Elasticsearch Service

智能搜索开发指南

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2026 腾讯云版权所有

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

【商标声明】

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

【服务声明】

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

【联系我们】

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

文档目录

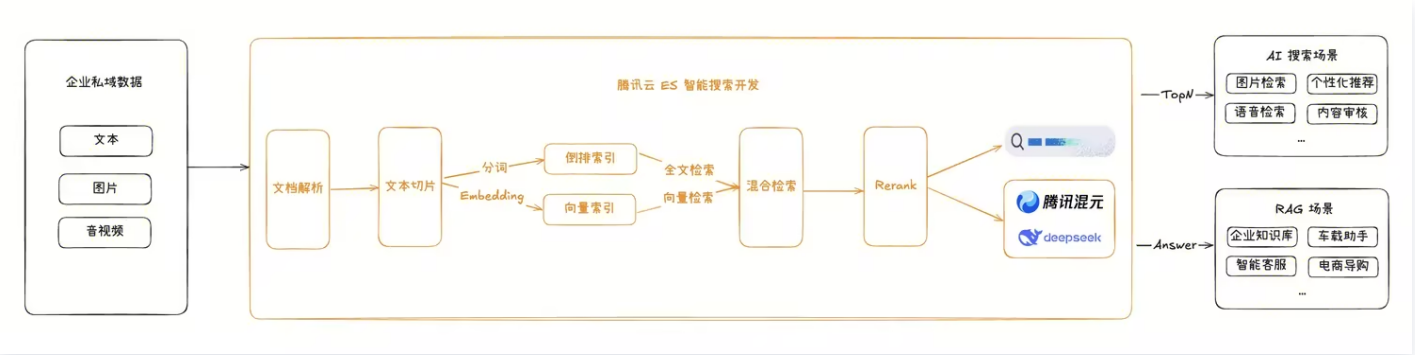
版权所有：腾讯云计算（北京）有限责任公司

智能搜索开发指南
智能搜索开发概述

最近更新时间：2026-08-10 15:22:01

产品能力

腾讯云 ES 智能搜索开发平台是基于云端全托管的 Elasticsearch 服务打造的一站式 AI 搜索解决方案，进一步优化了对全文与向量混合搜索的能力支持，从原始文档解析、向量化等原子能力，到查询性能、混合排序效率、搜索结果精准度等提供了全方位的支持和优化，让搜索有了更多想象空间。 在此基础上，还可以与混元、DeepSeek 等大语言模型无缝集成，从而帮助企业进一步高效、灵活地构建知识问答、多模态检索等 AI 应用，在关键环节也可介入调优。



应用场景

- 文本搜索：依托倒排索引与分词优化，腾讯云 ES 支持亿级数据毫秒级检索，具备拼写纠错、同义词扩展及高亮显示功能，广泛应用于电商商品搜索、新闻推荐等场景，精准匹配用户查询意图。
- 多模态搜索：提供多模态数据向量化服务，结合腾讯云 ES 的混合搜索能力，支持文本、图像等混合数据联合检索，例如图搜图、文搜图、文搜视频等场景，突破单一模态限制。
- 知识库问答：通过自然语言处理与语义分析，快速匹配用户问题与知识库内容，将用户问题与检索的内容交给大模型进行总结生成，适用于企业 FAQ、客户机器人等场景，实现高效知识检索与精准答案推送。
- AI 助手：为 AI 助手提供实时知识库检索和对话上下文管理，通过搜索增强与多轮对话支持，提升虚拟助手、智能客服等应答能力。

优势特性

- 文本/向量混合搜索：独有文本与向量的混合搜索能力，支持多模态数据联合检索，通过倒排索引优化与向量算法扩展，实现复杂场景下的精准匹配与召回扩大。
- 开放的原子服务：提供文档解析、文本切片、Embedding、Rerank、LLM 生成以及联网搜索原子服务，集成 ES Inference API，降低 RAG、AI 搜索应用等开发门槛。
- 灵活搭建 AI 应用：高度灵活的写入/检索工作流自定义能力，支持调用内置原子服务和自定义服务，快速构建智能问答、AI 搜索等场景。

版权所有：腾讯云计算（北京）有限责任公司

- 高性能与稳定性：千亿级向量规模支持，毫秒级响应延迟，自研存储优化节省 70%-90% 资源，结合分布式架构与多可用区容灾，保障高并发场景稳定运行。
- 权威认证：作为国内首个端到端 RAG 技术标准制定者和测评通过者（信通院认证），腾讯云 ES 过去一年已助力微信读书 AI 问书、ima、腾讯会议智能助手等顶流应用落地。

版权所有：腾讯云计算（北京）有限责任公司

应用开发
应用开发概述

最近更新时间：2026-08-10 15:47:00

什么是应用开发

应用开发是腾讯云 ES 提供的低门槛 AI Search 应用构建能力。您可以直接导入业务文档，经过检索效果测试与评估后，一键发布为独享服务，快速获得生产可用的 API，无需自建检索引擎或编写复杂的接入代码。
典型的使用流程如下：

1. **创建应用：**在控制台创建一个应用，当前已支持 RAG 应用（多模态搜索敬请期待），作为文档管理与效果调试的工作空间。
2. **导入文档：**将业务文档导入文档库，支持多种数据源与格式。
3. **在线测试：**手动调整检索、LLM 等参数，在线体验和多配置对比效果。
4. **效果评估：**通过上传测试集或基于文档自动生成测试集，自动生成搜索质量与生成质量量化报告。
5. **发布上线：**将调试完成的应用发布为独享服务，获取可直接调用的搜索 API，也可以直接采用生成的代码自主部署。
6. **监控运维：**通过服务监控观察线上服务的运行状态与调用情况，同时可查看 ES 集群资源利用以及原子服务调用情况。

核心价值

- **一站式构建 AI Search 服务**：从文档导入到服务发布全程可视化操作，无需编写检索逻辑即可获得生产级 API。
- **即测即调，效果可见**：在线测试与效果评估能力让您在发布前充分验证检索质量，降低上线风险。
- **独享部署，性能隔离**：发布后的应用服务以独享方式运行，计算资源与其他租户隔离，保障服务稳定性与可预期性。
- **端到端闭环**：文档管理、效果调优、服务发布、运行监控在同一应用内完成，无需在多个产品间切换。

前置条件

使用应用开发前，请确认满足以下条件：

条件	说明
购买 ES AI 搜索增强版	集群版本需为 **9.1.3 及以上**，低于此版本不支持应用开发功能。购买指引参见 创建集群
开通原子服务	应用开发底层依赖原子服务提供的模型推理能力，需先在控制台开通。详见 原子服务概述

购买 LLM 套餐	进入 资源管理 页面，购买 LLM 套餐。

计费说明

应用开发涉及以下费用，请注意区分：

计费内容	计费方式	说明
ES 集群	按量付费/包年包月	ES 集群用于数据存储和搜索的载体，将按照节点规模收费。
原子模型服务	按量付费	文档解析、文本切片、Embedding、Rerank 环节调用的原子模型服务，按实际用量按量计费。
LLM 套餐	包年包月	LLM 用于问答、效果评测，期间产生的 Token 将从包年包月套餐中抵扣。
发布服务	按量付费	发布为独享服务后，按服务实际占用的 CPU 和内存规格按量计费，停止服务后不再产生费用，CPU 和 内存将根据应用服务的负载进行自动扩缩容。

重要说明：

- 原子模型服务费用在应用开发的全流程中均可能产生（包括文档导入处理、在线测试、效果评估等环节），并非仅发布后才计费。
- LLM 服务请购买 LLM 套餐，一个套餐包含多个 LLM 模型调用，具体操作参考 资源管理 。
- 独享服务在不使用时将按最小 CPU 和 内存分配（1c2g）。
- 下线服务后仅保留配置数据，不收取计算资源费用。

功能概览

应用开发目前提供以下能力：

功能	说明
创建应用	创建搜索应用，作为文档管理、效果调试与发布的工作空间。
文档库	管理搜索应用的数据源，支持导入、更新、删除文档，配置数据处理策略。
在线测试	在控制台配置检索链路、LLM 提示词等参数，即可直接输入查询词进行检索问答测试，实时查看搜索结果与生成效果。
效果评估	基于评测数据集对检索效果进行量化评估，输出召回率、精确率、忠实度等指标，辅助效果调优。

版权所有：腾讯云计算（北京）有限责任公司

发布管理	将调试完成的应用发布为独享服务，支持版本管理，同时支持根据配置参数生成对应的代码，方便快速部署。
服务监控	监控已发布服务的运行状态、资源利用情况，同时支持查看对应的 ES 集群和原子服务关键指标。

各功能的详细使用方法请单击对应链接查看。

创建应用

最近更新时间：2026-08-10 15:47:01

操作场景

在使用智能搜索的应用开发能力之前，您需要先创建一个应用。应用是文档管理、效果调试与发布服务的工作空间，创建后即可导入文档、进行检索测试并发布为独享服务。

前提条件

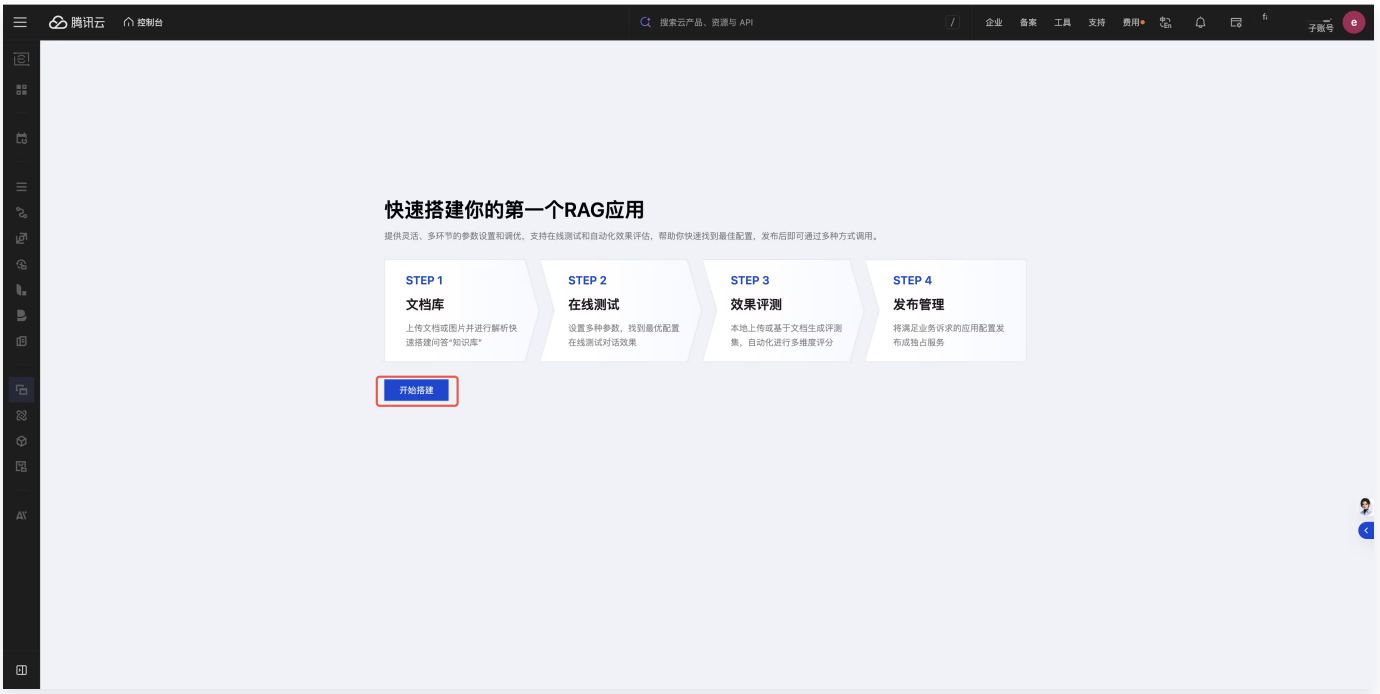
- 已开通原子服务，详情请参见 原子服务概述 。
- 已有 9.1.3 及以上版本的 ES AI 搜索增强版集群，详情请参见 创建集群 。
- 已购买 LLM 套餐，详情请参见 资源管理 。

操作步骤

1. 登录 腾讯云 ES 控制台 ，在左侧导航栏选择 **智能搜索开发** > **应用开发**。

应用开发入口

2. 在应用列表页面，单击**开始搭建**。



创建应用按钮

3. 在弹出的创建应用窗口中，配置以下参数：

创建应用配置

参数	说明
应用类型	RAG 文档，多模态检索敬请期待
所属集群	选择已购买的 ES AI 搜索增强版集群（建议版本 ≥ 9.1.3），用于文本和向量的存储和检索

应用名称	自定义应用的显示名称，长度不超过 50 个字
应用描述	（可选）填写应用的用途或备注说明，便于后续识别和管理

4. 单击 **创建**，完成应用创建。

   创建成功后，应用将出现在应用列表中，您可以在应用卡片或列表中看到应用名称、创建时间等信息。

创建成功

后续操作

应用创建完成后，您可以：

- **导入文档**：将业务文档导入应用的文档库，开始构建搜索数据源。详情请参见 文档库 。
- **在线测试**：在控制台中输入查询词，验证检索效果。详情请参见 在线测试 。
- **效果评估**：基于评测数据集对检索效果进行量化评估。详情请参见 效果评估 。
- **发布服务**：将调试完成的应用发布为独享服务。详情请参见 发布管理 。

