边缘安全加速平台 EO

边缘推理

腾讯云

【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

边缘推理
边缘推理概述

最近更新时间：2026-03-26 16:58:11

边缘安全加速平台 EO 所提供的边缘推理服务是基于 EdgeOne 边缘云分布式节点 + Serverless 弹性架构构建的高性能 AI 推理解决方案，核心目标是解决传统云端推理 “高延迟、高带宽成本” 和本地部署 “难运维、无弹性” 的痛点，为需要实时响应、数据本地化处理的 AI 业务，提供 “就近调度、弹性伸缩、免运维管理、安全合规” 的推理算力支持。

核心价值

1. **低延迟推理：就近响应，毫秒级反馈**
   - 核心亮点：利用 EdgeOne 全球边缘节点，使用户业务流量就近接入节点，推理响应延迟低至毫秒级。
   - 客户价值：满足实时性要求高的场景，避免云端传输导致的延迟损耗，提升业务响应速度和用户体验。
2. **弹性伸缩：按需分配，降本增效**
   - 核心亮点：基于 Serverless 架构，支持弹性扩缩容，根据推理请求量自动调整算力资源，闲时释放资源，忙时无缝扩容，无需预留冗余算力。
   - 客户价值：按实际算力使用时长计费，避免本地部署的硬件闲置成本，中小客户无需投入巨额硬件采购费用，大型客户可灵活应对流量峰值。
3. **免运维管理：简化部署，聚焦核心业务**
   - 核心亮点：提供全托管式推理服务，平台自动完成边缘节点运维、算力调度、模型部署、版本更新、故障自愈，开发者无需关注底层资源。
   - 客户价值：降低 AI 业务落地门槛，减少运维团队投入（无需专职运维人员维护边缘节点），缩短产品上线周期（模型上传到服务启动仅需30分钟）。
4. **安全防护：全栈护航，保障 API 服务稳定**
   - 核心亮点：针对推理服务 API 打造全栈安全防护体系，覆盖四层和七层防护能力；其中四层防护支持 DDoS 攻击防御，七层防护集成 WAF，可精准识别并拦截 SQL 注入、XSS 跨站脚本、恶意爬虫等应用层攻击。
   - 客户价值：避免因 API 被攻击导致的服务中断、数据泄露或算力资源被恶意占用，保障推理服务7×24小时稳定运行，降低业务运营风险，尤其适配金融、政务等对服务安全性要求极高的行业场景。

快速开始

1. 部署您的第一个模型服务—快速指引 。
2. 了解边缘推理的计费方式 —边缘推理费用 。

快速指引

最近更新时间：2026-03-26 16:58:11

本文档旨在引导您完成在边缘安全加速平台 EO 的边缘推理中部署开源或者自有模型及调用 API 的完整过程。您将了解如何创建推理服务、管理服务凭证、发起推理请求以及进行基础的故障排查，从而帮助您将边缘推理的 AI模型能力集成到您的应用中。我们将以部署 **Llama-3.2-3B-Instruct** 大语言模型为示例，演示从环境准备到模型上线调用的全流程。

说明：

- 边缘推理当前仅支持企业版套餐，同时需要申请白名单后使用，如您有需要请联系商务或 联系我们 。
- 本指南以 Linux/macOS 环境为例，Windows 用户请在 WSL 或 Docker Desktop 环境下操作。

前置条件

在开始之前，请确保您已准备好以下内容：

准备项	说明
腾讯云账号	已完成实名认证的腾讯云账号
企业版套餐	已订购 EdgeOne 企业版套餐
Docker 环境	本地已安装 Docker（建议 Docker 20.10+），用于构建和推送镜像
腾讯云容器镜像服务（TCR）	已开通腾讯云容器镜像服务（个人版即可），用于存储自定义镜像
GPU 服务器（可选）	如需在本地测试镜像运行，需具备 NVIDIA GPU 的机器；仅做构建则不需要

创建推理服务并调用

以下步骤演示了如何将一个模型部署为在线服务，并进行调用。此流程同时适用于传统模型（如 OCR、语音识别）和大语言模型（LLM）或者文生图模型等。

步骤一：登录腾讯云控制台

1. 打开浏览器访问 腾讯云控制台 ，使用您的腾讯云账号登录。
2. 在控制台顶部搜索栏中输入 **边缘安全加速平台EO**，或在左侧导航中找到 **边缘安全加速平台EO**，点击进入。
3. 在 EdgeOne 控制台左侧导航中，找到并点击**服务总览**菜单，选择**边缘推理**进入边缘推理管理页面。

