displayed dialog box, select the updated certificate, verify the updated certificate information, click **Confirm** to complete the update.

Edit Certificate

You can modify the certificate name and bound domain names.

1. On the certificate list page, select the certificate you want to update, and click **Edit** in the operation bar.
2. In the displayed dialog box, enter the certificate name, bind a domain name, and click **Confirm** to complete the modification.

Deleting a Certificate

> **Note:**
> After a certificate is deleted, the domain names bound to the certificate cannot use HTTPS to access the gateway.

1. On the certificate list page, select the certificate you want to delete, and click **Delete** in the operation bar.

2. In the displayed dialog box, click **Confirm** to complete the deletion.

Domain Name Management

Last updated：2026-05-07 17:26:54

This guide will quickly show you how to use a custom domain to access the AI gateway. The AI gateway supports IP address-based access but does not provide a domain name access point. Therefore, you need to add an A record in your domain's DNS pointing to the IP address provided by the AI gateway.

Overview

You need to complete the following steps sequentially:

1. Add an A record in DNS to bind to the IP address provided by the AI gateway.
2. Bind a certificate (if HTTPS is used to access the gateway).

Operation Steps

Step 1: Add an A Record in DNS to Bind to the IP Address Provided by the AI Gateway

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, select the instance ID you want to use to go to the gateway instance details page.
3. View **Network Configuration** > **Public Load Balancer** to obtain the access IP address.
4. Add an A record in DNS to bind the domain name to the public IP address.

Step 2: Binding a Certificate (If HTTPS Is Used to Access the Gateway)

If you need to access gateway resources using HTTPS, see the Certificate Management steps to bind a certificate.

> **Note:**
> If no certificate is bound, a certificate mismatch with the custom domain may occur when the custom domain is accessed using the HTTPS protocol. If you are testing with the curl tool, add the `-k` parameter to skip server certificate verification. In a production environment, using HTTPS requires binding a certificate.

Key Management
Model Key

Last updated：2026-05-07 17:26:54

Scenarios

Model Key is the core credential for securely invoking large model services through the AI gateway. When enterprises/developers connect to model APIs from multiple AI vendors via the AI gateway, a critical key scenario exists: **Model Key** is used to store authentication keys (such as vendor AccessKey and API Secret) for each AI vendor. The gateway uses these keys to initiate model API calls to vendors on behalf of users. To ensure the security of sensitive key information, the microservice TSF AI gateway is deeply integrated with Tencent Cloud KMS to achieve encrypted storage of keys throughout their entire lifecycle. KMS uses third-party certified Hardware Security Modules (HSM) to generate and protect keys, ensuring that no one, including Tencent Cloud, can obtain your plaintext master key, meeting strict compliance requirements. Through centralized management, this feature aims to enhance security controls, eliminate the risks of plaintext leakage and unauthorized access, and simultaneously simplify the Ops processes for key creation, update, disablement, and deletion.

Prerequisite

If the generation method uses KMS (KMS credentials), then credentials need to be created. For details, see SSM-Quick Start.

Operation Steps

View Key List

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. The list page displays all created model keys, including information such as Key Name, Type, Status, and Generation Method. You can perform operations like Create, Edit, or Delete here.
5. When the key status is "**Enabled**", the delete operation will be grayed out and unavailable. The system will prompt "Please disable the key first".

Creating Keys

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List Page**, click **Create**.
5. In the "Create Key" window, configure the following parameters:

Parameter	Required	Description
Key Name	Yes	The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("–", "_"). It cannot start with a digit or a separator, nor end with a separator.
Generation method	Yes	**Key Management Service (KMS Credential)**: Associate with a credential in Tencent Cloud KMS. Enter the "Credential Name" and "Credential Version". If no KMS credential exists, you can click "**Create Credential**" to navigate and create one. **Custom**: Manually enter the key value (the model key is the API–KEY value, and the consumer key is the credential content).
Description	No	The identification and description information for the key. Up to 200 characters can be entered.

6. Click **OK** to complete key creation. The gateway ensures encrypted and secure key storage by integrating the KMS service (when KMS credentials are selected as the generation method).

> **Note:**
> - For the generation method, select "**KMS (KMS credentials)**". To manage KMS credentials, navigate to the KMS console for operations.
> - When the generation method is set to "**Custom**", modification is not supported, but copying and viewing are allowed. The value is displayed as "***" by default to protect sensitive information.

Viewing Key Details

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.

3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List page**, click the "ID/Name" of the target key.
5. Go to the key details page, where you can view:
   - **Basic Information**: including key name, type, status, creation time, and so on
   - **Bound Model Resources**: displays the model service resources associated with the current key.

Edit Key

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List page**, locate the target key and click **Edit** under its Operation column; or click **Edit** in the top-right corner of the key **details page**.
5. In the edit window, you can modify the **Name** and **Description** (remarks) of the key.
6. Click **OK** to save the changes.