服务监控：发布后通过服务监控观察线上运行状态。详情请参见 服务监控 。

文档库

最近更新时间：2026-08-10 15:47:01

操作场景

文档库是应用的数据管理中心。您可以将业务文档导入文档库，系统会自动对文档进行解析、切片与向量化处理，完成处理后即可用于后续的检索测试与效果评估。

操作步骤

进入文档库

1. 登录 腾讯云 ES 控制台，在左侧导航栏选择 **智能搜索开发** > **应用开发**。
2. 在应用卡片中，单击目标应用。

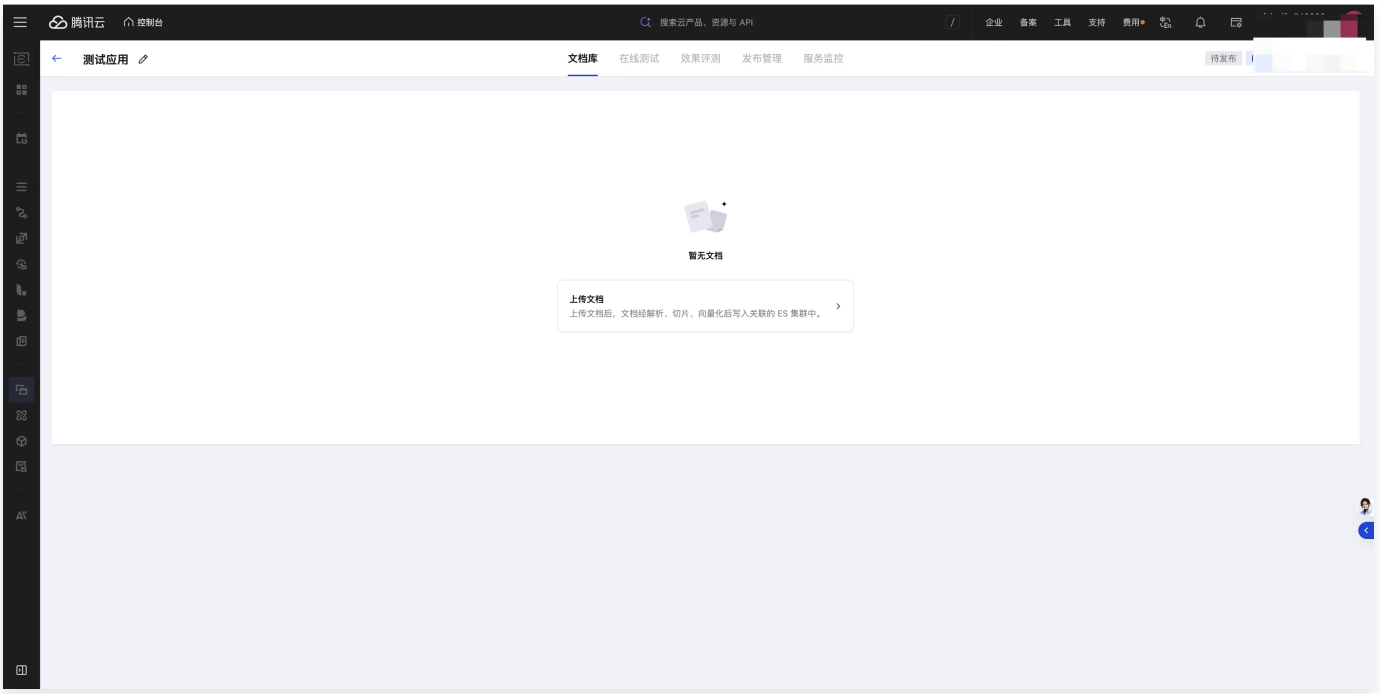
应用卡片

3. 进入 **文档库** 管理页面。

上传文档

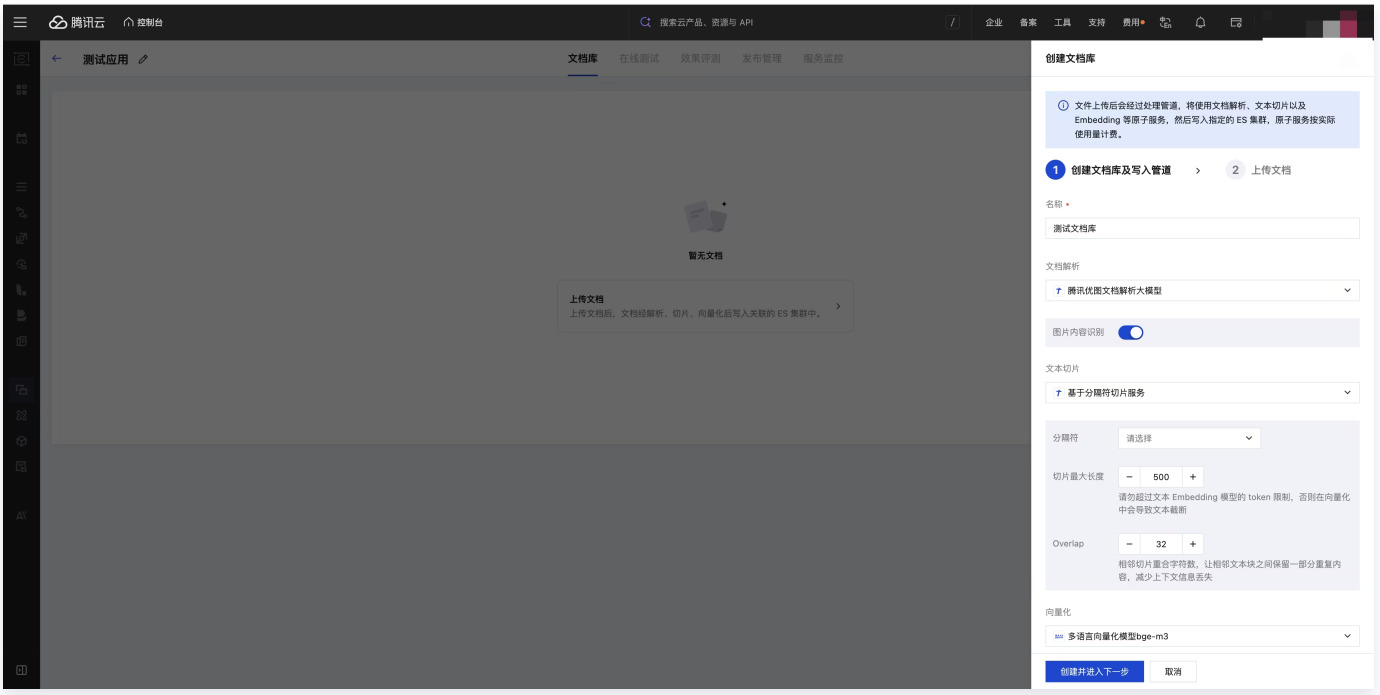
上传文档后，文档经解析、切片、向量化后写入关联的 ES 集群中。

1. 在文档库页面，单击**上传文档**。



文档库

2. 首次上传文档时，需先新建文档库。在弹出的导入文档窗口中，选择导入方式并完成配置。



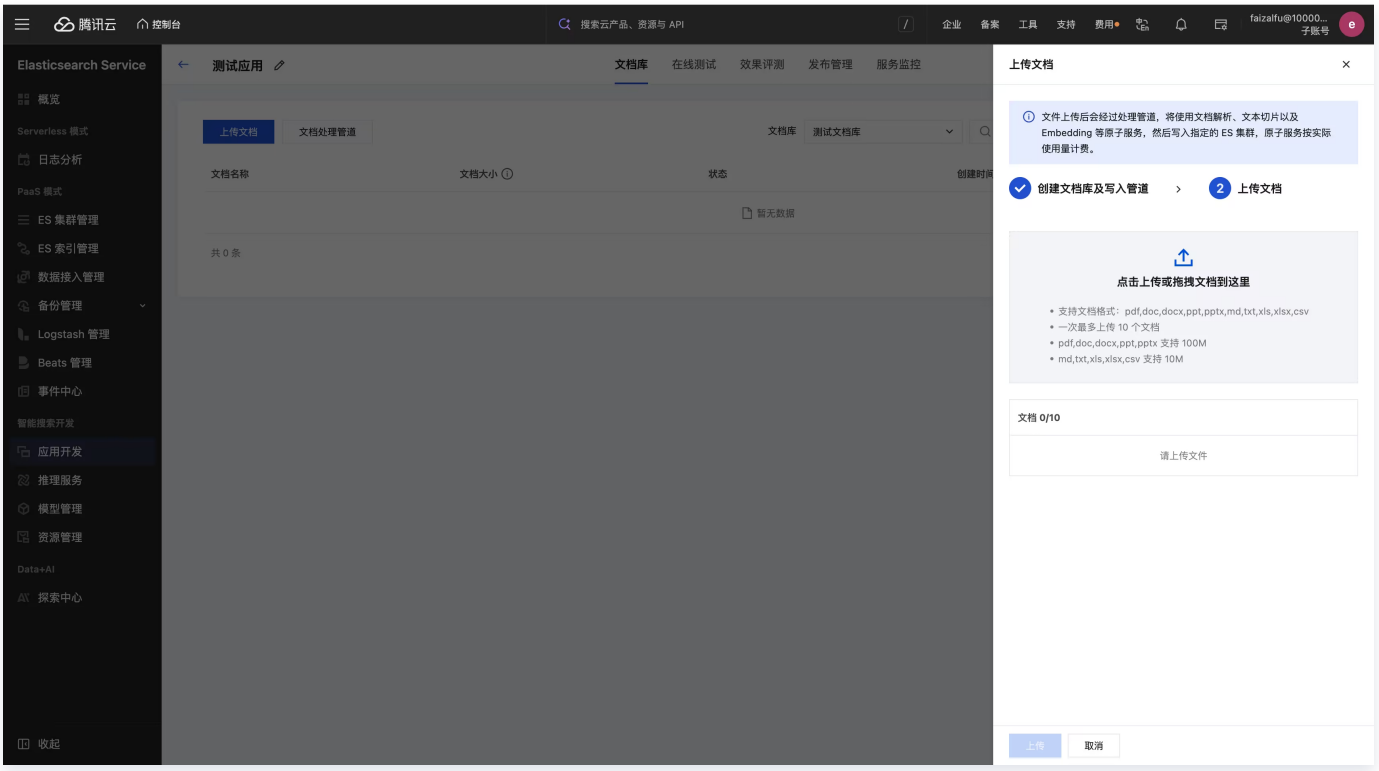
创建文档库

> **说明：**
> 文件上传后会经过处理管道，将使用文档解析、文本切片以及 Embedding 等原子服务，然后写入指定的 ES 集群，**原子服务按实际使用量计费**。

参数	说明

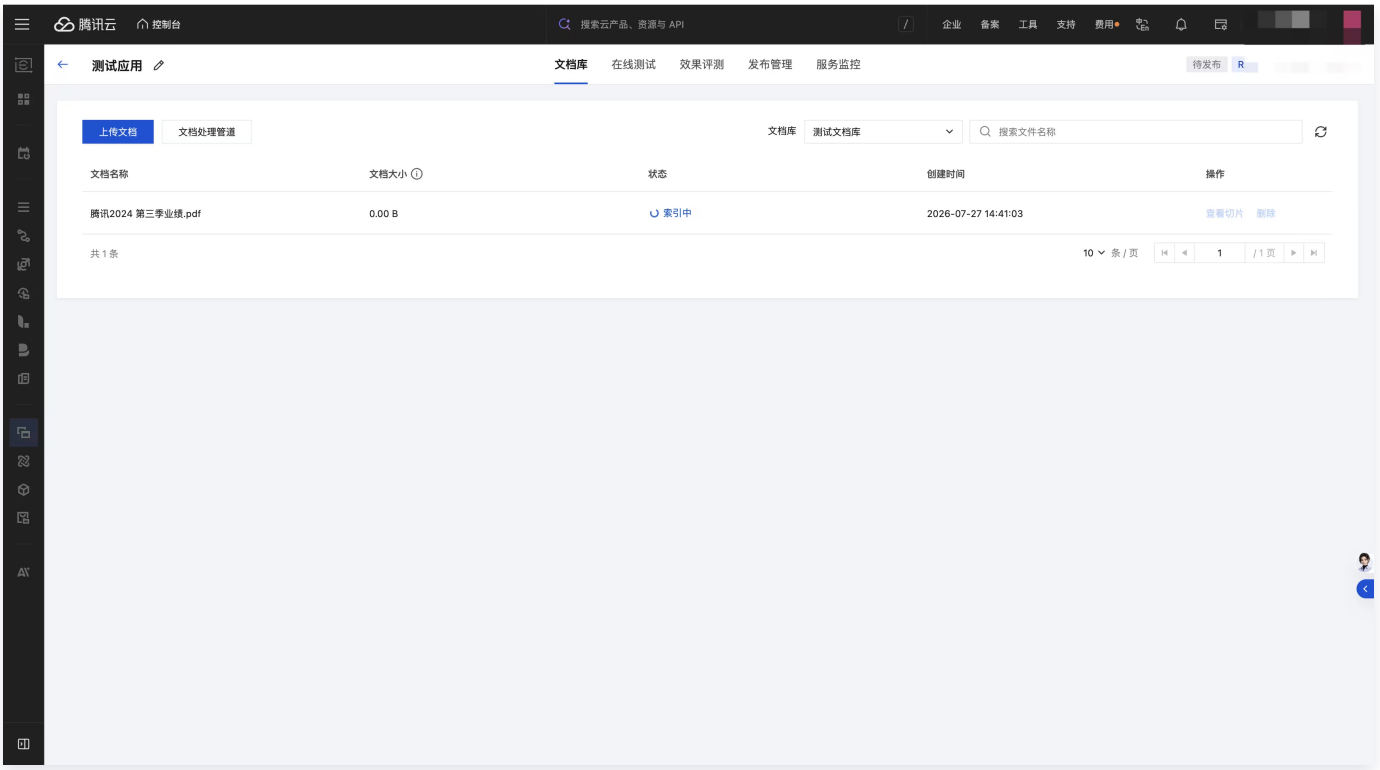
文档库名称	文档库名称
文档解析	选择文档解析模型，可选择是否开启图片内容识别
文本切片	选择文档切分方式，用于控制文档被拆分为检索片段的大小与规则，支持语义切片和规则切片
向量化	选择不同的 Embedding 模型，用于拆分后的文档片段向量化，确定后不支持修改

3. 单击 **创建并进入下一步**，系统基于关联集群，创建索引及 Schema ，开始上传文档。



上传文档

4. 单击 **上传**，进入文档处理。



管理文档

文档处理包括解析、切分、向量化等步骤，处理时间取决于文档数量和大小。处理完成后，文档状态将变更为“已归档”。

管理文档

在文档库页面，您可以对已导入的文档进行以下管理操作：

操作	说明
查看切片	单击文档名称，查看文档的切分结果、向量化状态等处理详情
删除文档	删除不再需要的文档，删除后对应的检索结果也将同步移除

注意事项

- 文档处理过程中调用的原子模型服务会按实际用量产生费用，详情请参见 计费说明 。
- 文档导入后，建议在 在线测试 中验证检索效果，确认文档内容可被正确检索。

后续操作

文档导入完成后，您可以：

- **在线测试**：在控制台中输入查询词，验证检索效果。详情请参见 在线测试 。
- **效果评估**：基于评测数据集对检索效果进行量化评估。详情请参见 效果评估 。
- **发布服务**：将调试完成的应用发布为独享服务。详情请参见 发布管理 。

- **服务监控**：发布后通过服务监控观察线上运行状态。详情请参见 服务监控 。

在线测试

最近更新时间：2026-08-10 15:47:01

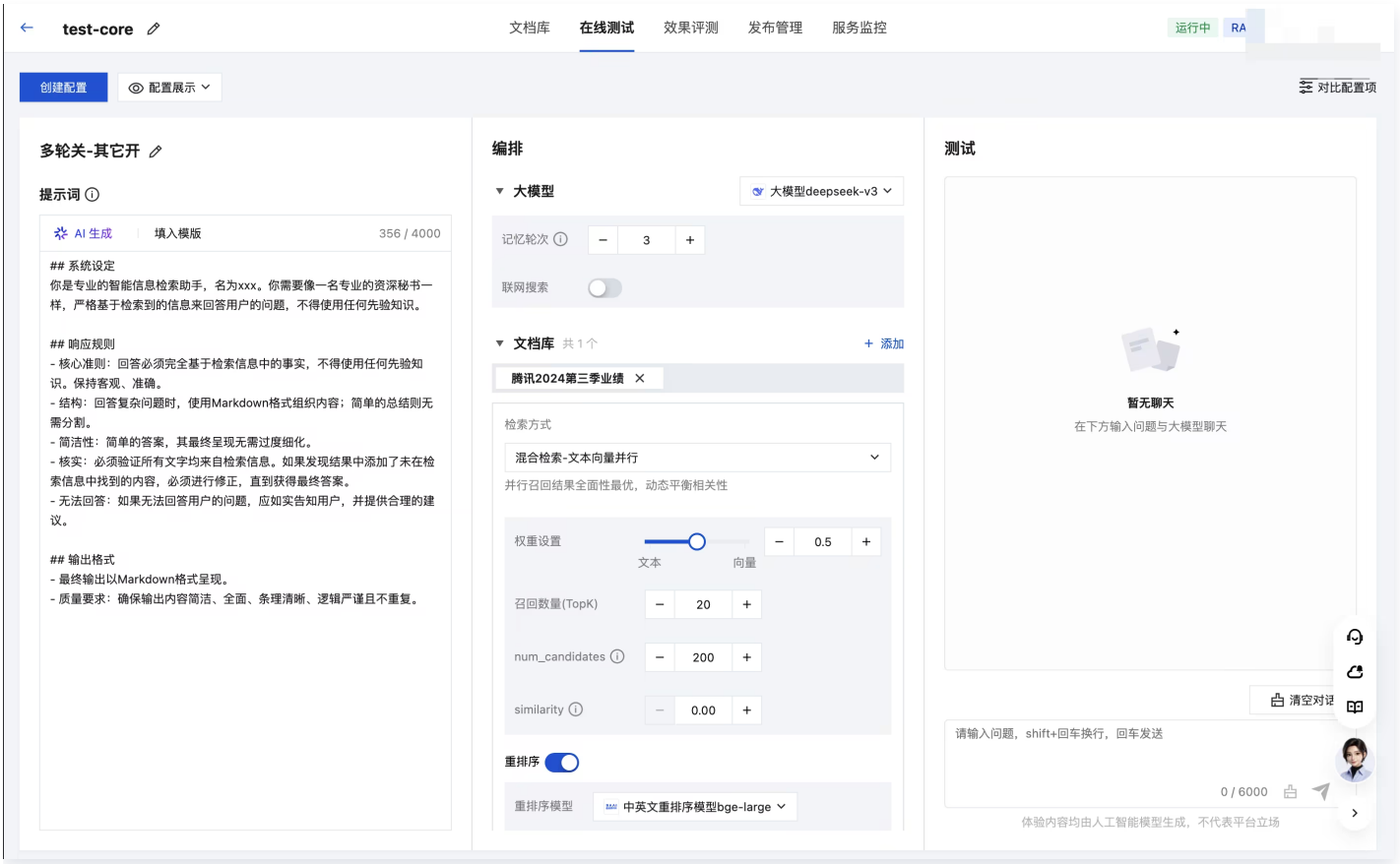
操作场景

在线测试用于在控制台中实时调试和验证智能搜索效果。页面采用三栏布局：**配置区、编排区、测试区**，您可以在配置区管理提示词模板，在编排区调整大模型与检索参数，在测试区即时验证问答效果。在线测试是效果调优的核心工作台，建议在文档导入完成后优先使用。

进入在线测试

1. 登录 腾讯云 ES 控制台，在左侧导航栏选择 **智能搜索开发 > 应用开发**。
2. 在应用列表中，单击目标应用的名称，进入应用详情页。
3. 在应用详情页，选择 **在线测试** 标签页，进入在线测试工作台。