步骤二：准备自定义镜像（以 Llama-3.2-3B 为例）

边缘推理通过容器化方式部署模型。您需要编写一个 Dockerfile，将模型文件、推理框架和依赖项打包成一个完整的 Docker 镜像。

2.1 创建项目目录

在本地创建一个工作目录：

```
mkdir llama3-edge-inference && cd llama3-edge-inference
```

2.2 编写 Dockerfile

在项目目录中创建 `Dockerfile` 文件。以下示例使用 vLLM 作为推理框架，从 ModelScope 下载 Llama-3.2-3B-Instruct 模型：

```
# ============================================================
# Dockerfile for Llama3.2-3B Model with vLLM 0.6.3.post1
# ============================================================

# 使用 vLLM 官方 OpenAI 兼容镜像作为基础镜像
FROM vllm/vllm-openai:v0.6.3.post1

# 设置环境变量，跳过 HuggingFace Hub 在线验证（离线模式）
ENV HF_HUB_OFFLINE=1
ENV HF_HOME=/data/models
ENV TRANSFORMERS_CACHE=/data/models

# 安装 modelscope 用于下载模型（使用国内镜像源加速）
RUN pip3 install --no-cache-dir modelscope -i
https://pypi.tuna.tsinghua.edu.cn/simple

# 创建模型目录并下载模型文件
RUN mkdir -p /data/models/LLM-Research/Llama-3.2-3B-Instruct && \
    python3 -c "from modelscope import snapshot_download;
snapshot_download('LLM-Research/Llama-3.2-3B-Instruct',
local_dir='/data/models/LLM-Research/Llama-3.2-3B-Instruct')"

# 暴露推理服务端口
EXPOSE 8000

# 启动参数说明：
```

```
# --host 0.0.0.0        监听所有网络接口
# --port 8000           服务监听端口
# --model               模型文件路径
# --trust-remote-code   信任远程代码（部分模型需要）
# --dtype half          使用半精度推理，降低显存占用
# --max-model-len 8192  最大上下文长度
CMD ["--host", "0.0.0.0", \
     "--port", "8000", \
     "--model", "/data/models/LLM-Research/Llama-3.2-3B-Instruct", \
     "--trust-remote-code", \
     "--dtype", "half", \
     "--max-model-len", "8192"]
```

关键参数说明：

参数	说明
`FROM vllm/vllm-openai:v0.6.3.post1`	基础镜像已内置 vLLM 推理引擎和 OpenAI 兼容 API 接口
`HF_HUB_OFFLINE=1`	禁用在线模型验证，确保容器在无外网环境下也可启动
`--dtype half`	使用 FP16 半精度推理，Llama-3.2-3B 模型约需 6GB 显存
`--max-model-len 8192`	设置最大上下文窗口为 8192 tokens

> **说明：**
> 请根据您的实际模型和推理框架调整 Dockerfile。如果您使用其他框架（例如 SGLang 等），请参考对应框架的容器化文档。

步骤三：构建 Docker 镜像

3.1 构建镜像

在 Dockerfile 所在目录执行以下命令构建镜像：

```
docker build -t llama3-3b-vllm:v1.0 .
```

构建过程将依次执行以下操作：

1. 拉取 vLLM 基础镜像。

2. 安装 ModelScope 依赖。
3. 下载 Llama-3.2-3B-Instruct 模型文件（约 6GB，请耐心等待）。

> **说明：**
> 首次构建可能需要 10-30 分钟，取决于网络速度。建议在网络稳定的环境下执行。如果因网络问题下载失败，可以重新执行构建命令，Docker 会从上次中断的步骤继续。更多 Docker 教程，您可以查看 官方 Docker 教程 。

3.2 验证镜像（可选）

构建完成后，可以查看镜像信息：

```
docker images | grep llama3-3b-vllm
```

预期输出：

```
llama3-3b-vllm    v1.0    xxxxxxxxxxxx    xx minutes ago    约15GB
```

如果您本地有 GPU 环境，可以运行以下命令测试镜像是否正常工作：

```
docker run --gpus all -p 8000:8000 llama3-3b-vllm:v1.0
```

等待服务启动完成后（日志中出现 `Uvicorn running on http://0.0.0.0:8000` ），在另一个终端窗口执行测试请求：

```
curl http://localhost:8000/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "/data/models/LLM-Research/Llama-3.2-3B-Instruct",
    "messages": [{"role": "user", "content": "Hello!"}],
    "max_tokens": 50
  }'
```