Key Binding Model Service

The relationship between model services and keys is many-to-many: a model service can bind to multiple keys, and a key can bind to multiple model services. You can bind multiple model services to a key.

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List page**, click the "ID/Name" of the target key to go to the **details page**.
5. Click **Add Resource**, in the "Add New Resource" pop-up window, the "Please select model service" section on the left lists all available model services, you can quickly search using the search box.
6. In the list on the left, select one or more model services to be bound to this key. The selected model services will appear in the "Selected" list on the right.
7. To remove a model service, click the × icon next to the corresponding entry in the "Selected" list on the right to unbind it from this group.
8. After making the adjustments, click **OK** to save the association.

Enabling/Disabling the Key

Model keys are only effective when enabled. This means that when a key is disabled or inactive, the AI gateway will not recognize or use it for any operations. Therefore, before use, it is essential to confirm

whether the key has been properly enabled.

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List** page, locate the target key and click **Disable** in the Actions column. The key will enter the "Disabled" state, and the AI gateway will not recognize or use it to perform any operations.
5. **Enabling Process** requires the target key to be in the "Disabled" state: Click **Enable** to change the key status to "Enabled".

Deleting the Key

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Model Key List Page**, locate the target key, click **Disable** in the Actions column before you can delete it. After the key is disabled, click **Delete**.
5. The system will perform a dependency check before deletion:
   - If the key has been disassociated from all related resources (model keys need to be disassociated from all model services), a pop-up window will display the key information. Click **OK** to delete it.
   - If the key is still associated with resources, a pop-up window will display "Unresolved dependencies exist" and list specific dependent items. You need to resolve all dependencies first, then click **Recheck**. The key can only be deleted after the validation is passed.

> **Note:**
> - KMS credential status changes: If you modify a credential in the Tencent Cloud KMS console, the AI Gateway will temporarily continue using the cached old credential content (default cache duration approximately 5 minutes) to ensure business continuity. We recommend creating a new version of the credential in KMS and associating it with the gateway before the old API Key version is deleted. This ensures the changes take effect promptly.
> - To enhance key high availability, we recommend configuring multiple credentials for model services. This prevents service disruptions when a specific credential is disabled.

Consumer Secret

Last updated：2026-05-07 17:26:54

Scenarios

Consumer Key is the core credential for client applications to securely invoke AI gateway services. When enterprises or developers integrate and schedule backend AI capabilities through the TSF AI Gateway for microservices, the Consumer Key creates unique authentication credentials for different clients (such as business applications, mobile Apps, or third-party services). These credentials are used for identity authentication and access control when these clients invoke gateway APIs.
To ensure the security of sensitive key information, the microservice TSF AI gateway is deeply integrated with Tencent Cloud KMS to achieve encrypted storage of keys throughout their entire lifecycle. KMS uses third-party certified Hardware Security Modules (HSM) to generate and protect keys, ensuring that no one, including Tencent Cloud, can obtain your plaintext master key, meeting strict compliance requirements. Through centralized management, this feature aims to enhance security controls, eliminate the risks of plaintext leakage and unauthorized access, and simultaneously simplify the Ops processes for key creation, update, disablement, and deletion.

Prerequisite

If the generation method uses KMS (KMS credentials), then credentials need to be created. For details, see SSM-Quick Start.

Operation Steps

View Key List

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. The list page displays all created consumer keys, including information such as Key Name, Type, Status, and Generation Method. You can perform operations like Create, Edit, or Delete here.
5. When the key status is "**Enabled**", the delete operation will be grayed out and unavailable. The system will prompt "Please disable the key first.".

Creating Keys

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.

2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List Page**, click **Create**.
5. In the "Create Key" window, configure the following parameters:

Parameter	Required	Description
Key Name	Yes	The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("-", "_"). It cannot start with a digit or a separator, nor end with a separator.
Generation method	Yes	**Key Management Service (KMS Credential)**: Associate with a credential in Tencent Cloud KMS. Enter the "Credential Name" and "Credential Version". If no KMS credential exists, you can click "**Create Credential**" to navigate and create one.
		Auto-generated: The gateway automatically generates a random API key.
		Custom: Manually enter the key value (the consumer key is the credential content).
Description	No	The identification and description information for the key. Up to 200 characters can be entered.

6. Click **OK** to complete key creation. The gateway ensures encrypted and secure key storage by integrating the KMS service (when KMS credentials are selected as the generation method).

> **Note:**
> - When the generation method is set to "**Key Management System (KMS Credentials)**", to handle KMS credentials, navigate to the KMS console to perform operations.
> - When the generation method is set to "**Custom**" or "**Auto-generated**", modification is not supported, but copying and viewing are allowed. The value defaults to *** to protect sensitive information.

Viewing Key Details

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.

3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List page**, click the "ID/Name" of the target key.
5. Go to the key details page, where you can view:
   - **Basic Information**: including key name, type, status, creation time, and so on.
   - **Bound Consumers**: displays the consumer information associated with the current key.