页面说明



在线测试

在线测试页面分为左中右三栏：

区域	说明

配置区（左栏）	管理提示词（Prompt），支持 AI 生成和手动填入两种模式。提示词用于规范大模型的回答行为
编排区（中栏）	配置大模型、文档库、检索方式和各项检索参数
测试区（右栏）	输入问题并实时查看大模型的回答结果，支持多轮对话、查看查询分析以及召回结果

页面顶部操作栏包含：

操作	说明
创建配置	新建一组完整的配置（提示词 + 编排参数），可保存多个配置方便对比
配置展示	控制页面左栏（配置区）的显示/隐藏
对比配置项	支持选择两组不同配置进行 A/B 对比测试

配置区

配置区用于编辑提示词（Prompt），规范大模型基于检索结果回答问题时的行为准则。

提示词编辑

提示词支持两种模式：

模式	说明
AI 生成	由系统基于文档库内容自动生成提示词模板，可在此基础上进行修改
填入模板	填入预置的模板后，手动编写完整的提示词内容

提示词内容限制 **4000 字符**。

提示词编写建议

一个完整的提示词通常包含以下部分：

- **系统设定**：定义大模型的角色和行为边界（如"你是专业的智能信息检索助手"）
- **响应规则**：规范回答的准确性、结构、简洁性和核实要求
- **输出格式**：约束输出的格式和质量标准

编排区

编排区用于配置大模型和检索相关参数，决定检索和回答的核心行为。

版权所有：腾讯云计算（北京）有限责任公司

大模型

参数	说明
大模型	选择用于生成回答的大语言模型
记忆轮次	多轮对话中保留的历史对话轮次数量，影响上下文连贯性，**此记忆仅用于在线测试**，发布后需自行开发
联网搜索	开启后大模型将结合互联网搜索结果回答问题，适合需要实时信息的场景

文档库

参数	说明
文档库	选择参与检索的文档库，支持同时关联多个文档库。单击 + **添加** 可追加文档库

检索方式

参数	说明
检索方式	选择检索算法，如"混合检索-文本向量并行"，同时利用文本匹配和向量语义匹配能力
权重设置	调节文本检索与向量检索的权重比例（如文本 0.5 / 向量 0.5），拖动滑块调整
召回数量（TopK）	每次查询最终返回的结果数量
num_candidates	召回阶段从索引中获取的候选文档数量，通常大于 TopK，值越大召回越全但耗时越长
similarity	相似度阈值，仅返回相似度高于此值的结果，设为 0.00 表示不过滤

重排序

参数	说明
重排序	开启后将对初步召回的结果进行二次排序，提升最终结果的相关性排序质量

大模型精排

参数	说明

大模型精排	开启后由大模型对召回结果进行精细化排序和筛选，进一步优化最终呈现给用户的结果质量 得分说明 • 4 (核心答案): 分片直接、完整地回答了用户问题。 • 3 (核心论据/方法): 分片提供了构建答案所需的核心方法、关键论据或关键范例。 • 2 (背景知识): 分片提供了有用的背景信息或相关上下文，能丰富答案但非核心。 • 1 (提及但无价值): 分片提及了问题关键词，但信息空洞或与问题无关，没有引用价值。 • 0 (完全无关): 分片内容与用户问题的领域完全不相关。

测试区

测试区用于即时验证当前配置的实际效果。

执行测试

1. 在测试区底部的输入框中输入问题（支持 Shift+回车换行，回车发送）。
2. 系统将基于当前编排区的配置执行检索和生成，实时返回大模型的回答。
3. 回答内容包含大模型的推理过程和最终答案，可观察模型如何利用检索到的文档内容作答。

操作说明

操作	说明
清空对话	清除当前测试区的所有对话历史，重新开始测试
多轮对话	支持连续提问，大模型会结合历史对话上下文（受"记忆轮次"参数控制）进行回答
新一轮对话	当修改了提示词及相关参数，将自动发起新一轮对话

注意事项

- 在线测试调用的原子模型服务会产生费用，每次检索和生成均消耗 Token，计费方式参见 智能搜索开发服务定价 。
- 调整参数后无需保存即可直接测试，但若需在发布时使用该参数组合，请通过 **创建配置** 保存为命名配置。
- 在线测试适用于单条或少量查询的效果验证。如需系统性评估检索质量，建议使用 效果评测 。
- 联网搜索和大模型精排功能会增加响应耗时和 Token 消耗，建议按需开启。

后续操作

在线测试中确认检索和生成效果后，您可以：

- 效果评估：基于评测数据集对检索效果进行量化评估。详情请参见 效果评估 。
- 发布服务：将调试完成的应用发布为独享服务。详情请参见 发布管理 。

版权所有：腾讯云计算（北京）有限责任公司

- 服务监控：发布后通过服务监控观察线上运行状态。详情请参见 服务监控 。

效果评测

最近更新时间：2026-08-10 15:47:01

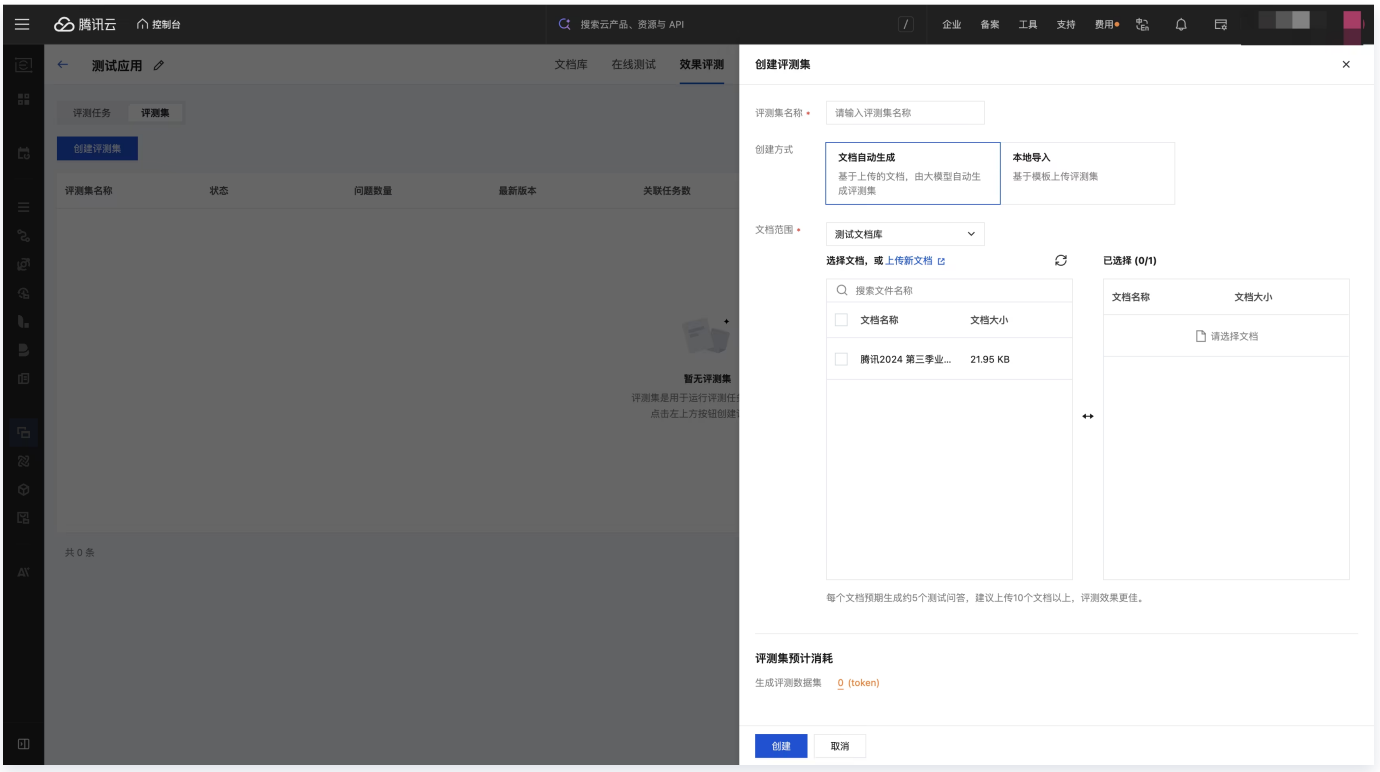
操作场景

效果评测用于对检索与生成的效果进行系统性的量化分析。与在线测试的单条验证不同，效果评测通过评测数据集批量执行查询，输出召回率、精确率等客观指标，帮助您全面衡量搜索、生成质量，为效果调优提供数据依据。
建议在在线测试确认基本检索效果后，使用效果评测进行更深入的量化分析。

创建评测数据集

评测数据集是效果评测的基础，定义了查什么和期望查到什么。

1. 在应用详情页，选择 **效果评测** > **评测数据集** 标签页，单击**创建评测集**。



配置以下参数：

参数	说明
评测集名称	自定义评测数据集的名称，便于区分不同评测任务
创建方式	• 文档自动生成：基于所选文档，由大模型自动生成评测集。系统会按文档内容自动构建测试问答对，无需手动编写。每个文档预期生成约 5 个测试问答。 • 本地导入：基于模板上传评测集，适用于已有离线评测数据的场景。
评测集消耗	基于经验评测，跟文档数据和大小有关，具体请依据实际账单为准。

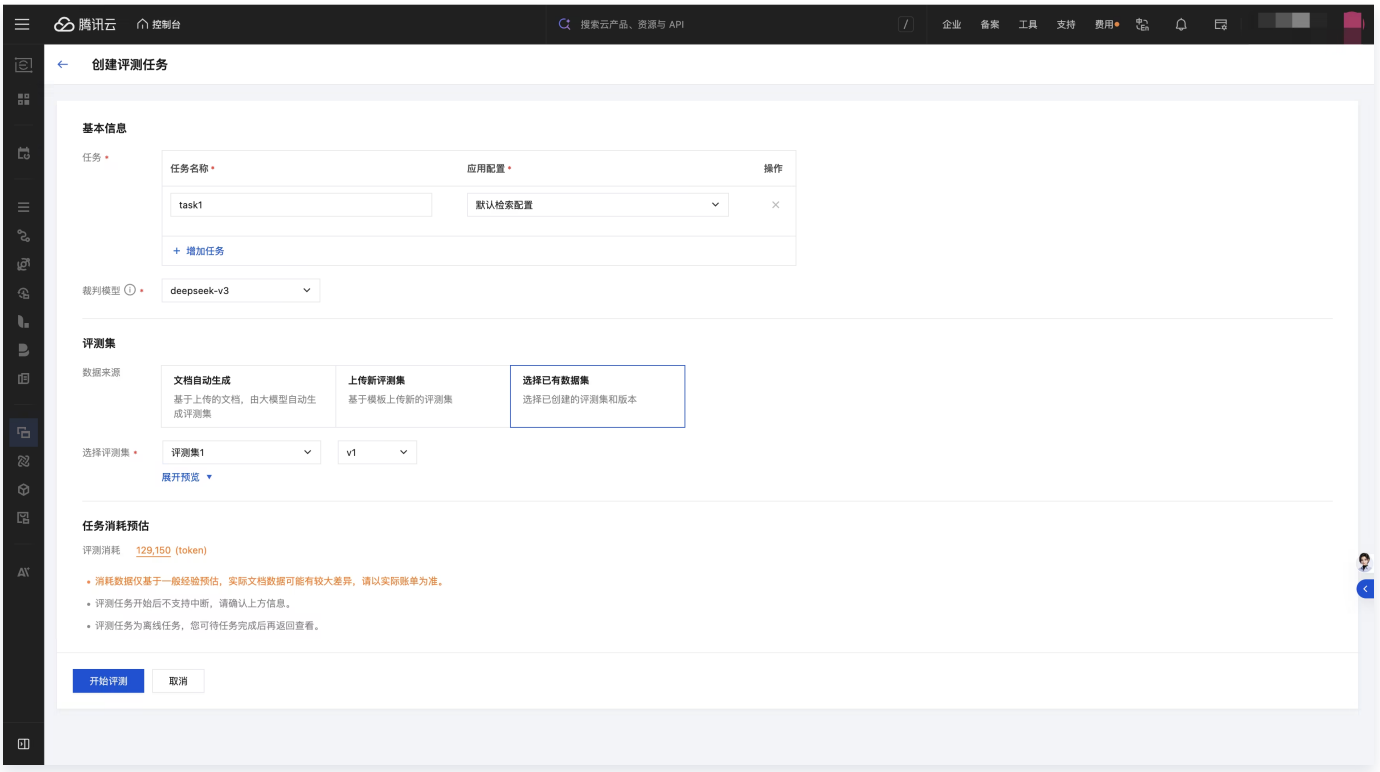
2. 单击**确定**，开始评测集创建。

执行效果评测

1. 在效果评测页面，选择 **评测任务** 标签页，单击 **创建评测任务**。

创建评测任务

2. 选择已创建的评测数据集，并确认评测配置。



配置评测任务

参数	说明
基本信息	配置本次评测包含的任务项，每个任务对应一个待评测的检索配置。
裁判模型	裁判模型用于自动评判检索结果与期望答案的匹配程度，建议选择效果稳定的大模型。该参数会直接影响评测打分的准确性。

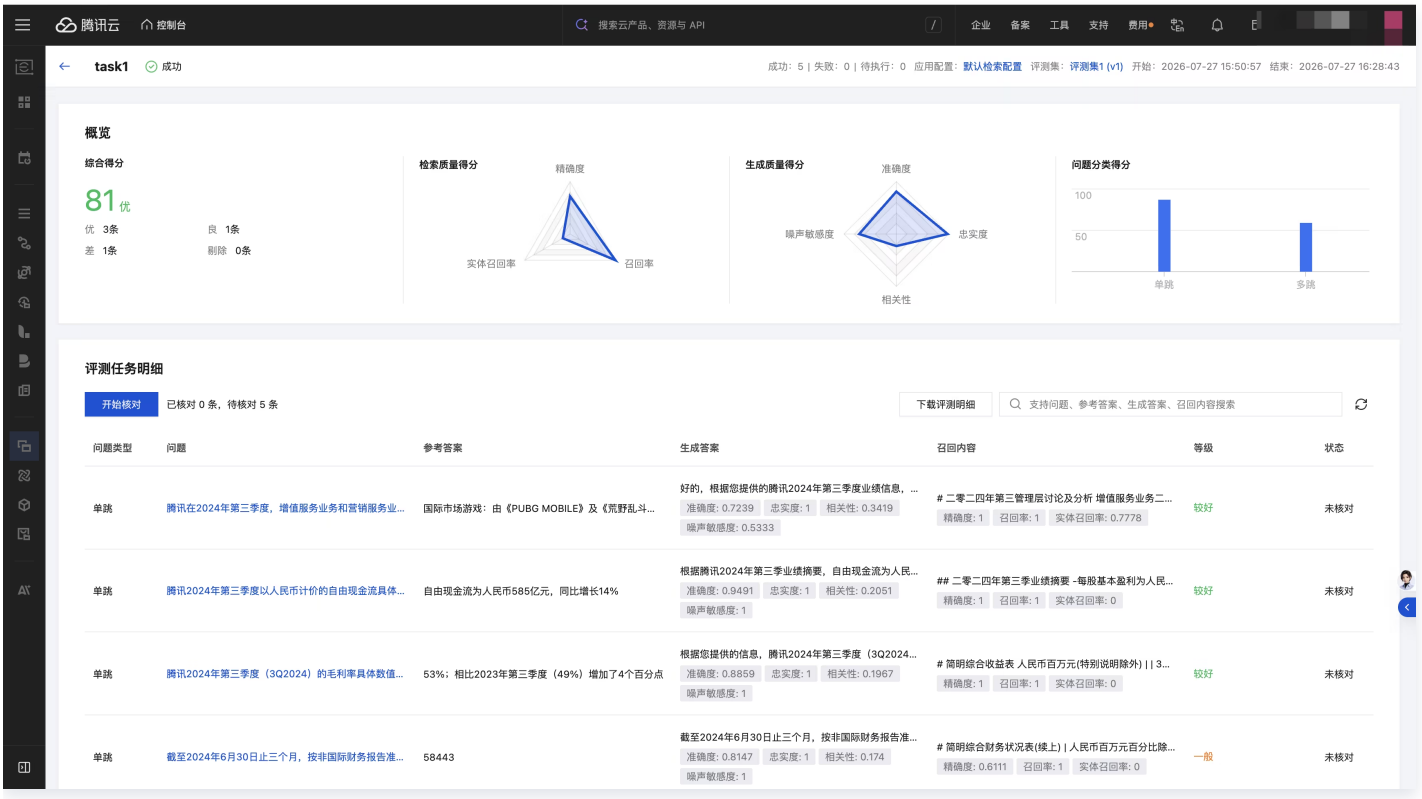
版权所有：腾讯云计算（北京）有限责任公司

评测集	数据来源，支持三种方式 • 文档自动生成：基于上传的文档，由大模型自动生成评测集。 • 上传评测集：基于模板上传评测集。 • 选择已有数据集：选择已创建的评测集及其版本。
任务消耗预估	任务消耗预估（按 token 计费），确认无误后单击**开始评测**。