步骤四：推送镜像至腾讯云容器镜像服务（TCR）

构建好的镜像需要上传到腾讯云容器镜像服务（TCR），以便边缘推理平台拉取和部署。

4.1 开通容器镜像服务

如果您尚未开通 TCR，请参见 容器镜像服务快速入门 。

1. 登录 腾讯云容器镜像服务控制台 。
2. 创建一个**命名空间**（如 `edge-inference`）。
3. 在命名空间下创建一个**镜像仓库**（如 `llama3-3b-vllm`）。

4.2 登录 TCR

在本地终端执行以下命令登录 TCR（请替换为您的实际实例信息）：

```
# 个人版 TCR 登录
docker login ccr.ccs.tencentyun.com --username=<腾讯云账号ID>
```

系统会提示输入密码，请输入您在 TCR 控制台中设置的镜像仓库密码。

> **说明：**
> 如果使用企业版 TCR，登录地址格式为 `<实例名>.tencentcloudcr.com`，请参考 TCR 控制台获取具体地址。

4.3 标记并推送镜像

```
# 为镜像打标签（请替换为您的实际仓库地址）
docker tag llama3-3b-vllm:v1.0 ccr.ccs.tencentyun.com/edge-
inference/llama3-3b-vllm:v1.0

# 推送镜像到 TCR
docker push ccr.ccs.tencentyun.com/edge-inference/llama3-3b-vllm:v1.0
```