Edit Key

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List page**, locate the target key and click **Edit** under its Operation column; or click **Edit** in the top-right corner of the key **details page**.
5. In the edit window, you can modify the **Name** and **Description** (remarks) of the key.
6. Click **OK** to save the changes.

Key Consumer Binding

The relationship between consumers and keys is one-to-many: a consumer can bind to multiple keys, but a key can only bind to one consumer. You can bind a consumer to a key.

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List page**, click the "ID/Name" of the target key to go to the **details page**.
5. Click **Add Resource**, in the "Add Resource" dialog box, the "Select Consumers" section on the left lists all available consumers. You can quickly search for them using the search box.
6. In the left panel, select a consumer to be bound to this key. The selected consumer will appear in the "Selected" list on the right.
7. To remove a consumer, click the × icon next to the consumer entry in the "Selected" list on the right to remove it from the group association.
8. After making the adjustments, click **OK** to save the association.

Enabling/Disabling the Key

Consumer Keys are only valid when in the enabled status. This means that when a key is disabled or inactive, the AI Gateway will not recognize or use it for any operations. Therefore, it is essential to confirm whether

the key has been properly enabled before use.

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List** page, locate the target key and click the **Disable** button in the Actions column. The key will then enter the "Disabled" state, and the AI Gateway will not recognize or use it for any operations.
5. **Enabling Process** requires the target key to be in the "Disabled" state: Click **Enable** to change the key status to "Enabled".

Deleting the Key

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway > Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Key Management**, go to the Key List page.
4. On the **Consumer Key List page**, locate the target key and click the **Disable** button in the Actions column before you can perform the deletion operation. After the key is disabled, click **Delete** to complete the process.
5. The system will perform a dependency check before deletion:
   - If the key has been disassociated from all related resources (for consumer keys, disassociate from all consumers), the pop-up window will directly display the key information. Click **Confirm** to delete it.
   - If the key still has associated resources, a pop-up will display "Unresolved dependencies exist" and list specific dependencies. You need to remove all dependencies first, then click **Recheck**. The key can only be deleted after the verification is passed.

Note:

- KMS credential status changes: If you modify a credential in the Tencent Cloud KMS console, the AI Gateway will temporarily continue using the cached old credential content (default cache duration approximately 5 minutes) to ensure business continuity. We recommend creating a new version of the credential in KMS and associating it with the gateway before the old API Key version is deleted. This ensures the changes take effect promptly.
- To enhance the high availability of keys, we recommend configuring multiple credentials. This prevents service disruptions for consumers when a specific credential is disabled.

Consumer Management
Consumer group

Last updated：2026-05-07 17:26:54

Scenarios

Consumer Group is used to logically group consumers (that is, API callers), typically corresponding to different business teams, projects, or applications. By authorizing the Consumer Group, you can batch manage the access permissions of all consumers under it to the model API. This article describes how to create, view, edit, and delete Consumer Groups, and manage the association between Consumer Groups and consumers.

Operation Steps

Create Consumer Group

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Consumer Management** > **Consumer Group** to go to the Consumer Group list page.
4. In the "New Consumer Group" dialog, enter the following information.

Parameter	Required	Description
Consumer group name	Yes	Enter a consumer group name. The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("-", "_"). It cannot start with a digit or a separator, and cannot end with a separator.
Status	Yes	Controls whether this consumer group is active. The default state is "enabled". After it is disabled, all consumers in the group cannot access any APIs through the gateway.
Consumer	No	Select one or more existing consumers from the list and add them directly to this group. After creation, you can add or remove consumers at any time.
Description	No	The description information for this consumer group, which facilitates

		subsequent management. Up to 200 characters.

5. Click **OK** to complete the consumer group creation.

View Consumer Group Details

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Consumer Management** > **Consumer Group** to go to the Consumer Group list page.
4. On the **Consumer Group** list page, click the "ID/Name" of any consumer group, and the details panel for that consumer group will expand on the right side of the page.
5. The details panel is divided into two parts: the upper part and the lower part.
   - Consumer Group Information: This section displays the basic information of this consumer group, including Group ID, Name, Status, Creation Time, Modification Time, and Description.
   - Consumers: This section displays all consumers associated with this consumer group in list form, including their ID/Name and Description. You can "unlink" associated consumers here.

Edit Consumer Group

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Consumer Management** > **Consumer Group** to go to the Consumer Group list page.
4. On the Consumer Group list page, locate the target consumer group and click **Edit** under the Operation column.
5. In the "Edit Consumer Group" dialog box, you can modify the group's Name, Status, and Description.
6. After the modifications are made, click **OK** to save.

> **Note:**
> Editing cannot directly modify consumers associated with the group. To manage associated consumers, see the "Manage Associated Consumers" operation below.

Manage Associated Consumers

Consumers and consumer groups have a many-to-many relationship. A consumer can belong to multiple groups, and a group can include multiple consumers. You can add or remove consumers for a consumer group.