3. 单击 **开始评测**，系统将按照评测集中的查询词逐一执行检索，并自动计算评测指标。评测耗时取决于数据集的规模，您可以在任务列表中查看评测进度。

查看评测结果

评测完成后，单击任务名称进入评测详情页。



评估报告详情

检索质量得分包含三个指标：

- **精确度（Precision）**：检索返回的结果中相关文档所占的比例。衡量系统的查准能力，精确度越高，用户看到的噪声结果越少。
- **召回率（Recall）**：所有相关文档中被检索命中的比例。衡量系统的查全能力，召回率越高，遗漏的相关内容越少。
- **实体召回率（Entity Recall）**：针对命名实体（如人名、地名、产品型号、专有术语等）的召回率。衡量系统对关键实体信息的覆盖能力，在知识密集型问答场景中尤为关键。

生成质量得分包含四个指标：

- **准确度（Accuracy）**：生成回答的事实正确性，即回答内容是否与客观事实一致。准确度直接决定用户对系统可信程度的判断。
- **忠实度（Faithfulness）**：生成回答对所召回文档内容的忠实程度，即回答是否严格基于检索到的材料，是否存在无依据的编造（幻觉）。忠实度越高，说明回答越可溯源。
- **相关性（Relevance）**：生成回答与用户问题的语义匹配程度，即回答是否紧扣问题主旨，有无偏题或冗余展开。
- **噪声敏感度（Noise Sensitivity）**：生成回答对召回内容中无关信息的抗干扰能力。噪声敏感度越低，说明模型越能忽略检索结果中的不相关内容，仅提取有效信息作答。

您可以根据各项指标的表现，判断评测是否达到预期：

- **优**：检索与生成质量都达到高标准
- **良**：基本可用，少量细节需优化
- **一般**：存在明显问题，建议调优
- **较差**：建议重新审视文档内容或检索配置

注意事项

- 评测数据集的构建直接影响评测结果的参考价值。建议覆盖常见的用户搜索场景，并为每条查询标注准确的期望文档。
- 效果评测过程中调用的原子模型服务会产生费用，计费方式参见 计费说明 。
- 评测结果仅供参考，实际线上搜索效果可能因真实用户的查询习惯不同而有所差异。

后续操作

- **发布服务**：将调试完成的应用发布为独享服务。详情请参见 发布管理 。
- **服务监控**：发布后通过服务监控观察线上运行状态。详情请参见 服务监控 。

版权所有：腾讯云计算（北京）有限责任公司

发布管理

最近更新时间：2026-08-10 15:47:01

操作场景

在文档库导入、在线测试与效果评估均完成后，您可以将应用发布为独享服务。发布后系统将提供一个可直接调用的API，同时以独享方式分配计算资源，保障服务性能与其他租户隔离。

进入发布管理

1. 登录 腾讯云 ES 控制台，在左侧导航栏选择 **智能搜索开发 > 应用开发**。
2. 在应用列表中，单击目标应用的名称，进入应用详情页。
3. 在应用详情页，选择 **发布管理** 标签页，进入发布管理页面。

发布服务

1. 在发布管理页面，单击**发布新版本**。

发布新版本

2. 在发布服务配置页面，设置以下参数：

配置详情

参数	说明
选择配置	选择已配置好的检索配置。选定后，下方参数详情会展示该配置的只读参数，请确认无误
版本描述	输入当前版本的简要说明，便于区分历史版本。限制 200 字以内
费用信息	系统将根据您的实际负载合理分配资源，并按量计算 CPU 费用和内存费用。详情请参见 计费详情

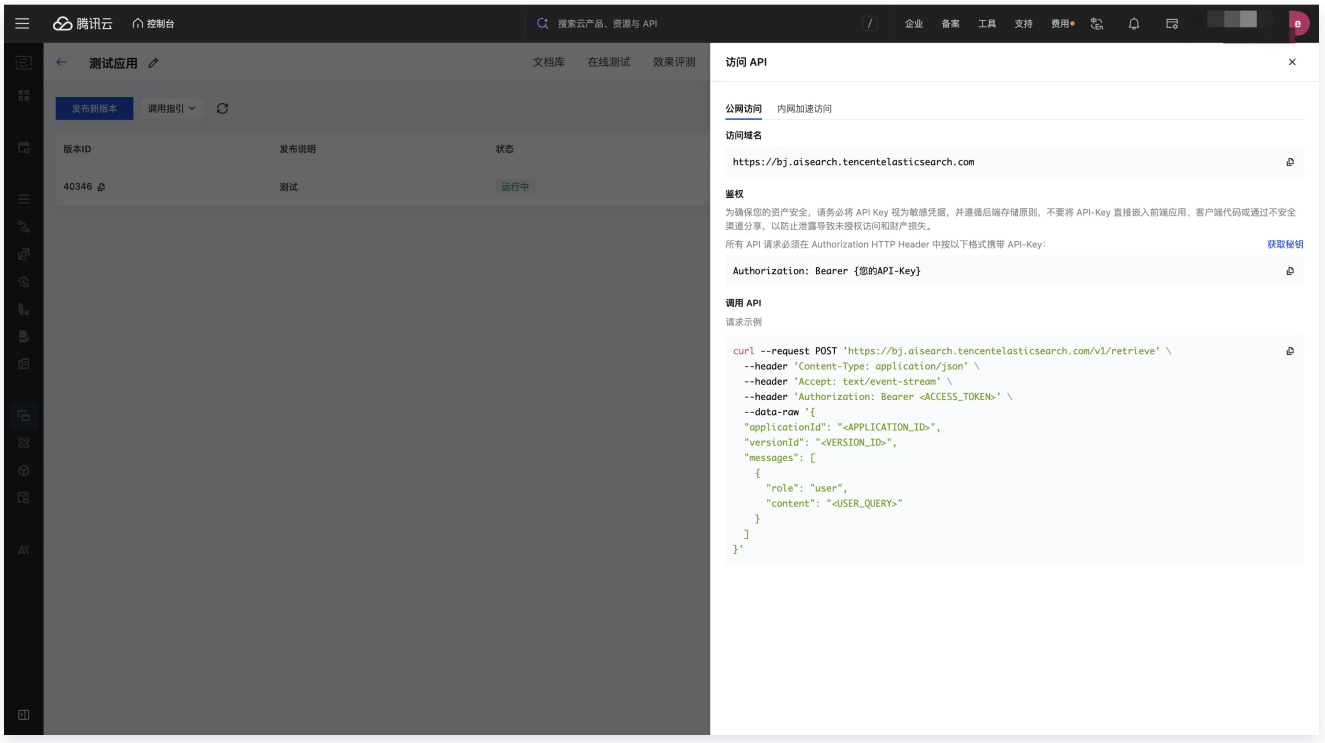
3. 确认配置信息和费用后，单击 **发布**，系统开始部署独享服务。发布过程需要一段时间，您可以在发布管理页面查看发布进度。发布完成后，服务状态将变更为 **运行中**，并自动生成服务访问地址。

版本发布中

4. 发布完成后，单击**调用指引**，支持两种方式调用，用于后续业务系统接入。

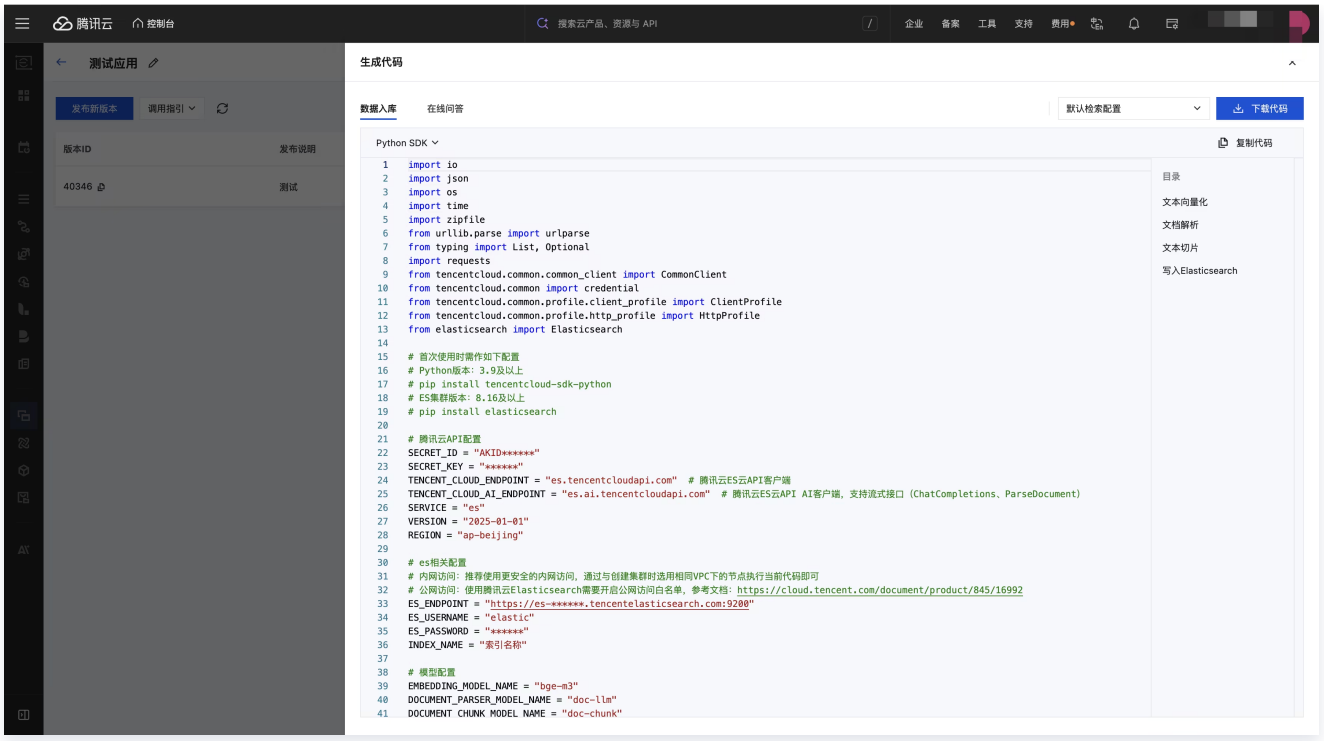
调用指引

4.1 访问 API：支持**公网访问和内网加速访问**，内网访问需要配置相应的 VPC。



访问 API

4.2 生成代码，您可直接下载代码，或选取可用部分，其中相关的配置参数自动更新。



管理发布版本

如果后续对应用的文档或配置进行了调整，您可以重新发布以更新线上服务。

操作	说明
更新	将当前应用的最新状态再次发布，生成新版本。新版本发布完成后，线上服务将自动切换至最新版本
下线	停止运行中的独享服务，停止后仅保留配置数据，不再收取计算资源费用
重启	针对已经停止运行的服务，重启后将重新计算资源费用，服务配置保持不变
删除	删除不再使用的历史版本。已下线的版本方可删除，正在运行的版本需先下线

注意事项

- 独享服务按 CPU 和内存规格按量付费，发布即开始计费，停止服务后不再产生计算资源费用。计费详情参见 计费说明 。
- 建议在充分验证检索和生成效果（通过在线测试和效果评估）后再发布服务，避免频繁重新发布。
- 重新发布会生成新的服务版本，线上服务将自动更新，请确保新版本已通过充分验证。

后续操作

服务监控：发布后通过服务监控观察线上运行状态。详情请参见 服务监控 。

服务监控

最近更新时间：2026-08-10 15:22:01

服务监控用于实时查看智能搜索应用中各服务的运行状态和性能指标，帮助您及时发现异常、定位问题，保障线上服务的稳定性。

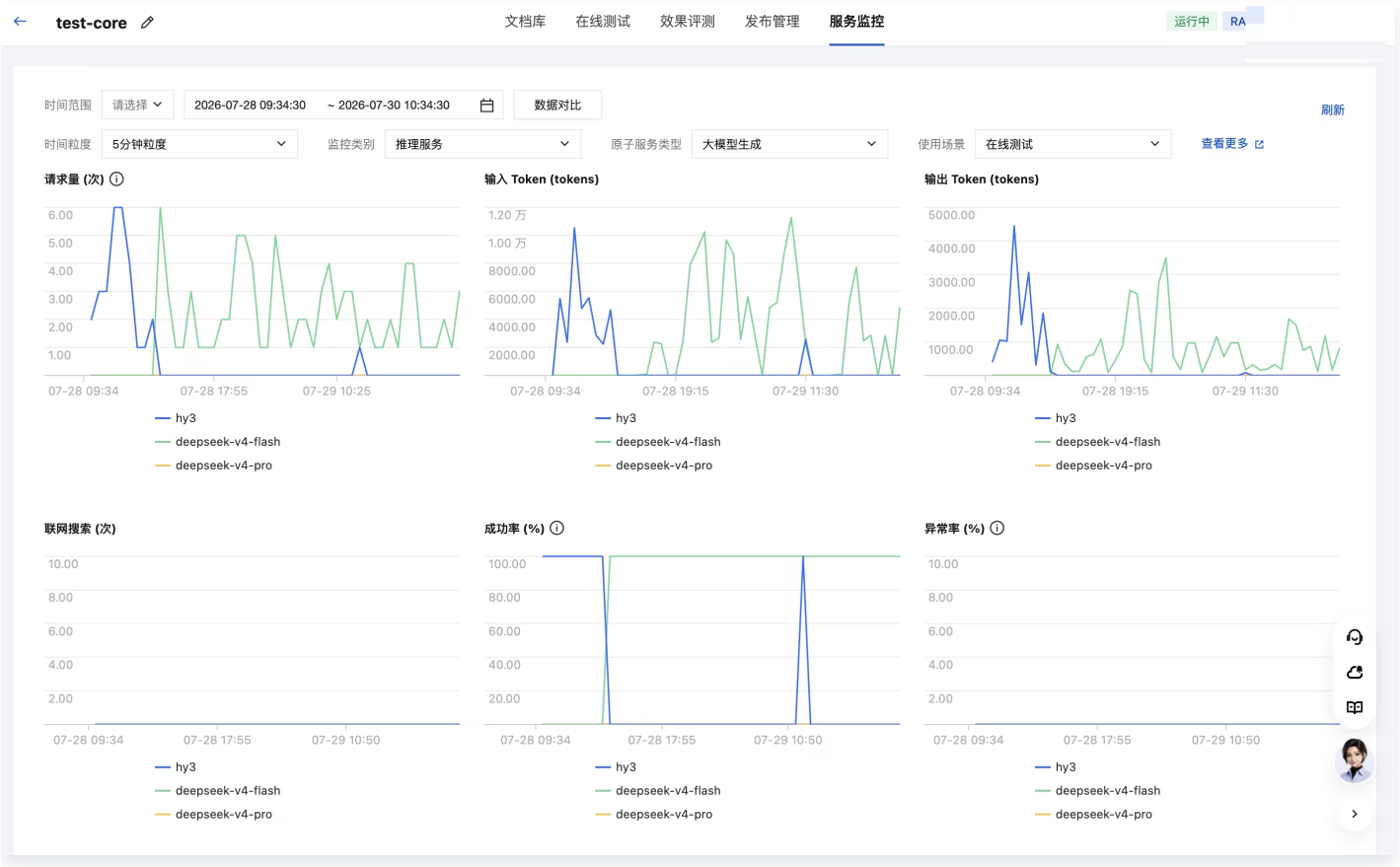
前提条件

- 已创建智能搜索应用。
- 已登录 腾讯云 ES 控制台 。

操作步骤

1. 选择应用，进入应用详情页，单击**服务监控**标签页。
2. 通过页面顶部的筛选条件定位需要查看的监控数据。

页面说明



服务监控

筛选条件

筛选项	说明
时间范围	选择监控数据的起止时间，支持自定义时间段
时间粒度	数据聚合的时间精度，支持 1 分钟/5 分钟粒选项
监控类别	选择监控的服务类型，可选：**应用服务**、**推理服务**、**ES 集群**
原子服务类型	选择具体的原子服务类型，可选：**文档解析**、**文本切片**、**查询分析**、**向量化**、**重排序**、**大模型生产**
使用场景	选择服务的使用场景，可选：**数据入库**、**在线测试**、**应用服务**
文档库	筛选特定文档库的监控数据