> **说明：**
> 由于模型镜像较大（约 15GB），推送时间取决于上行带宽，建议在带宽充足的环境下操作。

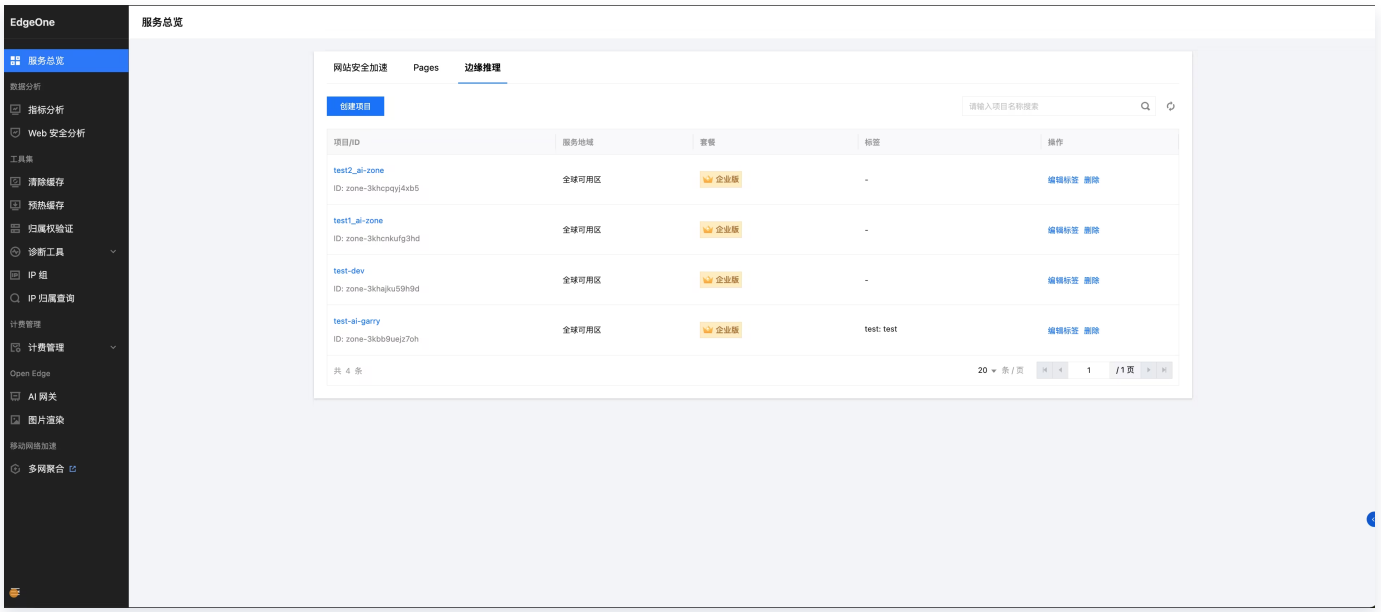
4.4 确认上传成功

登录 TCR 控制台，在对应的镜像仓库中确认镜像已成功上传，可以看到 `v1.0` 标签的镜像。

步骤五：创建项目

项目是边缘推理的一级资源，用于组织和管理多个推理服务。一个项目可以包含多个推理服务，您还可以通过项目绑定标签来实现权限管控。

1. 在 EdgeOne 控制台的边缘推理页面，单击**创建项目** 。



2. 填写项目信息：

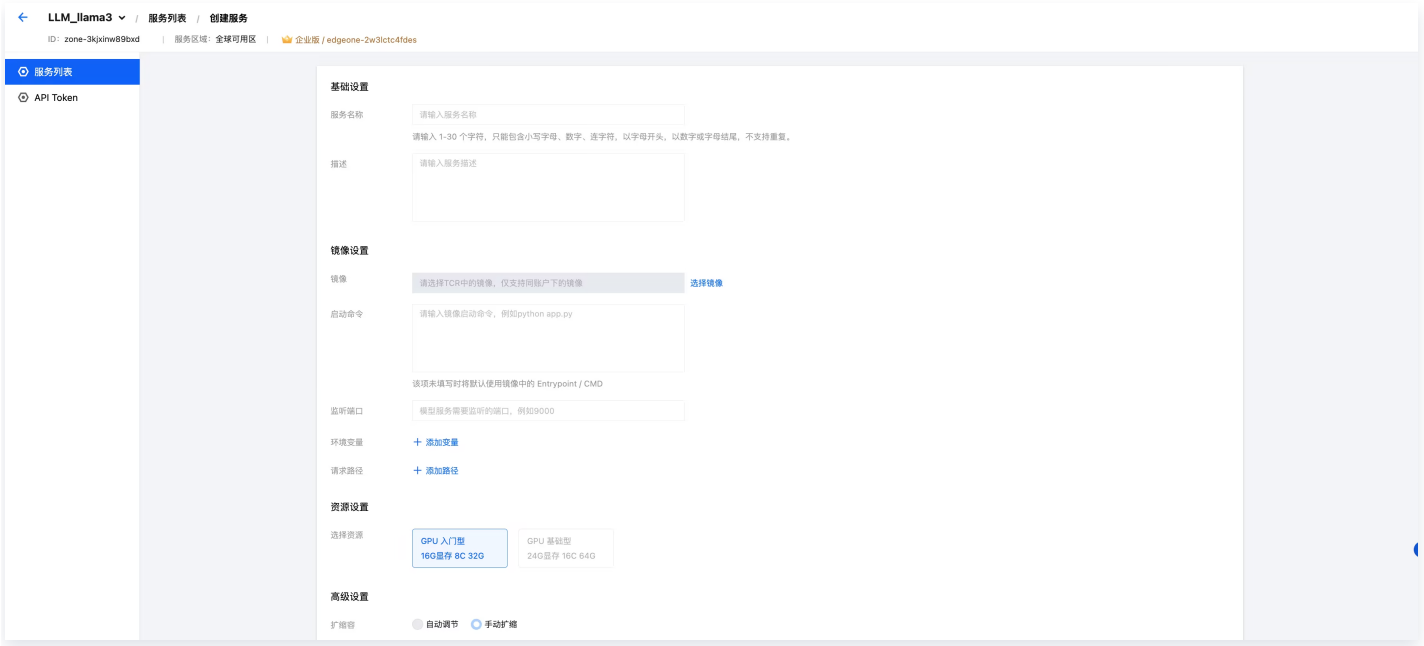
- **项目名称**：输入项目名称（如 `llm-inference-project`）
- **绑定套餐**：选择您已订购的企业版套餐

3. 完成项目创建后，点击项目名称进入项目详情页，即可开始创建推理服务。

步骤六：创建推理服务

服务是边缘推理的核心单元。创建服务意味着将您上传的镜像以容器形式部署到边缘节点，平台将为您提供一个公网可访问的服务地址。

在已创建的项目中，单击**创建服务**。



6.1 基础设置

配置项	说明	示例值
服务名称	服务的唯一标识，创建后无法修改	`llama3-3b-service`
描述	服务的用途说明，最多60字符	`Llama-3.2-3B` 推理服务

6.2 镜像设置

配置项	说明	示例值
镜像	选择您账号下已上传至 TCR 的镜像	`ccr.ccs.tencentyun.com/edge-inference/llama3-3b-vllm:v1.0`
启动命令	容器启动时执行的命令。如不填写，则使用镜像中的 ENTRYPOINT/CMD	留空（使用 Dockerfile 中已定义的 CMD）
监听端口	您的推理服务 HTTP Server 监听的端口	`8000`
环境变量	运行时环境变量配置	变量名：`HF_HOME`，变量值：`/data/models`
请求路径	客户端调用推理服务的 API 路径	`/v1/chat/completions`

关于启动命令：

本示例中 Dockerfile 已通过 `CMD` 设置了启动参数，因此可以留空。如果需要覆盖默认参数，可以填写完整的启动命令，例如：

```
python3 -m vllm.entrypoints.openai.api_server --host 0.0.0.0 --port 8000
--model /data/models/LLM-Research/Llama-3.2-3B-Instruct --trust-remote-
code --dtype half --max-model-len 8192
```