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Consumer Management** > **Consumer Group** to go to the Consumer Group list page.
4. On the Consumer Group list page, locate the target consumer group and click **Associate Consumers** under the Operation column.
5. In the "Associate Consumers" dialog box, the "Select Consumers" section on the left lists all available consumers. You can quickly search for them using the search box.
6. In the left panel, select one or more consumers to be added to this group. The selected consumers will appear in the "Selected" list on the right.
7. To remove a consumer, click the × icon next to the consumer entry in the "Selected" list on the right to remove it from the group association.
8. After making the adjustments, click **OK** to save the association.

Delete Consumer Group

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Consumer Management** > **Consumer Group** to go to the Consumer Group list page.
4. On the **Consumer Group** list page, locate the target consumer group and click **Delete** under the Operation column. The system will then perform a dependency check before deletion.
5. The system will pop up a dialog box prompting you to confirm the deletion and automatically check whether the consumer group is bound to other resources (such as "Model API" authorization).
   - If there are no dependencies: the pop-up window will display "Can be safely deleted", and click Confirm to proceed with deletion.
   - If there are dependencies: the pop-up window will display "Resource Deletion Dependency Check Results: There are unresolved dependencies" and list the specific dependencies.
6. If there are dependencies, you need to first remove all listed dependencies. After the dependencies are removed, click the **Recheck** link in the pop-up window, and the system will perform the validation again.

7. When the validation passes and the prompt changes to "Can be safely deleted", click **Confirm** to delete. After deletion, all consumers under this group will immediately lose the access permissions to model APIs obtained through this group. To cancel the deletion, click **Cancel**.

Consumer

Last updated：2026-05-07 17:26:55

Scenarios

A consumer represents the final API caller, such as an independent application, service, or subsystem. Each consumer needs to configure its identity credentials (such as API Key), used for authentication when accessing the gateway. Consumers must join at least one consumer group to obtain access authorization for model APIs through this group. This article describes how to create, view, edit, and delete consumers, and manage consumers' access credentials.

Operation Steps

Create Consumer and Add Model API

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. Click in the left sidebar **Consumer Management** > **Consumer**, go to the Consumer List page.
4. In the "New Consumer" dialog, enter the following information.

Parameter	Required	Description
Consumer name	Yes	Enter a consumer name. The name can contain up to 60 characters, including uppercase and lowercase letters in Chinese and English, digits, and separators ("-", "_"). It cannot start with a digit or a separator, and cannot end with a separator.
Consumer group	No	Select one or more existing consumer groups from the drop-down list. A consumer must belong to at least one group to obtain API access permissions.
Selecting a secret	No	Select one or more created keys from the drop-down list as the identity credentials for this consumer. You can also quickly create a key by clicking the "New Key" link to jump to the key management page.
Description	No	The description information for this consumer, which facilitates subsequent management. Up to 200 characters.

5. Click **OK** to complete the consumer creation. The system will generate a unique consumer ID for the consumer.

View Consumer Details and Manage Credentials

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. Click in the left sidebar **Consumer Management** > **Consumer**, go to the Consumer List page.
4. On the **Consumer** list page, click the "ID/Name" of any consumer, and the details panel for that consumer will expand on the right side of the page. The details panel is divided into two parts:
   - Basic Information: Displays this consumer's ID, Name, Consumer Group, Creation Time, Modification Time, and Description.
   - Credential Management:
     - Authentication Method: Displays the currently used authentication method, such as "API Key".
     - Manage Credentials: This list displays all keys (credentials) bound to this consumer. You can bind more keys to the consumer by clicking the **Add Credential** button above.
     - To add a credential: Click "Add Credential", select the authentication method and specific key in the dialog box, then click **OK** to complete the binding.
     - To remove a credential: Click "Unbind", enter the key name in the dialog box to confirm unbinding, then click **OK** to complete the removal.

Edit Consumer

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. Click **Consumer Management** > **Consumer** in the left sidebar to go to the Consumer List page.
4. On the Consumer List page, locate the target consumer and click **Edit** under the Actions column.
5. In the "Edit Consumer" pop-up window, you can modify the consumer's name and description.
6. After completing modifications, click **OK** to save.

> **Note:**
> Editing operations cannot directly modify the consumer's "Consumer Group" or bound keys. If you need to make changes, view the consumer group in the "Basic Information" section of the Consumer Details page, and manage keys in the "Credentials Management" section.