应用服务

监控应用层面的整体运行情况，适用于已发布的独享服务，包含**会话指标**和**服务资源**两组指标。

会话指标

指标	说明
会话数（次）	指定时间范围内服务接收到的会话请求总数
会话成功率（%）	会话请求成功处理的比例，反映服务可用性
会话 QPM（次/min）	每分钟会话请求数，反映服务的实时负载情况
会话 Token 输出速度（s）	生成回答时每秒输出的 Token 数量，反映生成响应的速度

服务资源

指标	说明
CPU 分配量（核）	当前独享服务分配的 CPU 核数
CPU 利用率（%）	CPU 实际使用量占分配量的比例
内存分配量（GB）	当前独享服务分配的内存容量
内存利用率（%）	内存实际使用量占分配量的比例

推理服务

监控原子模型推理服务的调用情况，包括文档解析、向量化等原子服务在不同使用场景下的表现。

版权所有：腾讯云计算（北京）有限责任公司

指标	说明
请求量（次）	指定时间范围内推理服务接收到的请求总数
Token 消耗（tokens）	推理服务处理请求所消耗的 Token 总量，直接关联计费，LLM 会有输入和输出 Token 之分
解析页数（页）	涉及文档解析的处理数量
图片识别（次）	在调用文档解析接口时，开启图片识别次数
图片向量化（张）	涉及图片向量化处理的请求次数
平均耗时（ms）	推理服务处理单次请求的平均响应时间
成功率（%）	推理请求成功处理的比例
异常率（%）	推理请求处理失败或超时的比例

ES 集群

监控底层 Elasticsearch 集群的运行状态，分为**文档库级别指标**和**集群级别指标**两组。

文档库级别指标

指标	说明
查询速度（次/秒）	当前文档库的查询 QPS
写入速度（次/秒）	当前文档库的写入 QPS

集群级别指标

指标	说明
CPU 使用率（%）	集群节点的 CPU 使用率
内存使用率（%）	集群节点的内存使用率
硬盘存储使用率（%）	集群节点的磁盘存储使用率

使用建议

- **日常巡检**：关注会话成功率和推理服务成功率，持续低于 99% 时应排查原因。
- **性能调优**：通过平均耗时和 Token 输出速度判断服务响应是否达标，结合 QPM 判断是否存在性能瓶颈。
- **资源评估**：通过应用服务的 CPU/内存利用率评估当前资源配置是否合理，利用率持续偏高时考虑扩容。

版权所有：腾讯云计算（北京）有限责任公司

- **成本监控**：关注推理服务的 Token 消耗指标，结合资源包余量评估计费情况。
- **问题定位**：通过切换监控类别（应用服务 → 推理服务 → ES 集群）逐层下钻，定位问题出现在哪个环节。
- **容量规划**：结合 ES 集群的硬盘存储使用率和写入速度趋势，评估存储是否需要扩容。

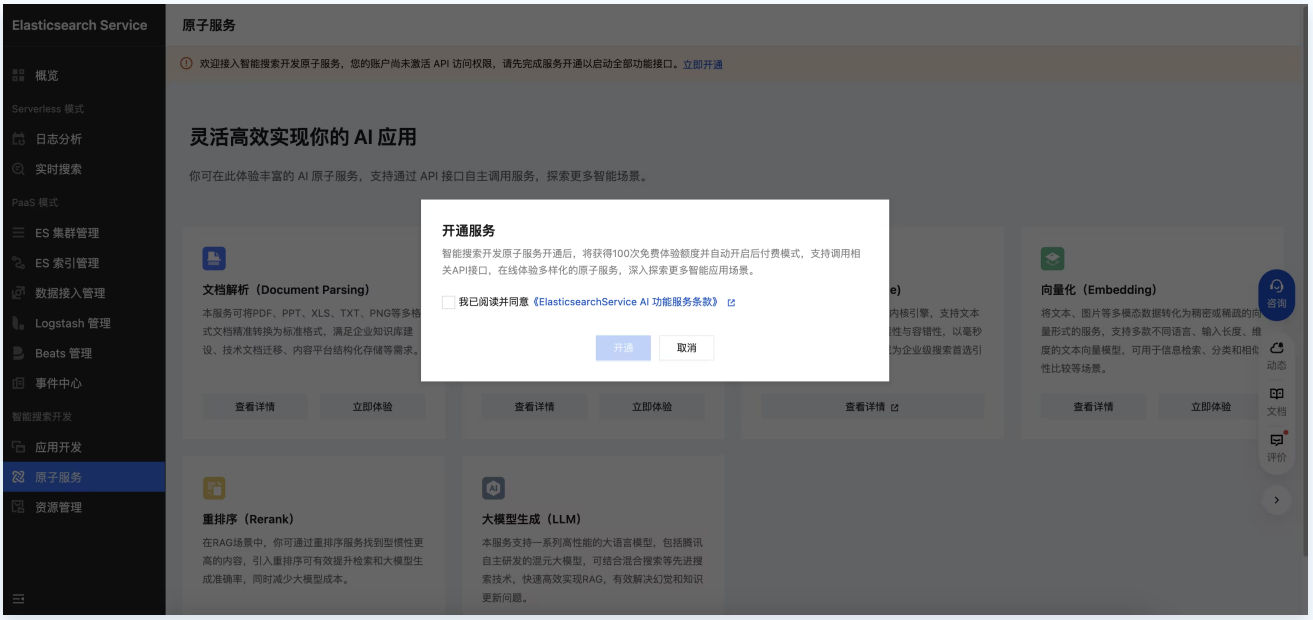
原子服务
原子服务概览

最近更新时间：2026-08-10 15:22:01

原子服务是构成智能搜索链路的基石。腾讯云 ES 将复杂的 AI 搜索能力拆解为一系列独立、可复用的标准化组件，涵盖从非结构化数据解析、语义向量化到大模型推理及召回排序的完整环节。在接下来的章节中，我们将详细展示平台所提供的核心原子服务能力。您可以通过灵活组合这些服务，构建出符合业务需求的定制化 AI 应用。

前置条件

原子服务开通说明：
原子服务开通后才支持接口调用，请及时前往 控制台 开通服务，原子服务开通后将获得 100 次免费体验额度，超出后自动开启**后付费**模式。



原子服务列表

目前智能搜索开发提供以下类型的原子服务，调用方式请参见 原子服务调用指引 。每个模型服务都有默认限频，如有更多需求，请 联系我们 。

类型	描述	服务	默认限频（QPS）	说明
文档解析	本服务可将 PDF、DOC、PPT、XLS、HTML、TXT、PNG 等多格式文档转换为标准格	doc-llm	5	腾讯优图实验室推出的精品 OCR 大模型，覆盖论文、书籍、说明书、试卷、

	式，满足企业级知识库建设、技术文档迁移、内容平台结构化存储等需求。			PPT、海报等众多场景，可满足各行业的文档处理需求。
文本切片	文本切片是将长文本分割为短片段的技术，用于适配模型输入、提升处理效率或信息检索，平衡片段长度与语义连贯性，适用于 NLP、数据分析等场景，本服务支持语义切分。	doc-chunk	100	基于分隔符、文本长度进行切片，适用规则性较强的文本，仅支持 Markdown 与 TXT 格式文件输入。
		doc-tree-chunk	5	腾讯自研文档拆分模型，支持多种文件类型，能够解析并深入理解图表中的信息，语义切分成文本块。
查询分析	承担搜索链路入口的认知决策，通过意图识别精准路由下游交互模式，多轮改写消除上下文指代歧义，搜索改写将复杂问题拆解为多路并行子查询，该模块能有效提升数据检索的准召率。	youtu-intent	2	腾讯优图自研，支持**多语言，最大 1600 token**，支持自定义意图标签识别，负责判定用户请求的交互类型，区分是信息查询、任务执行还是闲聊等，决定后续的链路走向。
		ima-rewrite	3	腾讯 ima 自研，**支持中英文，最大 8000 token**，将用户原始提问智能拆解为多个语义独立、面向不同检索维度的子查询，通过并行或递进检索全面覆盖信息需求。
		ima-rewrite-multiround	2	腾讯 ima 自研，支持**中英文，最大 8000 token**，针对对话式搜索设计，解决指代消解与上下文关联问题，结合历史会话，消除语义歧义，确保每一轮交互建立在连贯的上下文逻辑之上。

Embedding	将文本、图片等多模态数据转化为稠密或稀疏的向量形式的服务，支持多款不同语言、输入长度、维度的文本向量模型，可用于信息检索、分类和相似性比较等场景。	bge-base-zh-v1.5	20	**768 维，最大 512 token** 经典轻量级中文 Embedding 模型，适合资源有限、轻量级 RAG。
		bge-large-zh-v1.5	20	**1024 维，最大 512 token** 经典轻量级中文 Embedding 模型，适合资源有限、轻量级 RAG。
		KaLM-embedding-multilingual-mini-v1	20	**896 维，最大 131072 token** 微信自研多语言模型，支持多语言。
		bge-m3	20	**1024 维，最大 8194 token** 经典 BGE Embedding 模型，支持 100+ 语言，适合多语言检索场景。
		Conan-embedding-v1	20	**1792 维，最大 512 token** 腾讯内部自研，曾登顶中文 CMTEB 榜单，适合高精度中文检索。
		WeCLIPv2-Base	10	**768维，最大 72 token** 微信自研图文 Embedding 模型，在中文场景下表现优秀，适合文搜图、图搜图等场景
		WeCLIPv2-Large	10	
重排序	在 RAG 场景中，可通过 RAG 排序服务找到相关性更高的内容，引入排序服务可有效提升检索和大模	bge-reranker-large	20	**单个 doc 最大 514 token** 经典 BGE 重排序模型，通过交叉编码提升 Top-K 结果相关

	型生成准确率，同时减少大模型成本。			性，支持中文、英文，适合检索结果精排后进行问答。
		bge-reranker-v2-m3	20	**单个 doc 最大 8192 token** 升级版 BGE 重排模型，有更大的文本长度，同时支持多语言。
		ima-post-rerank	5	**单个 doc 最大 4000 token** 腾讯 ima 自研，支持中文、英文，基于 LLM 将用户 query 与一组候选文档进行相关性打分，返回每个文档的得分（0-4 分），用于检索结果重排。
大模型生成	本服务支持一系列高性能的大语言模型，包括 DeepSeek 系列和自主研发的混元大模型，通过深度优化和精确训练，目前已具备强大的自然语言处理能力和高效的文本理解能力，结合混合搜索等先进搜索技术，快速高效实现 RAG，有效解决幻觉和知识更新问题。	deepseek-r1	5	**最大输入 128k，最大输出 8k** 擅长复杂需求拆解、技术方案直译，提供精准结构化分析及可落地方案，实现了与 GPT-4o 和 Claude Sonnet 3.5等模型相媲美的性能。
		deepseek-v3（v3.2）	5	**最大输入 128k，最大输出 8k** 通用型 AI 模型，拥有庞大参数规模及强大多任务泛化能力，擅长开放域对话、知识问答、创意生成等多样化需求。
		deepseek-r1-distill-qwen-32b	5	**最大输入 128k，最大输出 8k** deepseek-r1 32b 参数蒸馏模型，使用

				成本更低，性价比较高。
		hunyuan-turbo	5	**最大输入 28k，最大输出 4k** 腾讯新一代旗舰大模型，混元 Turbo 模型，在语言理解、文本创作、数学、推理和代码等领域都有较大提升，具备强大的知识问答能力。
		hunyuan-large	5	**最大输入 28k，最大输出 4k** 腾讯开发的开源业界参数规模最大、效果最好的 transformer 结构的 MoE 模型，适用对模型效果、复杂指令有较高的要求的场景。
		hunyuan-large-longcontext	5	**最大输入 128k，最大输出 6k** 擅长处理长文任务如文档摘要和文档问答等，同时也具备处理通用文本生成任务的能力。在长文本的分析和生成上表现优异，能有效应对复杂和详尽的长文内容处理需求。
		hunyuan-standard	5	**最大输入 30k，最大输出 2k** 在通用效果提升的基础上，训练数据中融合了医疗、金融领域的长文数据、长文翻译数据和长文多文档问答等高质量精标数据。

		hunyuan-standard-256K	5	**最大输入 250k，最大输出 6k** 256k 极长窗口特化版，复用7B-MoE框架压缩显存占用，支持百页文献级处理，适用绝大部分场景，同时兼顾效果及推理性能。

ES Inference API 调用原子服务

最近更新时间：2026-08-10 15:22:01

本文介绍如何通过 Elasticsearch Inference API 调用智能搜索开发的原子模型服务，将原子服务直接集成到 ES 的检索与数据处理流程中。每个模型服务都有默认限频，如有更多需求，请 联系我们 。

概述

Elasticsearch 提供 Inference API 用于对接外部推理服务。智能搜索开发的原子模型服务兼容 OpenAI 协议，因此可以通过 Elasticsearch 的 `openai` 和 `custom` **两种服务类型**创建推理端点（Inference Endpoint），之后即可在 ES 内部像使用本地模型一样调用这些服务。

通过 ES Inference API 调用的优势：

- 创建一次推理端点后可反复复用，无需在业务代码中管理 API Key 和请求格式。
- 可与 `semantic_text` 字段类型、Ingest Pipeline、`text_similarity_reranker` 检索器等 ES 原生能力直接组合使用。
- 向量化、重排序等操作在 ES 内部完成，减少一次网络往返。

本文涵盖的服务如下：

服务	Inference 任务类型	服务类型	说明
文本向量化	`text_embedding`	`openai` `custom`	将文本转换为稠密向量
重排序	`rerank`	`custom`	基于 Cross-Encoder 模型对候选文档重排
大模型精排	`rerank` / `completion`	`custom`	基于 LLM 对文档做 0-4 级相关性评分
搜索改写	`completion`	`custom`	将单个 query 改写为多个检索 query
多轮改写	`completion`	`custom`	结合对话历史消解指代，补全为独立语句
意图识别	`completion`	`custom`	判断用户意图并路由到候选项

LLM 对话	`completion`	`openai` `custom`	大语言模型文本生成

> **说明：**
> - 文本向量化、重排序、精排同时支持 OpenAI 协议调用，详情请参见 OpenAI 协议调用 。
> - 搜索改写、多轮改写、意图识别、LLM 对话推荐通过 ES Inference API 调用。

搜索改写、多轮改写、意图识别三项能力合称**查询分析**，配合向量化、重排序和 LLM 对话，通常组合使用于 RAG 检索链路：

```
用户问题
  → 多轮改写（消解"它""那个"等指代词，补全为独立问句）
  → 意图识别（判断是闲聊还是检索，多知识库时决定检索哪个库）
  → 搜索改写（拆成多个检索角度，多路召回）
  → 向量/文本混合检索
  → 重排序（Cross-Encoder 粗排）
  → 精排（LLM 细粒度评分与过滤）
  → LLM 生成回答
```

前提条件

- 已开通原子服务。
- 已购买 ES AI 搜索增强版 9.1.3 及以上版本实例。
- 已获取原子服务 API Key。API Key 可在 腾讯云 ES 控制台 的 **资源管理 > API Key** 页面创建和管理。

通用说明

服务地址

项目	说明
原子服务 Base URL	形如 `https://bj.aisearch.tencentelasticsearch.com`，在 **资源管理 > API Key** 页面获取
ES 集群地址	在控制台实例详情页查看，形如 `http://es-xxxxxxxx.vpc.tencentelasticsearch.com:9200`

custom 服务配置说明

创建推理端点的请求格式为：

```
PUT _inference/<task_type>/<inference_id>
```

`service` 固定为 `custom`，`service_settings` 中的核心字段如下：

字段	必填	说明
`url`	是	原子服务的接口地址
`headers`	是	HTTP 请求头，用于携带认证信息和 Content-Type
`request`	是	请求体模板。需为 JSON 字符串，通过占位符注入运行时参数
`response.json_parser`	是	响应解析规则，使用 JSONPath 表达式从响应中提取结果
`secret_parameters`	是	密钥参数，如 `api_key`。配置后 ES 会加密存储，查询端点信息时不返回明文
`batch_size`	否	`semantic_text` 场景下单次请求的最大输入条数，默认 10

不同任务类型的 `json_parser` 字段要求不同：

task_type	json_parser 必需字段
`text_embedding`	`text_embeddings`
`rerank`	`relevance_score`（`reranked_index`、`document_text` 可选）
`completion`	`completion_result`

请求模板占位符

`request`、`headers`、`url` 中可使用 `${}` 形式的占位符，由 ES 在调用时替换为实际值：

占位符	说明
`${input}`	推理请求中 `input` 字段的输入字符串数组
`${query}`	`rerank` 任务的查询文本
`${top_n}`	`rerank` 任务中返回前 N 条结果的数量

版权所有：腾讯云计算（北京）有限责任公司

`${return_documents}`	`rerank` 任务中是否返回文档原文
`${api_key}`	引用 `secret_parameters` 中定义的 `api_key`

> **注意：**
> 占位符不能被引号包裹。正确写法是 `"input": ${input}` ，错误写法是 `"input": "${input}"` 。
> 若占位符在 `secret_parameters` 或 `task_settings` 中找不到对应定义，创建端点时会直接报错。

复杂参数的传递方式

ES Inference API 的 `task_settings.parameters` **不支持数组和对象类型**，因此多轮改写的 `messages` 、意图识别的 `candidates` 这类结构化参数无法通过 ES 模板变量直接传递。

原子服务针对这一限制做了适配：**这些接口的 `query` 字段支持接收 JSON 字符串**，将复杂参数打包后通过 `${input}` 一次性传入。

例如多轮改写， `input` 传入的是这样一个 JSON 字符串：

```
{"query":"它的性能怎么样","messages":[{"role":"user","content":"腾讯云ES是什么"}]}
```

服务端会自动识别 `query` 字段是普通文本还是 JSON 字符串，并做相应解析。涉及此机制的接口会在对应章节中说明。

双解析模式

搜索改写、多轮改写、意图识别、精排四个接口的响应中都包含一个 `completion_result` 字段，其值为**完整响应的 JSON 字符串**。这使得每个接口都支持两种解析方式：

模式	json_parser 配置	返回内容	适用场景
核心结果	指向具体业务字段，如 `$.queries`	仅业务结果	在 ES 搜索管道中直接消费
完整响应	`$.completion_result`	完整 JSON 字符串（含 model、usage 等）	需要 token 用量、置信度等完整信息

建议按需注册不同的 `inference_id` ，例如 `aisearch-query-rewrite` （核心结果）与 `aisearch-query-rewrite-full` （完整响应）。

文本向量化（Embedding）

文本向量化服务将文本转换为向量表示，用于语义检索、相似度计算等场景。

支持的模型

模型名称：

- `bge-base-zh-v1.5`
- `bge-large-zh-v1.5`
- `bge-m3`
- `Conan-embedding-v1`
- `KaLM-embedding-multilingual-mini-v1`
- `Qwen3-Embedding-0.6B`

创建推理端点

有两种方式，第一种直接 openai 协议：

Kibana Dev Tools

```
PUT _inference/text_embedding/tencent-embedding
{
    "service": "openai",
    "service_settings": {
    "api_key": "sk-****",
    "model_id": "bge-m3",
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/embeddings"
  }
}
```

采用 custom：

Kibana Dev Tools

```
PUT _inference/text_embedding/tencent-embedding
{
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/embeddings",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
```

版权所有：腾讯云计算（北京）有限责任公司

```
    },
    "request": "{\"model\": \"bge-m3\", \"input\": ${input}}",
    "response": {
      "json_parser": {
        "text_embeddings": "$.data[*].embedding[*]"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

> **注意：**
> 若需切换模型，将 `request` 中的 `model` 字段替换为目标模型名称即可。建议为不同模型创建不同的 `inference_id`，便于区分和管理。

调用推理

单次调用：

Kibana Dev Tools

```
POST _inference/text_embedding/tencent-embedding
{
  "input": "你好，这是一个测试"
}
```

批量调用：

Kibana Dev Tools

```
POST _inference/text_embedding/tencent-embedding
{
  "input": ["人工智能正在改变世界", "深度学习是机器学习的一个子领域"]
}
```

响应示例

```
{
  "text_embedding": [
    {
      "embedding": [0.0123, -0.0456, ...]
    }
  ]
}
```

使用举例

创建推理端点后，可将其绑定到 `semantic_text` 字段，由 ES 在写入和查询时自动完成向量化，无需业务侧手动调用向量化接口。

```
PUT my-index
{
  "mappings": {
    "properties": {
      "content": {
        "type": "semantic_text",
        "inference_id": "tencent-embedding"
      }
    }
  }
}
```

写入文档：

```
POST my-index/_doc
{
  "content": "Elasticsearch 是一个分布式搜索引擎"
}
```

语义检索：

```
GET my-index/_search
{
  "query": {
    "semantic": {
```

```
        "field": "content",
        "query": "分布式搜索"
      }
    }
}
```

重排序（Rerank）

重排序服务包含两种类型，一种是基于 Cross-Encoder 模型，一种是 LLM-Based 的精排（下文中单独说明的 PostRerank），对候选文档相对于查询的相关性进行重新排序，提升检索结果的精确度。

> **说明：**重排序（Rerank）与精排（PostRerank）是检索链路中两个不同阶段的能力。重排序基于 Cross-Encoder 模型，速度快、成本低，适合对较多候选文档做粗排；精排基于 LLM，理解能力更强、成本更高，适合对粗排后的少量文档做细粒度评分与过滤。典型用法是两者串联：检索 → 重排序 → 精排。

支持的模型

模型名称：

- `bge-reranker-large`
- `bge-reranker-v2-m3`

创建推理端点

Kibana Dev Tools

```
PUT _inference/rerank/tencent-rerank
{
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/rerank",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"bge-reranker-v2-m3\", \"query\": ${query},
\"documents\": ${input}}",
    "response": {
      "json_parser": {
        "reranked_index": "$.results[*].index",
```

```
        "relevance_score": "$.results[*].relevance_score"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

调用推理

Kibana Dev Tools

```
POST _inference/rerank/tencent-rerank
{
  "query": "什么是人工智能",
  "input": [
    "人工智能是计算机科学的一个分支",
    "今天天气很好",
    "机器学习是人工智能的核心技术"
  ]
}
```

响应示例

```
{
  "rerank": [
    {
      "index": 0,
      "relevance_score": 0.9856
    },
    {
      "index": 2,
      "relevance_score": 0.8234
    },
    {
      "index": 1,
```

```
      "relevance_score": 0.0123
    }
  ]
}
```

使用举例

通过 `text_similarity_reranker` 检索器，可在一次查询中完成召回与重排序：

```
GET my-index/_search
{
  "retriever": {
    "text_similarity_reranker": {
      "retriever": {
        "standard": {
          "query": {
            "match": {
              "content": "什么是人工智能"
            }
          }
        }
      },
      "field": "content",
      "inference_id": "tencent-rerank",
      "inference_text": "什么是人工智能",
      "rank_window_size": 20
    }
  }
}
```

大模型精排（Post Rerank）

精排服务基于 LLM 对 query 和文档进行细粒度的相关性评估，输出 0~4 的离散等级评分。与基于 Cross-Encoder 的重排序相比，精排的语义理解能力更强，可识别文档与问题之间更复杂的关联关系，适合在重排序之后对少量文档做最终筛选。

支持的模型

模型名称：`ima-post-rerank`

评分标准

分数	等级	含义
4	核心答案	文档直接、完整地回答了用户问题
3	核心论据/方法	文档提供了构建答案所需的核心方法、关键论据或关键范例
2	背景知识	文档提供了有用的背景信息或相关上下文，能丰富答案但非核心
1	提及但无价值	文档提及了问题关键词，但信息空洞或与问题无关
0	完全无关	文档内容与用户问题的领域完全不相关

实际使用时，通常设定一个分数阈值来过滤低相关文档。**推荐阈值为 2.0**，即保留"背景知识"及以上等级的文档。若希望送入 LLM 的上下文更精炼，可将阈值提高到 3.0。

文档格式说明

精排接口的 `documents` 字段支持三种格式：

格式	示例	适用场景
纯字符串	`"腾讯云ES支持自动扩缩容"`	无标题信息，ES `rerank` 任务默认形式
JSON 字符串	`"{\"content\":\"...\",\"title\":\"弹性伸缩\"}"`	需保留标题，ES `rerank` 任务下的推荐形式
对象	`{"content": "...", "title": "弹性伸缩"}`	直接通过 OpenAI 协议调用时使用

ES `rerank` 任务的 `${input}` 是字符串数组，因此在 ES 中使用时对应前两种格式。文档标题往往包含重要的语义信息，建议采用 JSON 字符串格式以保留 `title`。单次请求最多传入 64 个文档

创建推理端点

精排支持两种注册方式，对应双解析模式。

方式一：rerank 任务类型（推荐）

由 ES 自动解析评分与索引，可直接用于 `text_similarity_reranker` 检索器。

Kibana Dev Tools

```
PUT _inference/rerank/aisearch-post-rerank
{
```

```
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/post-
rerank",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-post-rerank\", \"query\": ${query},
\"documents\": ${input}}",
    "response": {
      "json_parser": {
        "relevance_score": "$.results[*].relevance_score",
        "reranked_index": "$.results[*].index"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

方式二：completion 任务类型（获取完整响应）

返回含 `model`、`usage` 的完整 JSON 字符串。两者代码差异不大，以 Kibana Dev Tools 为例：

Kibana Dev Tools

```
PUT _inference/completion/aisearch-post-rerank-full
{
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/post-
rerank",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-post-rerank\", \"query\":
\"placeholder\", \"documents\": [${input}]}",
```

```
    "response": {
      "json_parser": {
        "completion_result": "$.completion_result"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

> **说明：**
> `"query": "placeholder"` 是占位符。因为 `completion` 任务类型不支持 `${query}` 占位符，实际 query 值需打包到 `input` JSON 字符串中。

调用推理

纯字符串格式（不带标题）：

Kibana Dev Tools

```
POST _inference/rerank/aisearch-post-rerank
{
  "query": "腾讯云ES的性能优势",
  "input": [
    "腾讯云ES支持自动扩缩容",
    "腾讯云ES基于最新内核，查询性能提升30%"
  ]
}
```

JSON 字符串格式（保留标题）：

Kibana Dev Tools

```
POST _inference/rerank/aisearch-post-rerank
{
  "query": "腾讯云ES的性能优势",
  "input": [
```

```
    "{\"content\":\"腾讯云ES支持自动扩缩容\",\"title\":\"弹性伸缩\"}",
    "{\"content\":\"腾讯云ES基于最新内核，查询性能提升30%\",\"title\":\"性能优
化\"}"
  ]
}
```

响应示例

方式一（rerank 任务类型）： 按评分从高到低排序。

```
{
  "rerank": [
    {
      "index": 1,
      "relevance_score": 3
    },
    {
      "index": 0,
      "relevance_score": 1
    }
  ]
}
```

方式二（completion 任务类型）： 返回完整响应的 JSON 字符串。

```
{
  "completion": [
    {
      "result": "{\"object\":\"post_rerank\",\"model\":\"ima-post-
rerank\",\"results\":[{\"index\":1,\"relevance_score\":3},
{\"index\":0,\"relevance_score\":1}],\"usage\":
{\"prompt_tokens\":50,\"completion_tokens\":10,\"total_tokens\":60}}"
    }
  ]
}
```

搜索改写（Query Rewrite）

搜索改写将用户的单个 query 改写为多个语义相近但表述不同的检索 query，用于多路召回，提升检索覆盖率。适用于用户提问较简短、单一表述难以覆盖全部相关文档的场景。
例如“如何使用腾讯云 ES”会被改写为：“腾讯云 ES 使用方法”、“腾讯云 ES 基本操作指南”、“腾讯云 ES 使用步骤详解”三个检索 query，分别召回后合并结果。

支持的模型

模型名称：`ima-rewrite`

创建推理端点

搜索改写只需传入单个 query 字符串，`${input}` 可直接使用，无需打包 JSON。

方式一：仅提取核心结果

`json_parser` 指向 `$.queries`，ES 会为每个改写后的 query 返回一个 `result` 元素。

Kibana Dev Tools

```
PUT _inference/completion/aisearch-query-rewrite
{
  "service": "custom",
  "service_settings": {
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/query/rewrite",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-rewrite\", \"query\": ${input}}",
    "response": {
      "json_parser": {
        "completion_result": "$.queries"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

方式二：获取完整响应

与上述代码差异不大，以 Kibana Dev Tools 为例。

Kibana Dev Tools

```
PUT _inference/completion/aisearch-query-rewrite-full
{
  "service": "custom",
  "service_settings": {
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/query/rewrite",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-rewrite\", \"query\": ${input}}",
    "response": {
      "json_parser": {
        "completion_result": "$.completion_result"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

调用推理

Kibana Dev Tools

```
POST _inference/completion/aisearch-query-rewrite
{
  "input": "腾讯云ES的性能优势"
}
```

响应示例

方式一（核心结果）： 每个改写后的 query 对应一个 `result` 元素。

```
{
  "completion": [
    {
      "result": "腾讯云Elasticsearch性能优势"
    },
    {
      "result": "腾讯云ES性能怎么样"
    }
  ]
}
```

方式二（完整响应）： 单个 `result`，值为完整响应的 JSON 字符串。

```
{
  "completion": [
    {
      "result": "{\"object\":\"query.rewrite\",\"model\":\"ima-
rewrite\",\"queries\":[\"腾讯云Elasticsearch性能优势\",\"腾讯云ES性能怎么样
\"],\"usage\":
{\"prompt_tokens\":0,\"completion_tokens\":0,\"total_tokens\":0}}"
    }
  ]
}
```

多轮改写（Multi Round Rewrite）

多轮改写结合用户的历史对话上下文，对当前 query 进行指代消解和语义补全，使其成为一个独立完整的检索语句。
例如在“腾讯云 ES 是什么”之后追问“它的性能怎么样”，多轮改写会将其补全为“腾讯云 ES 的性能怎么样”，避免直接用“它的性能怎么样”检索导致召回失败。

支持的模型

模型名称：`ima-rewrite-multiround`

input 参数格式

多轮改写需要传入对话历史（`messages` 数组），而 ES 的模板变量不支持数组类型。因此需将 `query` 和 `messages` 打包为 **JSON 字符串**，通过 `${input}` 一次性传入。

`input` JSON 字符串的内部字段：

字段	类型	必填	说明
`query`	string	是	当前轮用户问题
`messages`	object[]	否	历史对话，每个元素含 `role`（`user` / `assistant`）和 `content`

打包后的形态：

```
{"query":"它的性能怎么样","messages":[{"role":"user","content":"腾讯云ES是什么"},{"role":"assistant","content":"腾讯云ES是腾讯云提供的Elasticsearch服务"}]}
```

创建推理端点

方式一：仅提取核心结果

`json_parser` 指向 `$.content`，返回改写后的 query 文本。

Kibana Dev Tools

```
PUT _inference/completion/aisearch-multi-round-rewrite
{
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/query/multi-round-rewrite",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-rewrite-multiround\", \"query\": ${input}}",
    "response": {
      "json_parser": {
        "completion_result": "$.content"
      }
    },
```

```
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

方式二：获取完整响应

```
PUT _inference/completion/aisearch-multi-round-rewrite-full
{
  "service": "custom",
  "service_settings": {
    "url": "https://bj.aisearch.tencentelasticsearch.com/v1/query/multi-
round-rewrite",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"ima-rewrite-multiround\", \"query\":
${input}}",
    "response": {
      "json_parser": {
        "completion_result": "$.completion_result"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

调用推理

Kibana Dev Tools

```
POST _inference/completion/aisearch-multi-round-rewrite
{
  "input": "{\"query\":\"它的性能怎么样\",\"messages\":
[{\"role\":\"user\",\"content\":\"腾讯云ES是什么\"},
```

```
{\"role\":\"assistant\",\"content\":\"腾讯云ES是腾讯云提供的Elasticsearch服务\"}]}"
}
```

响应示例

方式一（核心结果）：

```
{
  "completion": [
    {
      "result": "腾讯云ES的性能怎么样"
    }
  ]
}
```

方式二（完整响应）：

```
{
  "completion": [
    {
      "result": "
{\"object\":\"query.multi_round_rewrite\",\"model\":\"ima-rewrite-multiround\",\"content\":\"腾讯云ES的性能怎么样\",\"usage\":{\"prompt_tokens\":118,\"completion_tokens\":25,\"total_tokens\":143}}"
    }
  ]
}
```

意图识别（Intent Recognition）

意图识别根据用户 query 和一组候选项描述，判断用户意图并路由到最匹配的候选项。典型应用场景：

- **闲聊与检索二分类**（单知识库）：识别为闲聊时跳过知识库检索，直接由 LLM 回答，避免无意义的检索开销。
- **多知识库路由**（多知识库）：从多个知识库候选中选出最匹配的一个，只检索该库，提升准确率并降低成本。

支持的模型

模型名称：`youtu-intent`

input 参数格式

意图识别需要传入候选项列表（ `candidates` ）和可选的对话历史（ `messages` ），两者都是数组类型。因此需将 `query` 、 `candidates` 、 `messages` 打包为 **JSON 字符串**，通过 `${input}` 一次性传入。