Delete Consumer

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. Click **Consumer Management > Consumer** in the left sidebar to go to the Consumer List page.
4. On the "Consumer" list page, locate the target consumer and click **Delete** under the Operation column. The system will then perform a dependency check before deletion.
5. The system will pop up a window prompting you to confirm deletion and automatically check whether the consumer is bound to other resources.
   - If there are no dependencies: the pop-up window will display "Can be safely deleted", and click **OK** to proceed with deletion.
   - If there are dependencies: the pop-up window will display "There are unresolved dependencies" and list the dependencies.
6. If there are dependencies, you need to first remove all listed dependencies. After removing the dependencies, click the **Recheck** link in the pop-up window, and the system will perform the validation again.
7. When the validation passes and the dependency prompt disappears, click **OK** to finally delete the consumer. To cancel the deletion, click **Cancel.**

Data Observation
Viewing Default Monitoring

Last updated：2026-05-07 17:26:54

Scenarios

AI gateway provides multi-dimensional monitoring metrics for running gateway instances to comprehensively monitor instance operating status and AI invocation quality. The monitoring metrics cover general gateway performance metrics (such as number of requests, latency, error codes) and LLM-specific metrics for large model scenarios (such as Token consumption, model response time).
You can use these metrics to understand the operating status of gateway instances and various model APIs in real time, gain insights into AI invocation costs and performance, and address potential risks promptly to maintain the stability of AI services and cost controllability. This document describes how to view default monitoring metrics for gateways through the TSF console.

Operation Steps

1. Log in to Microservices Platform Console, in the left sidebar, click **Cloud Native Intelligent Gateway** > **Instance List**.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Data Observation**.
4. You can use the filters at the top of the page to view monitoring data from different dimensions.

Supported Monitoring Metrics and Their Meanings

Request monitoring

Instance/Node

This set of metrics applies to all traffic passing through the gateway, used to evaluate the general performance and health status of the gateway and backend services.

Metric Name	Metric Meaning
Total number of requests	Total number of requests, summed based on the selected time granularity
Average request latency	Average request latency, calculated based on the selected time granularity.

Maximum request latency	Maximum request latency, calculated based on the selected time granularity.
Number of requests directly returned by the gateway	Number of requests that are not forwarded to the backend but directly responded to by the gateway (for example, when authentication fails or traffic throttling is triggered), summed based on the selected time granularity.
Average gateway latency	Average time taken by the gateway itself to process requests.
Maximum gateway latency	Maximum time taken by the gateway itself to process requests.
Number of 2xx requests	Number of requests sent from the client to the AI Gateway that are successful (for example, 200 OK), summed based on the selected time granularity.
Number of 3xx requests	Number of requests sent from the client to the AI Gateway that are redirected, summed based on the selected time granularity.
Number of 4xx requests	Number of requests sent from the client to the AI Gateway that are illegal (for example, due to authentication failure or exceeding the throttling threshold) and are directly responded to by the gateway with client error codes (such as 401 Unauthorized, 403 Forbidden, 429 Too Many Requests), summed based on the selected time granularity.
Number of 5xx requests	Number of requests forwarded by the AI Gateway to the backend service that result in server error responses from the backend (for example, 500 Internal Server Error, 502 Bad Gateway, 504 Gateway Timeout), summed based on the selected time granularity.
Number of 404 requests	Number of requests that fail to reach the backend service because the requested resource is not found on the backend server, summed based on the selected time granularity.
Number of 429 requests	Number of requests failed to be sent to the backend service because the requests are throttled, summed based on the selected time granularity
Number of 499 requests	Number of requests that fail to reach the backend service because the client actively disconnects before a response is received from the backend, summed based on the selected time granularity.
Number of 502 requests	Number of requests that fail to reach the backend service because the requests are throttled, summed based on the selected time granularity.
Number of 504 requests	Number of requests that fail to reach the backend service because the backend server is unreachable when the gateway attempts to execute the

	requests, summed based on the selected time granularity.
Number of requests forwarded to the backend	Number of requests successfully forwarded by the gateway to the backend service, summed based on the selected time granularity.
Average backend latency	Average time taken by the backend service to process requests, calculated based on the selected time granularity.
Maximum backend latency	Maximum time taken by the backend service to process requests, calculated based on the selected time granularity.
Number of backend 2xx requests	Number of requests from the backend service that are successful (for example, 200 OK), summed based on the selected time granularity.
Number of backend 3xx requests	Number of requests from the backend service that are redirected, summed based on the selected time granularity.
Number of backend 4xx requests	Number of requests from the backend service that are illegal, summed based on the selected time granularity.
Number of backend 5xx requests	Number of server-side errors returned by the backend service (for example, 500 backend exception, 502 backend invalid response, 504 backend unreachable), summed based on the selected time granularity.
Number of backend 404 requests	Number of requests that fail because the requested backend service resource is not found on the backend server, summed based on the selected time granularity.
Number of backend 429 requests	Number of requests from the backend service that fail because the requests are throttled, summed based on the selected time granularity.
Number of backend 499 requests	Number of requests from the backend service that fail because the client actively disconnects before it receives a response from the backend, summed based on the selected time granularity.
Number of backend 502 requests	Number of requests from the backend service that fail because the backend service receives an invalid response, summed based on the selected time granularity.
Number of backend 504 requests	Number of requests from the backend service that fail because the backend server is unreachable, summed based on the selected time granularity.

LLM Dedicated Monitoring

This set of metrics is specifically designed for monitoring large model invocation scenarios, helping you analyze Token consumption costs and model provider performance.