`input` JSON 字符串的内部字段：

字段	类型	必填	说明
`query`	string	是	用户问题
`candidates`	object[]	是	候选项列表
`messages`	object[]	否	历史对话上下文

candidates 元素结构：

字段	类型	必填	说明
`id`	string	是	候选项唯一标识，如知识库 ID `kb-001` ，或内置标识 `chat`
`name`	string	是	候选项名称，如"腾讯财报知识库""闲聊"
`description`	string	否	候选项描述，供模型理解该意图的含义范围。**强烈建议填写**，描述质量直接影响识别准确率

messages 元素结构：

字段	类型	说明
`role`	string	消息角色： `user` / `assistant` / `intention`
`content`	string	消息内容。 `role` 为 `intention` 时，内容为该轮识别出的意图 ID

> **说明：**
> `intention` 是意图识别特有的角色类型，用于告知模型历史轮次的意图判断结果，有助于在多轮对话中保持意图连续性。

创建推理端点

方式一：仅提取核心结果

`json_parser` 指向 `$.selected_candidates` ，返回命中的候选项信息。

Kibana Dev Tools

```
PUT _inference/completion/aisearch-intent-recognition
{
  "service": "custom",
  "service_settings": {
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/intent/recognition",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"youtu-intent\", \"query\": ${input}}",
    "response": {
      "json_parser": {
        "completion_result": "$.selected_candidates[*].id"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

方式二：获取完整响应

包含 `confidence` 置信度和每个候选项的 `score`，便于业务侧做阈值判断。

```
PUT _inference/completion/aisearch-intent-recognition-full
{
  "service": "custom",
  "service_settings": {
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/intent/recognition",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"youtu-intent\", \"query\": ${input}}",
```

```
    "response": {
      "json_parser": {
        "completion_result": "$.completion_result"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

> **说明：**
> 注册推理端点时，ES 会自动发送一个验证请求（内容为简单字符串，不含 `candidates`）。服务端已做兜底处理，此时返回空结果而不报错，注册可正常完成。

调用推理

仅传入候选项：

Kibana Dev Tools

```
POST _inference/completion/aisearch-intent-recognition
{
  "input": "{\"query\":\"帮我查一下ES集群的监控数据\",\"candidates\":
[{\"id\":\"knowledge_search\",\"name\":\"知识搜索\",\"description\":\"用户
想搜索知识库中的信息\"},{\"id\":\"task_execution\",\"name\":\"任务执行
\",\"description\":\"用户想执行一个具体操作\"},
{\"id\":\"chitchat\",\"name\":\"闲聊\",\"description\":\"用户想闲聊或打招呼
\"}]}"
}
```

带对话历史（含 `intention` 角色）：

Kibana Dev Tools

```
POST _inference/completion/aisearch-intent-recognition
{
```

```
  "input": "{\"query\":\"那性能监控呢\",\"candidates\":
[{\"id\":\"knowledge_search\",\"name\":\"知识搜索\",\"description\":\"用户
想搜索知识库中的信息\"},{\"id\":\"chitchat\",\"name\":\"闲聊
\",\"description\":\"用户想闲聊或打招呼\"}],\"messages\":
[{\"role\":\"user\",\"content\":\"帮我查一下ES集群的监控数据\"},
{\"role\":\"assistant\",\"content\":\"好的，我来帮你查看ES集群的监控数据\"},
{\"role\":\"intention\",\"content\":\"knowledge_search\"}]}"
}
```

响应示例

方式一（核心结果）：

```
{
  "completion": [
    {
      "result": "knowledge_search"
    }
  ]
}
```

方式二（完整响应）：

```
{
  "completion": [
    {
      "result": "{\"object\":\"intent.recognition\",\"model\":\"youtu-
intent\",\"confidence\":1,\"selected_candidates\":
[{\"id\":\"knowledge_search\",\"score\":1}],\"usage\":
{\"prompt_tokens\":1180,\"completion_tokens\":14,\"total_tokens\":1194}}
"
    }
  ]
}
```

使用建议

- **候选项描述要具体：** `description` 是模型判断的主要依据。"包含腾讯 2020-2024 年季度/年度财报数据，涵盖营收、利润等财务指标"这类具体描述，效果远好于"财报相关"。

- **务必包含闲聊候选项**：否则"你好""谢谢"这类输入会被强行归类到某个知识库，触发无意义的检索。
- **利用置信度兜底**：置信度较低说明模型无法明确区分，此时建议回退到检索全部知识库，而非采纳一个不确定的路由结果。
- **多轮场景传入 intention 历史**：将上一轮的意图 ID 以 `intention` 角色写入 `messages`，有助于模型在追问场景下保持意图连续。

LLM 对话（Completion）

LLM 对话服务提供大语言模型文本生成能力。

支持的模型

模型名称：

- `deepseek-r1`
- `deepseek-v3`

创建推理端点

LLM 也支持两种协议方式，直接用 openai 协议：

Kibana Dev Tools

```
{
  "service": "openai",
  "service_settings": {
    "api_key": "sk-****",
    "model_id": "deepseek-v3",
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/chat/completions"
  }
}
```

使用 custom：

Kibana Dev Tools

```
PUT _inference/completion/tencent-llm
{
  "service": "custom",
  "service_settings": {
```

```
    "url":
"https://bj.aisearch.tencentelasticsearch.com/v1/chat/completions",
    "headers": {
      "Authorization": "Bearer ${api_key}",
      "Content-Type": "application/json"
    },
    "request": "{\"model\": \"deepseek-v3\", \"stream\": false,
\"messages\": [{\"role\": \"user\", \"content\": ${input}}]}",
    "response": {
      "json_parser": {
        "completion_result": "$.choices[*].message.content"
      }
    },
    "secret_parameters": {
      "api_key": "sk-****"
    }
  }
}
```

> **⚠ 注意：**
> `request` 模板中的 `stream` 必须为 `false`。ES Inference API 的 `completion` 任务通过 JSONPath 解析完整响应体，无法处理 SSE 流式返回。

调用推理

Kibana Dev Tools

```
POST _inference/completion/tencent-llm
{
  "input": "用一句话介绍 Elasticsearch"
}
```

响应示例

```
{
  "completion": [
    {
```

```
      "result": "Elasticsearch 是一个基于 Lucene 的分布式搜索与分析引擎，擅长处
理海量数据的近实时检索。"
    }
  ]
}
```

推理端点管理

查看推理端点

Kibana Dev Tools

```
GET _inference/text_embedding/tencent-embedding
```

查看全部推理端点：

```
GET _inference/_all
```

删除推理端点

Kibana Dev Tools

```
DELETE _inference/text_embedding/tencent-embedding
```

若该端点已被 `semantic_text` 字段或 Ingest Pipeline 引用，删除操作会被拒绝。如需强制删除，添加 `force` 参数：

```
DELETE _inference/text_embedding/tencent-embedding?force=true
```

> ⚠ **注意：**
> 强制删除后，引用该端点的写入和查询操作将失败，请谨慎操作。建议先用 `dry_run` 参数确认引用情况。

推理端点速查表

服务	推荐 inference_id	task_type	接口路径	json_parser 核心字段

文本向量化	`tencent-embedding`	`text_embedding`	`/v1/embeddings`	`text_embeddings: $.data[*].embedding[*]`
重排序	`tencent-rerank`	`rerank`	`/v1/rerank`	`relevance_score: $.results[*].relevance_score`
精排	`aisearch-post-rerank`	`rerank`	`/v1/post-rerank`	`relevance_score: $.results[*].relevance_score`
搜索改写	`aisearch-query-rewrite`	`completion`	`/v1/query/rewrite`	`completion_result: $.queries`
多轮改写	`aisearch-multi-round-rewrite`	`completion`	`/v1/query/multi-round-rewrite`	`completion_result: $.content`
意图识别	`aisearch-intent-recognition`	`completion`	`/v1/intent/recognition`	`completion_result: $.selected_candidates`
LLM 对话	`tencent-llm`	`completion`	`/v1/chat/completions`	`completion_result: $.choices[*].message.content`

注意事项

- **模型与端点一一对应：** `request` 模板中的 `model` 字段在创建端点时即固定，调用时无法动态切换。如需使用多个模型，请分别创建推理端点。
- **占位符不加引号：** `${input}` 、 `${query}` 等占位符在 `request` 模板中不能被引号包裹，否则会被当作字符串字面量而非注入点。
- **复杂参数需打包为 JSON 字符串：** 多轮改写的 `messages` 、意图识别的 `candidates` 无法通过 ES 模板变量直接传递，需打包为 JSON 字符串通过 `${input}` 传入。
- **查询分析模型的超时：** 若在同步检索链路中串联查询分析多个能力，请评估总耗时是否满足业务要求，并为每个环节设计降级策略（如模型超时时跳过该环节，使用原始 query 继续检索）。
- **限流：** 原子服务对每个模型设有默认限流。在 `semantic_text` 批量写入场景下，建议通过 `batch_size` 控制单次请求的输入条数，避免触发限流。

版权所有：腾讯云计算（北京）有限责任公司

- **计费**：通过 ES Inference API 调用原子服务会产生原子模型服务费用，按资源包或按量计费。查询分析与精排均基于 LLM，会消耗 token，在检索链路中全量开启会显著增加成本，建议按业务价值选择性启用。

相关文档

- OpenAI 协议调用：通过 OpenAI 兼容协议调用文本向量化、重排序、精排、多模态向量化服务。
- 资源管理 > API Key：API Key 的创建与管理。

OpenAI API 调用原子服务

最近更新时间：2026-08-10 15:22:02

本文介绍如何通过 OpenAI 兼容协议调用智能搜索开发的原子模型服务。原子服务的接口设计兼容 OpenAI API 风格，可直接使用 OpenAI 官方 SDK 或任意 HTTP 客户端接入，无需依赖 Elasticsearch 实例。每个模型服务都有默认限频，如有更多需求，请 联系我们 。

概述

支持通过 OpenAI 协议调用的服务如下，具体模型详情请参见： 原子服务概览

服务	接口路径	说明
文本向量化	`POST /v1/embeddings`	将文本转换为稠密向量
重排序	`POST /v1/rerank`	对候选文档按相关性重新排序
多模态向量化	`POST /v1/multimodal-embeddings`	将文本和图片映射到统一向量空间

适用场景：

- 业务代码需要独立调用向量化、重排序能力，不经过 Elasticsearch。
- 已有基于 OpenAI SDK 的代码，希望以最小改动切换到腾讯云原子服务。
- 需要使用多模态向量化能力。

前提条件

- 已开通原子服务。
- 已获取 API Key。API Key 可在 腾讯云 ES 控制台 的 **资源管理 > API Key** 页面创建和管理。

通用说明

服务地址

项目	说明
Base URL	`https://bj.aisearch.tencentelasticsearch.com`
协议	HTTP/HTTPS
Content-Type	`application/json`
字符编码	UTF-8

超时时间	默认 60 秒

认证方式

所有接口均需在请求头中携带 Bearer Token：

```
Authorization: Bearer sk-<your-api-key>
```

错误响应格式

所有错误响应遵循统一格式：

```
{
  "error": {
    "message": "错误描述信息",
    "type": "错误类型",
    "code": "错误码"
  }
}
```

错误码列表

HTTP 状态码	code	type	说明
401	`missing_api_key`	`invalid_request_error`	缺少 Authorization 请求头
401	`invalid_api_key`	`invalid_request_error`	Token 无效或签名校验失败
403	`account_disabled`	`invalid_request_error`	账号已被禁用
403	`ip_blocked`	`invalid_request_error`	IP 地址被黑名单拦截
403	`ip_not_allowed`	`invalid_request_error`	IP 地址不在白名单中
403	`UnsupportedOperati`	`permission_error`	原子服务未开通

版权所有：腾讯云计算（北京）有限责任公司

	`on`		
429	`LimitExceeded`	`rate_limit_error`	免费额度用完
429	`RequestLimitExceeded`	`rate_limit_error`	后端限流触发
429	`rate_limit_exceeded`	`invalid_request_error`	请求频率超限

限流响应头

每个成功的请求响应中会包含以下限流信息头（OpenAI 兼容）：

响应头	说明	示例
`X-RateLimit-Limit-Requests`	令牌桶容量（最大突发请求数）	`20`
`X-RateLimit-Remaining-Requests`	当前剩余可用令牌数	`19`
`X-RateLimit-Reset-Requests`	令牌恢复时间	`50ms` 或 `1.0s`

文本向量化 – POST /v1/embeddings

将文本转换为向量表示。兼容 OpenAI Embeddings API 格式。

支持的模型

模型名称：

- `bge-base-zh-v1.5`
- `bge-large-zh-v1.5`
- `bge-m3`
- `Conan-embedding-v1`
- `KaLM-embedding-multilingual-mini-v1`
- `Qwen3-Embedding-0.6B`

请求参数

参数	类型	必填	说明
`model`	string	是	模型名称
`input`	string | string[]	是	输入文本，支持单条字符串或字符串数组（批量）

请求示例

单次请求：

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "bge-m3",
    "input": "你好，这是一个测试"
  }' | jq .
```

批量请求：

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "bge-m3",
    "input": ["人工智能正在改变世界", "深度学习是机器学习的一个子领域"]
  }' | jq .
```

响应示例

```
{
  "object": "list",
  "data": [
    {
      "object": "embedding",
      "index": 0,
      "embedding": [0.0123, -0.0456, ...]
    }
  ],
  "model": "bge-m3",
  "usage": {
```

```
    "prompt_tokens": 12,
    "total_tokens": 12
  }
}
```

> **说明：**
> 批量请求时，响应 `data` 数组中的元素顺序与请求 `input` 数组一致，也可通过 `index` 字段对应。

重排序 – POST /v1/rerank

对候选文档相对于查询的相关性进行重新排序，通常用于检索结果的重排环节。

支持的模型

模型名称：

- `bge-reranker-large`
- `bge-reranker-v2-m3`

请求参数

参数	类型	必填	说明
`model`	string	是	模型名称
`query`	string	是	查询文本
`documents`	string[]	是	候选文档列表
`top_n`	number	否	返回前 N 个结果，默认返回全部
`return_documents`	boolean	否	是否在结果中返回文档内容，默认 `false`

请求示例

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/rerank \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "bge-reranker-v2-m3",
```

```
    "query": "What is artificial intelligence",
    "documents": ["Artificial intelligence is a branch of computer
science", "The weather is nice today", "Machine learning is the core
technology of AI"],
    "top_n": 3,
    "return_documents": true
  }' | jq .
```

响应示例

```
{
    "object": "rerank",
    "results": [{
        "index": 0,
        "relevance_score": 0.9992678761482239,
        "document": {
            "text": "Artificial intelligence is a branch of computer
science"
        }
    }, {
        "index": 2,
        "relevance_score": 0.939024806022644,
        "document": {
            "text": "Machine learning is the core technology of AI"
        }
    }, {
        "index": 1,
        "relevance_score": 0.00007484622619813308,
        "document": {
            "text": "The weather is nice today"
        }
    }],
    "model": "bge-reranker-large",
    "usage": {
        "total_tokens": 218
    }
}
```

> **说明：**
> `results` 数组按 `relevance_score` 降序排列，`index` 字段对应请求中 `documents` 数组的原始下标。若 `return_documents` 为 `false`（默认值），响应中不包含 `document` 字段，需通过 `index` 自行映射回原文档。

多模态向量化 - POST /v1/multimodal-embeddings

将文本和图片转换为统一向量空间中的向量表示，适用于以文搜图、以图搜图等跨模态检索场景。

支持的模型

- 模型名称：
- `WeCLIPv2-Base`
- `WeCLIPv2-Large`

请求参数

参数	类型	必填	说明
`model`	string	是	模型名称
`texts`	string[]	否*	文本数组
`image_url`	string[]	否*	图片 URL 数组
`image_data`	string[]	否*	base64 图片数组，格式：`data:<MIME>;base64,<内容>`

> **说明：**
> 该接口使用自定义格式，不兼容 OpenAI 标准的 input 数组格式，必须使用顶层字段 `texts`、`image_url`、`image_data` 三者至少传一个。`image_data` 和 `image_url` 不能同时传。

请求示例

纯文本模式：

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/multimodal-embeddings \
```

```
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "WeCLIPv2-Large",
    "texts": ["一只可爱的猫咪", "美丽的风景照"]
  }' | jq .
```

文本 + 图片 URL 混合模式：

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/multimodal-
embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "WeCLIPv2-Large",
    "texts": ["这是一段文本"],
    "image_url": ["https://example.com/image.png"]
  }' | jq .
```

文本 + 图片 base64 混合模式：

cURL

```
curl -s https://bj.aisearch.tencentelasticsearch.com/v1/multimodal-
embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer sk-****" \
  -d '{
    "model": "WeCLIPv2-Base",
    "texts": ["一只猫"],
    "image_data": [
      "..."
    ]
  }' | jq .
```

响应示例

```
{
  "object": "multimodal_embedding",
  "data": {
    "text_embeddings": [
      {
        "object": "embedding",
        "index": 0,
        "embedding": [0.0123, -0.0456, ...]
      }
    ],
    "image_embeddings": [
      {
        "object": "embedding",
        "index": 0,
        "embedding": [0.0789, -0.0321, ...]
      }
    ]
  },
  "model": "WeCLIPv2-Large",
  "usage": {
    "total_tokens": 5,
    "total_images": 1
  }
}
```


与文本向量化接口不同，多模态接口的向量位于 `data.text_embeddings` 和 `data.image_embeddings` 两个数组中，而非顶层 `data` 数组。

文本向量与图片向量位于同一向量空间，可直接计算余弦相似度实现跨模态检索。实际生产场景中，图片向量通常预先计算并写入 Elasticsearch 的 `dense_vector` 字段，检索时只需实时计算查询文本的向量，再通过 kNN 检索完成匹配。

接口速查表

接口路径	方法	支持的模型	响应格式	说明
`/v1/embeddings`	POST	bge-base-zh-v1.5、bge-large-zh-v1.5、bge-m3、Conan-embedding-v1、KaLM-embedding-multilingual-mini-v1、Qwen3-Embedding-0.6B	JSON	文本向量化

`/v1/rerank`	POST	bge-reranker-large、bge-reranker-v2-m3	JSON	重排序
`/v1/multimodal-embeddings`	POST	WeCLIPv2-Base、WeCLIPv2-Large	JSON	多模态向量化

注意事项

- **API Key 安全**：不要将 API Key 硬编码在前端代码或提交到代码仓库。建议由后端服务代理调用，或通过环境变量、密钥管理服务注入。
- **限流处理**：向量化和重排序模型默认限流为 20 QPS。触发限流时接口返回 429，建议在客户端实现指数退避重试，并结合响应头中的 `X-RateLimit-Reset-Requests` 控制重试间隔。
- **批量优先**：文本向量化接口支持数组批量输入，同一批文本合并为一次请求可显著降低请求次数，避免触发限流。
- **多模态接口响应结构特殊**：多模态向量化的响应结构与其他接口不同，向量嵌套在 `data.text_embeddings` 和 `data.image_embeddings` 中，解析时请注意区分。
- **计费**：调用原子服务会产生原子模型服务费用，按实际 Token 量计费。计费详情请参见 智能搜索开发定价 。

相关文档

- ES Inference API 调用：通过 Elasticsearch Inference API 调用文本向量化、重排序、LLM 对话，以及查询分析、文档解析、文本切片服务。
- 资源管理 > API Key：API Key 的创建与管理。

原子服务监控

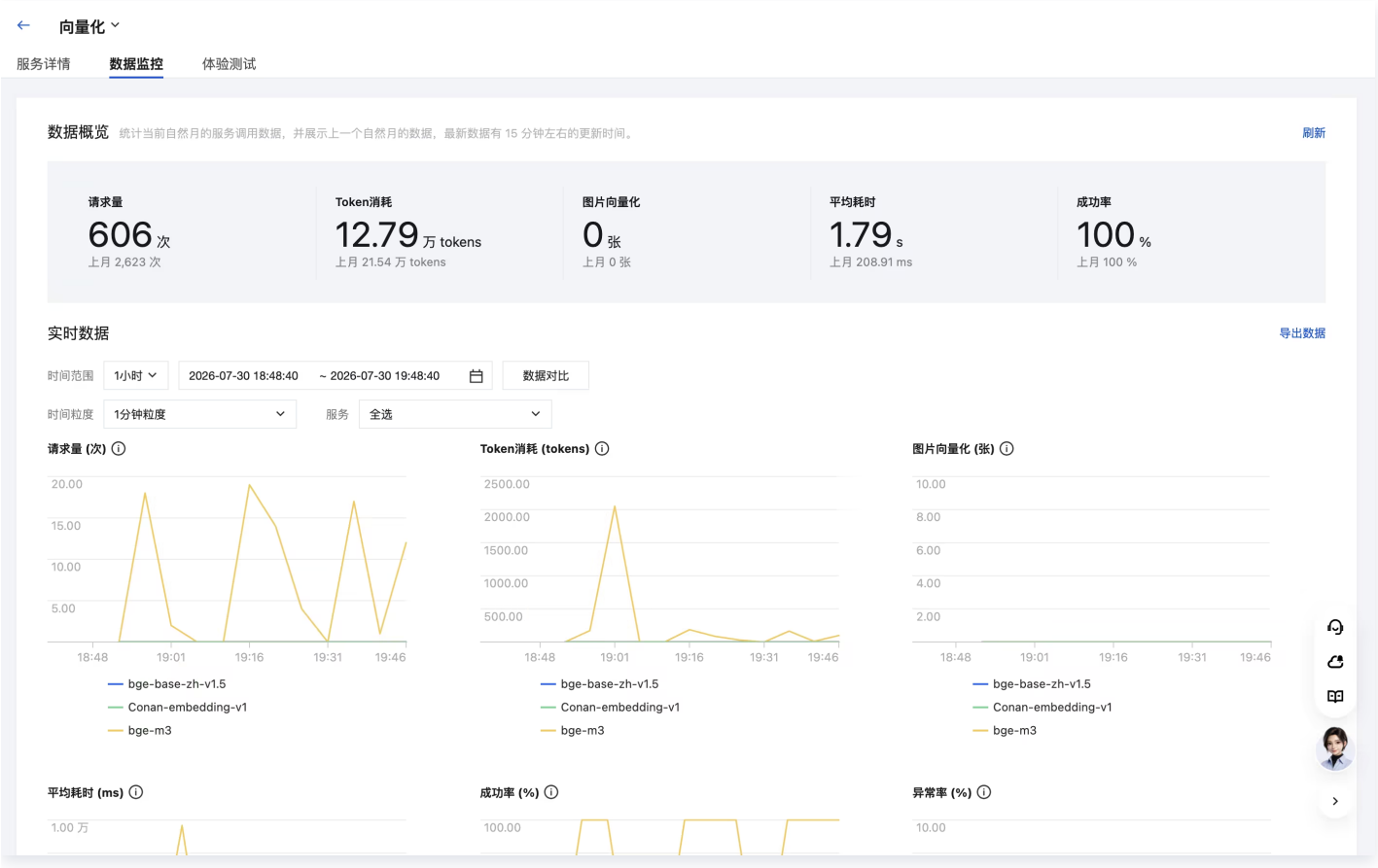
最近更新时间：2026-08-10 15:22:02

数据监控用于查看原子服务的调用情况，包括请求量、Token 消耗、耗时、成功率等指标，帮助您了解服务用量、评估计费成本并及时发现异常。

进入数据监控

1. 登录 腾讯云 ES 控制台 。
2. 在左侧导航栏选择 **智能搜索开发 > 原子服务**，进入原子服务列表页。
3. 单击目标服务名称，进入服务详情页。
4. 选择 **数据监控** 标签页。

页面结构



数据监控页面分为两个区域：

区域	说明
数据概览	统计当前自然月的服务调用数据，并对比上一个自然月的数据

实时数据	按所选时间范围和粒度展示各指标的趋势图表

数据概览

数据概览统计当前自然月的服务调用数据，并展示上一个自然月的数据作为对比，便于快速判断用量变化趋势。

指标	说明
请求量（次）	指定时间范围内推理服务接收到的请求总数
Token 消耗（tokens）	推理服务处理请求所消耗的 Token 总量，直接关联计费，LLM 会有输入和输出 Token 之分
解析页数（页）	涉及文档解析的处理数量
图片向量化（张）	涉及图片向量化处理的请求次数
平均耗时（ms）	推理服务处理单次请求的平均响应时间
成功率（%）	推理请求成功处理的比例

实时数据

实时数据以图表形式展示各指标随时间的变化趋势，用于定位具体时间点的异常。

筛选条件

筛选项	说明
时间范围	选择查询的时间跨度，如 1 小时。也可通过右侧日期控件自定义起止时间
时间粒度	数据聚合的时间精度，如 1 分钟粒度。时间范围越长，建议粒度越粗
服务	筛选具体的模型服务，默认 **全选**，即展示该原子服务下所有模型的数据

提供 **导出数据** 功能，可将当前筛选条件下的监控数据导出，用于离线分析或留档。
图表区上方提供 **数据对比** 按钮，可将当前时间段与另一时间段的数据叠加展示，便于对比变更前后的效果（如调整模型或优化调用逻辑后的性能变化）。

监控指标

指标	说明
请求量（次）	指定时间范围内推理服务接收到的请求总数
Token 消耗	推理服务处理请求所消耗的 Token 总量，直接关联计费，LLM 会有输入和输

（tokens）	出 Token 之分
解析页数（页）	涉及文档解析的处理数量
图片识别（次）	在调用文档解析接口时，开启图片识别次数
图片向量化（张）	涉及图片向量化处理的请求次数
平均耗时（ms）	推理服务处理单次请求的平均响应时间
成功率（%）	推理请求成功处理的比例
异常率（%）	推理请求处理失败或超时的比例

使用建议

- **成本监控**：以 Token 消耗为主要观察指标，若发现某个模型的 Token 消耗异常偏高，检查是否存在重复调用或输入文本过长的问题。
- **性能排查**：平均耗时突增时，先看请求量是否同步上涨。若请求量平稳但耗时上升，可能是单次输入内容变长；若两者同步上涨，则可能触及限流。
- **异常定位**：成功率下降或异常率上升时，将时间粒度调细（如 1 分钟粒度）定位到具体时间点，再结合业务侧日志排查。持续异常需检查 API Key 是否有效、是否触发限流、资源包是否耗尽。
- **模型选型参考**：通过按模型拆分的耗时和 Token 消耗曲线，对比不同模型的性能与成本，为生产环境的模型选择提供数据依据。
- **变更效果验证**：调整模型或优化调用逻辑后，使用 **数据对比** 功能叠加变更前后的数据，量化评估改动效果。

版权所有：腾讯云计算（北京）有限责任公司

资源管理

API Key

最近更新时间：2026-08-10 15:22:02

API Key 是调用原子服务和应用服务的身份凭证。通过 OpenAI 协议或 ES Inference API 调用原子服务时，均需在请求中携带 API Key 完成身份认证。本文介绍如何创建、查看、管理 API Key。

进入 API Key 管理页

1. 登录 腾讯云 ES 控制台 。
2. 在左侧导航栏选择 **智能搜索开发 > 资源管理**。
3. 选择 **API Key** 标签页。

使用限制

项目	限制
数量上限	主账号与子账号累计最多创建 **100 个**
可见范围	主账号与子账号均可创建，且 **互不可见**
权限范围	具备原子服务与应用服务的完整调用权限

访问域名

页面上方展示原子服务的访问地址，调用接口时需使用该地址作为 Base URL。

项目	说明
公网地址	形如 `https://bj.aisearch.tencentelasticsearch.com`

创建 API Key

1. 在 API Key 页面，单击 **创建 API Key**。
2. 创建成功后，新的 API Key 出现在列表中。

API Key

列表包含以下信息：

字段	说明
API Key	密钥内容，默认脱敏展示，形如 `sk-eyJh****c6be`
创建时间	API Key 的创建时间
备注	用途说明。单击铅笔图标可编辑
操作	**查看**、**删除**

备注命名建议

备注是区分多个 API Key 的唯一依据，建议在创建后立即填写，包含以下信息：

- **使用方**：哪个业务系统或团队在用，如"订单系统""算法团队"。
- **环境**：生产、预发、测试。
- **负责人**：便于后续排查归属。

例如："生产环境-商品搜索-张三"。默认无备注的 Key 在数量增多后极难辨识，会给后续的密钥轮换和故障排查带来困难。

查看 API Key

列表中的 API Key 默认脱敏展示。单击操作列的 **查看** 可获取完整的密钥内容。完整的 API Key 具备原子服务与应用服务的完整调用权限。请仅在必要时查看，并避免在共享屏幕、录屏或截图中暴露。

删除 API Key

单击操作列的 **删除**，可删除不再使用的 API Key。建议按以下顺序执行删除，避免业务中断：

1. 在 **数据监控** 页面确认该 Key 对应的调用量已降为 0。
2. 检查业务代码、配置文件、环境变量、ES 推理端点中是否仍引用该 Key。
3. 确认无引用后再执行删除。

安全须知

API Key 是敏感凭证，泄露会导致未授权访问和财产损失。请遵循以下原则：

请勿在客户端存储

请勿将 API Key 直接嵌入前端应用或客户端代码。 前端代码可被任意查看，移动应用可被反编译，一旦嵌入即等同于公开。正确做法是由后端服务代理调用，客户端请求后端，后端携带 API Key 请求原子服务。

请勿通过不安全渠道传递

请勿通过即时通讯工具、邮件、明文文档等方式分享 API Key。团队内部共享建议使用密钥管理服务或加密的凭证管理工具。

请勿提交到代码仓库

将 API Key 写入代码后提交到 Git，即使后续删除，历史提交中依然可查。推荐通过环境变量注入：

```
export AISEARCH_API_KEY="sk-your-actual-api-key"
```

代码中通过环境变量读取，并将包含密钥的本地配置文件加入 `.gitignore` 。

在 ES 中使用时配置为加密字段

通过 ES Inference API 调用原子服务时，务必将 API Key 配置在推理端点的 `secret_parameters` 中。该字段由 Elasticsearch 加密存储，查询端点信息时不会返回明文。若写在 `headers` 或 `request` 的明文位置，任何有权查询推理端点的用户都能看到密钥。详情请参见 ES Inference API 调用 。

按用途隔离并定期轮换

- **按用途隔离**：为不同业务系统、不同环境（生产/测试）创建独立的 API Key。这样在某个 Key 泄露时，只需吊销该 Key，不影响其他业务。
- **定期轮换**：建议定期更换 API Key。轮换时先创建新 Key 并完成业务切换，确认新 Key 调用正常后再删除旧 Key，避免服务中断。

版权所有：腾讯云计算（北京）有限责任公司

设置

最近更新时间：2026-08-10 15:22:02

设置页面用于管理原子服务的计费模式和 COS 访问授权。本文介绍两项配置的作用、开通方式及注意事项。

进入设置页

1. 登录 腾讯云 ES 控制台 。
2. 在左侧导航栏选择 **智能搜索开发 > 资源管理**。
3. 选择 **设置** 标签页。

设置

原子服务后付费

后付费是资源包耗尽后的兜底计费方式。开通后，超出资源包额度的调用量将按实际用量结算，保证业务连续性。

配置说明

项目	说明
开关	**开通后付费**，通过右侧开关控制
生效时间	变更后约 **10 秒** 生效
变更频率	每月仅能变更 **5 次**

开通方式

在 **原子服务后付费** 卡片中，打开 **开通后付费** 开关即可，在开通原子服务时默认打开后付费。

是否需要开通

场景	建议
生产环境、业务不可中断	**建议开通**。作为资源包耗尽的兜底，避免因额度不足导致服务中断
测试环境、需严格控制成本	可不开通。资源包耗尽后调用直接失败，形成硬性成本上限
用量波动较大、难以预估	**建议开通**。资源包覆盖基础用量，突增部分由后付费承接

COS 访问授权

COS 访问授权用于允许智能搜索访问您的对象存储（COS）资源。

涉及的功能

授权后，以下功能可正常使用：

- **文档解析**：读取存放在 COS 中的待解析文件
- **文本切片**：读取待切片的文档内容
- **图片 Embedding**：读取需向量化的图片
- **效果评测**：读取评测数据集文件

授权方式

在 **COS 访问授权** 卡片中，单击授权入口完成授权。授权成功后，卡片右上角显示 **已授权**。若未完成授权，您需要在 COS 对象存储控制台手动打开对应存储桶的全公开可读权限，才能让上述功能正常读取文件。

推荐使用授权而非公开读

从安全角度出发，**强烈建议使用 COS 访问授权，而不是将存储桶设为公开可读**：

方式	安全性	说明
COS 访问授权（推荐）	高	仅授权智能搜索服务访问，权限可控、可回收
存储桶公开可读	低	任何知晓 URL 的人都能下载文件，存在数据泄露风险

若您的文档包含内部资料、客户数据等敏感内容，公开可读会导致这些文件在互联网上可被任意访问。完成 COS 访问授权后，应及时关闭存储桶的公开读权限。

版权所有：腾讯云计算（北京）有限责任公司