Metric Name	Metric Meaning
Number of LLM HTTP Requests	The number of HTTP calls initiated by the gateway to the LLM provider. This metric directly reflects the invocation frequency of the model API.
Total LLM Token Consumption	The total number of tokens consumed by the gateway from the LLM provider, which is the sum of the actual tokens consumed for input (Prompt) and output (Completion). It is used to evaluate the total data throughput of Token consumption.
LLM prompt token Consumption	The total number of tokens actually consumed by the model for the input (Prompt) part when the large language model processes a request.
LLM completion token consumption	The total number of tokens actually consumed by the model for the output (Completion) part when the large language model generates a response. This metric is one of the core bases for evaluating model invocation costs.
Average LLM Provider Response Time (ms)	The average duration from when the gateway sends a request to the model provider to when the complete response is received. This metric reflects the end-to-end response performance of the model provider.
Average Time per token for LLM Provider (ms)	The average time spent by the model provider to consume each Token. This metric reflects the Token consumption speed of the model provider.

System Monitoring

This set of metrics applies to all traffic passing through the gateway, used to evaluate the general performance and health status of the gateway and backend services.

Instance/Node monitoring metrics

Metric Name	Metric Meaning
CPU Utilization	CPU utilization of the AI Gateway, averaged based on the selected time granularity.
Memory Utilization	Memory utilization of the AI Gateway, averaged based on the selected time granularity.
Inbound bandwidth traffic	Ingress bandwidth traffic of the AI Gateway, averaged based on the selected time granularity.
Outbound bandwidth traffic	Egress bandwidth traffic of the AI Gateway, averaged based on the selected time granularity.
TCP inbound connections	The number of TCP connections of the AI Gateway, averaged based on the selected time granularity.

Maximum memory utilization	Maximum memory utilization of the AI Gateway within the selected time granularity. It is used to observe memory usage peaks and determine whether there is a risk of a sudden memory surge, such as memory leaks or sudden traffic pressure.
Maximum CPU utilization	Maximum CPU utilization of the AI Gateway within the selected time granularity. It is used to discover CPU load peak fluctuations and locate performance surges caused by compute-intensive operations, such as complex authentication and protocol conversion.
Number of running nodes	Number of healthy nodes in the AI Gateway within the selected time granularity. It reflects the deployment scale and available node status. An abnormal decrease in the number of nodes may indicate a failure or scaling operation.
New connections from client to gateway process	Number of newly established TCP connections between the client and the gateway process within the selected time granularity. It is used to observe the connection establishment frequency over a short period and determine client connection activity.
Active connections from client to gateway process	Number of TCP connections in an active communication state between the client and the gateway process within the selected time granularity. It reflects the effective connection load currently borne by the gateway.
Inactive connections from client to gateway process	Number of TCP connections that are established but have no active communication between the client and the gateway process within the selected time granularity. It assists in determining connection resource idleness. An excessive number may indicate that the connection reclamation / management mechanism needs optimization.
Concurrent connections from client to gateway process	Total number of concurrent TCP connections (including active and inactive) between the client and the gateway process within the selected time granularity. It directly reflects the concurrent connection pressure on the gateway and is a key metric for evaluating the gateway's connection capacity.
Inbound traffic from client to gateway process	Total data volume sent from the client to the gateway process within the selected time granularity.
Outbound traffic from gateway process to client	Total data volume sent from the gateway process to the client within the selected time granularity.

Inbound bandwidth from client to gateway process	Average bandwidth usage from the client to the gateway process within the selected time granularity (traffic transmission rate per unit of time). It is used to evaluate the bandwidth pressure from the client to the gateway and to avoid connection / transmission delays caused by bandwidth bottlenecks.
Outbound bandwidth from gateway process to client	Average bandwidth usage from the gateway process to the client within the selected time granularity (traffic transmission rate per unit of time). It is used in conjunction with "inbound bandwidth" to analyze the gateway's outbound bandwidth load and prevent bandwidth bottlenecks from affecting response transmission.

Public Network CLB Monitoring Metrics

1. **Client to LB Monitoring**

Metric Name	Metric Meaning
Inbound traffic	Traffic from the client to CLB within the statistical granularity
Outbound traffic	Traffic from CLB to the client within the statistical granularity
Number of inbound packets	Number of data packets sent from the client to CLB within the statistical granularity
Number of outbound packets	Number of data packets sent from CLB to the client within the statistical granularity
Inbound bandwidth	Bandwidth used by traffic from the client to CLB within the statistical granularity
Outbound bandwidth	Bandwidth used by traffic from CLB to the client within the statistical granularity
Number of Active Connections	Number of active connections from the client to CLB within the statistical granularity
Inactive connections	Number of inactive connections from the client to CLB within the statistical granularity
Number of concurrent connections	Number of concurrent connections from the client to CLB within the statistical granularity
New connections	Number of new connections from the client to CLB within the statistical granularity

2. **Discard/Utilization Monitoring**

Metric Name	Metric Meaning
Inbound bandwidth utilization	Utilization of bandwidth used by the client to access CLB through the public network within the statistical granularity
Outbound bandwidth utilization	Utilization of bandwidth used by CLB to access the public network within the statistical granularity
Concurrent connection utilization	Utilization of concurrent connections from the client to CLB at a specific moment within the statistical granularity compared to the performance upper limit of concurrent connections specified in the CLB specifications.
New connection utilization	Ratio of new connections from the client to CLB within the statistical granularity to the maximum number of new connections in the CLB specifications
Discarded connections.	Number of connections discarded by CLB within the statistical granularity
Discarded inbound bandwidth	Discarded data when the client accesses CLB through the public network within the statistical granularity
Discarded outbound bandwidth	Discarded data when CLB accesses the public network within the statistical granularity
Discarded inbound packets	Number of data packets discarded when the client accesses CLB through the public network within the statistical granularity
Discarded outbound packets	Number of data packets discarded when CLB accesses the public network within the statistical granularity
Discarded QPS	Number of requests discarded by CLB within the statistical granularity
QPS utilization	Ratio of QPS of CLB within the statistical granularity to the maximum QPS in the CLB specifications

3. **Monitoring from LB to Backend**

Metric Name	Metric Meaning
Outbound traffic	Traffic from backend servers to the CLB within the statistical granularity.
Inbound bandwidth	Bandwidth used by traffic from the CLB to backend servers within the statistical granularity.

Outbound bandwidth	Bandwidth used by traffic from backend servers to the CLB within the statistical granularity.

4. **Layer 7 Protocol Monitoring**

Metric Name	Metric Meaning
3xx status codes returned by CLB	Number of requests with status code 3xx returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
4xx status codes returned by CLB	Number of requests with status code 4xx returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
5xx status codes returned by CLB	Number of requests with status code 5xx returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
404 status code returned by CLB	Number of requests with status code 404 returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
499 status code returned by CLB	Number of requests with status code 499 returned by CLB within the statistical granularity (sum of codes returned by CLB and gateway node)
502 status code returned by CLB	Number of requests with status code 502 returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
503 status code returned by CLB	Number of requests with status code 503 returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
504 status code returned by CLB	Number of requests with status code 504 returned by CLB within the statistical granularity (sum of codes returned by CLB and the gateway node)
2xx status codes	Number of requests with status code 2xx returned by the backend service within the statistical granularity.
3xx status codes	Number of requests with status code 3xx returned by the backend service within the statistical granularity.
4xx status codes	Number of requests with status code 4xx returned by the backend service within the statistical granularity.
5xx status codes	Number of requests with status code 5xx returned by the backend service within the statistical granularity.
404 status code	Number of requests with status code 404 returned by the backend service within the statistical granularity.
499 status code	Number of requests with status code 499 returned by the backend service

	within the statistical granularity.
502 status code	Number of requests with status code 502 returned by the backend service within the statistical granularity.
503 status code	Number of requests with status code 503 returned by the backend service within the statistical granularity.
504 status code	Number of requests with status code 504 returned by the backend service within the statistical granularity.
Maximum request time	Maximum request time of CLB within the statistical granularity
Average Response Time	Average response time of CLB within the statistical granularity
Maximum response time	Maximum response time of CLB within the statistical granularity
Number of response timeouts	Number of responses from CLB timed out within the statistical granularity
Successful requests per minute	Number of successful requests of CLB within the statistical granularity
Requests per second.	Number of requests of CLB within the statistical granularity

5. **Health Check Monitoring**

Metric Name	Metric Meaning
Number of Health Check Exceptions	Number of health check exceptions for the CLB within the statistical period

Viewing Default Logs

Last updated：2026-05-07 17:26:55

Scenarios

By default, the AI gateway provides you with real-time log service and simple search capability for the gateway, free to use.
By default, logs are primarily categorized into user access logs and gateway error logs. You can view the AI Gateway's access logs to understand request-related information for data analysis, audit, and business troubleshooting. You may also check the AI Gateway's error logs to locate issues.

- Access logs (accessLog) record information related to users' requests and can be used for data analysis, audit, and business troubleshooting.
- Error logs (errorLog) are error logs of the gateway, which are used for gateway troubleshooting.

This document describes the usage instructions for the AI Gateway default log feature.

Prerequisite

AI gateway instances have been created. For detailed operations, see Create AI Gateway.

Viewing Default Logs

1. Log in to the Microservices Platform Console, and click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. Click **Data Observation** > **Default Logs** in the left sidebar.
4. Set the logs you want to view. Related log content is displayed on the page. You can query related logs by keywords. For example, enter "info" to query related logs. Note that log search is case-sensitive.

Edit Default Log Rule

On the Default Log page, click **Edit Logging Rule** in the upper right corner to modify the default log rules. You can choose to continue to use the default rules or customize log rules based on your business requirements.

> **Note:**
> After the default log rules are modified, the log rules of logs shipped to Cloud Log Service (CLS) are also modified. Proceed with caution.

Log Fields

The following table lists the access log fields supported by the AI gateway. You can configure them as needed:

HTTP/HTTPS Log Fields

Field	Description
$remote_addr	Client address.
$status	HTTP status code.
$remote_user	Username provided in basic authentication.
$time_local	Request time.
$request	Complete request line.
$body_bytes_sent	Body size of the file sent to the client.
$request_method	Request method.
$host	Value of the Host field when a request carries the Host request header or the virtual domain name of the host when the request does not carry the Host request header.
$upstream_addr	IP address of the backend service.
$upstream_status	HTTP response code in the response returned by the upstream service.
$upstream_response_time	Upstream service response duration (in milliseconds), from when the gateway establishes a connection to the backend service and receives data to when it disconnects the connection.
$scheme	HTTP or HTTPS protocol.
$url	Request URL.
$request_length	Request data size, in bytes, including the request line, request header, and request body.
$bytes_sent	Number of bytes of the response.
$http_referer	Page source, that is, the URL of the page referenced by the header Referer.
$http_user_agent	Client agent information.
$request_time	Request duration, from when a request is received to when response data is sent, including receiving request data, processing the request,

	and returning response data.

Nginx Variables

Unsupported Nginx variables:

1. The following variables:
   - $connection_time
   - $http3
   - $jwt_claim_
   - $jwt_header_
   - $jwt_payload
   - $memcached_key
   - $mqtt_preread_clientid
   - $mqtt_preread_username
   - $otel_parent_id
   - $otel_parent_sampled
   - $otel_span_id
   - $otel_trace_id
   - $proxy_protocol_tlv_
   - $proxy_protocol_tlv_aws_vpce_id
   - $proxy_protocol_tlv_azure_pel_id
   - $proxy_protocol_tlv_gcp_conn_id
   - $secure_link
   - $secure_link_expires
   - $session_log_binary_id
   - $session_log_id
   - $slice_range
   - $ssl_alpn_protocol
   - $ssl_curve
   - $upstream_queue_time
2. Variables starting with geo

Log Shipping to CLS

Last updated：2026-05-07 17:26:54

Scenarios

If you require persistent log storage for scenarios such as troubleshooting and auditing, it is recommended to enable the CLS log service to ship gateway logs to CLS. Before enabling, please confirm that you have activated the CLS log service. The log service is provided by CLS and incurs fees. For specific billing items, see Pricing Details.

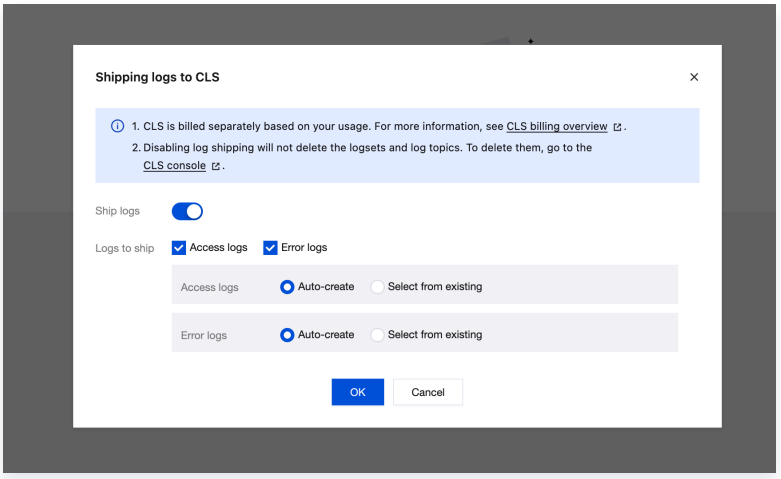
Prerequisite

An AI gateway instance has been created. For specific operations, see Creating an AI Gateway.

Enabling Log Shipping

1. Log in to the Microservices Platform Console, click **Cloud Native Intelligent Gateway** > **Instance List** in the left sidebar.
2. On the instance list page, click the "ID" of the gateway instance to be configured to go to its basic information page.
3. In the left sidebar, click **Data Observation** > **Log Shipping**.
4. On the CLS Log Shipping page, click **Enable Now**, enable the **Log Shipping** button in the dialog box, select the log items to be shipped, which supports automatically creating or selecting existing logsets and log topics. After clicking **OK**, you can start shipping logs.

> **Note:**
> Disabling log shipping will not delete logsets or log topics. To delete them, go to the CLS console for operations.



5. After configuration, you can click the **associated log topic** URL to automatically redirect to the gateway-shipped log database to view logs.

Edit CLS Log Rules

On the Log Shipping page, click **Edit Log Rules** in the upper right corner to modify CLS log rules. You can choose to continue using the default rules or customize log rules based on your business requirements.

> **Note:**
> Logs shipped to CLS use the same log rules as the default logs. After the CLS log rules are modified, the default log rules are also modified. Proceed with caution.

Editing Log Shipping to CLS

On the Log Shipping page, click **Edit Log Shipping** in the upper right corner and modify the log shipping rules in the displayed dialog box.