6.3 资源设置

配置项	说明
选择资源	当前提供**入门型**和**基础型**两种 GPU 资源规格，选择后无法更改

> **注意：**
> 所选的实例规格（尤其是 GPU 显存）必须满足模型运行的要求。Llama-3.2-3B 使用 FP16 推理时约需 6GB 显存，请选择显存 ≥ 8GB 的规格，否则可能导致显存溢出（OOM）而部署失败。

6.4 高级设置

配置项	说明	建议配置
扩缩容	• **自动**：根据请求量自动扩缩容 • **手动**：固定实例数，常驻运行并持续收费	建议选择**自动**，节省成本
并发数	单实例的并发请求数上限	对于 LLM 推理，建议设为 `1` - `5`，取决于模型大小和显存

6.5 完成创建

完成以上配置后单击**创建**，系统将开始部署服务。请耐心等待服务状态从「部署中」转为「运行中」，表示服务已就绪。

> **说明：**
> 首次部署需要拉取镜像到边缘节点，由于镜像较大，部署时间可能需要 5-15 分钟。您可以在服务详情页查看部署日志。

步骤七：创建 API Token 并获取服务信息

服务成功运行后，您需要创建 API Token 用于鉴权，并获取服务的访问地址。

7.1 进入服务详情

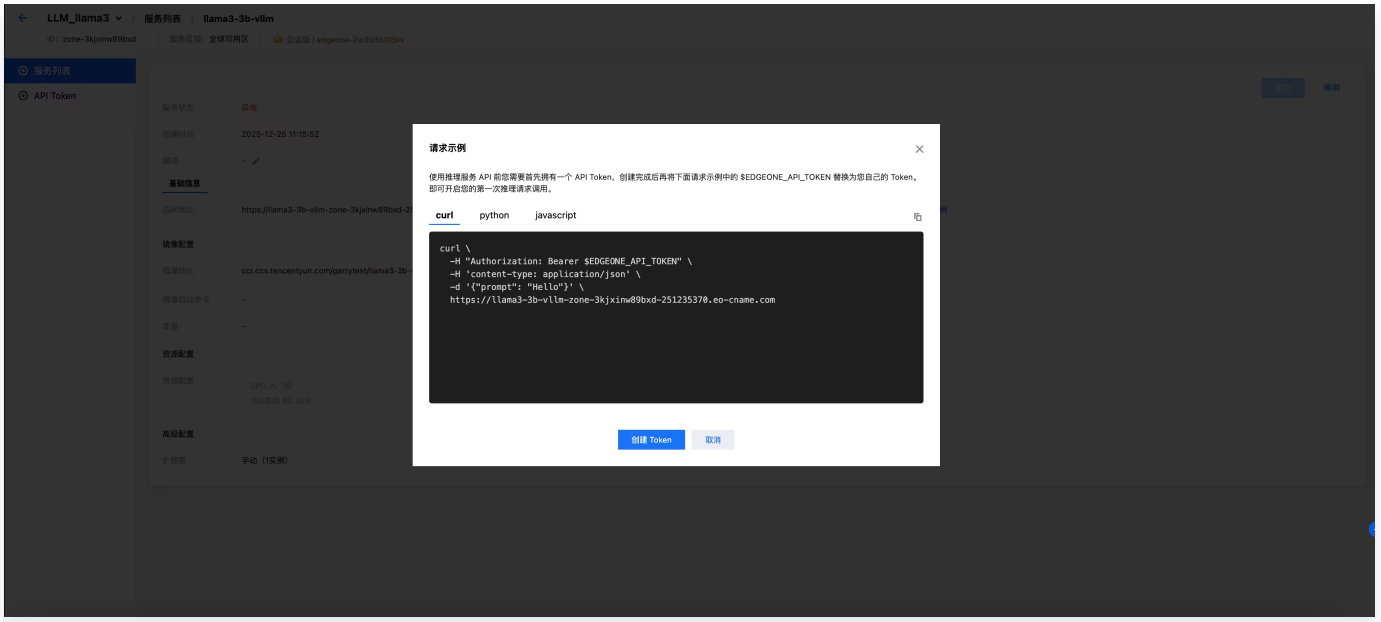
点击服务名称进入服务详情页面，在基础信息中您可以看到：

- **服务状态**：应显示为「运行中」。
- **访问地址**：平台分配的公网访问 URL（如 `https://your-service-id.edgeone-infer.com`）。
- **请求示例**：平台自动生成的调用示例。

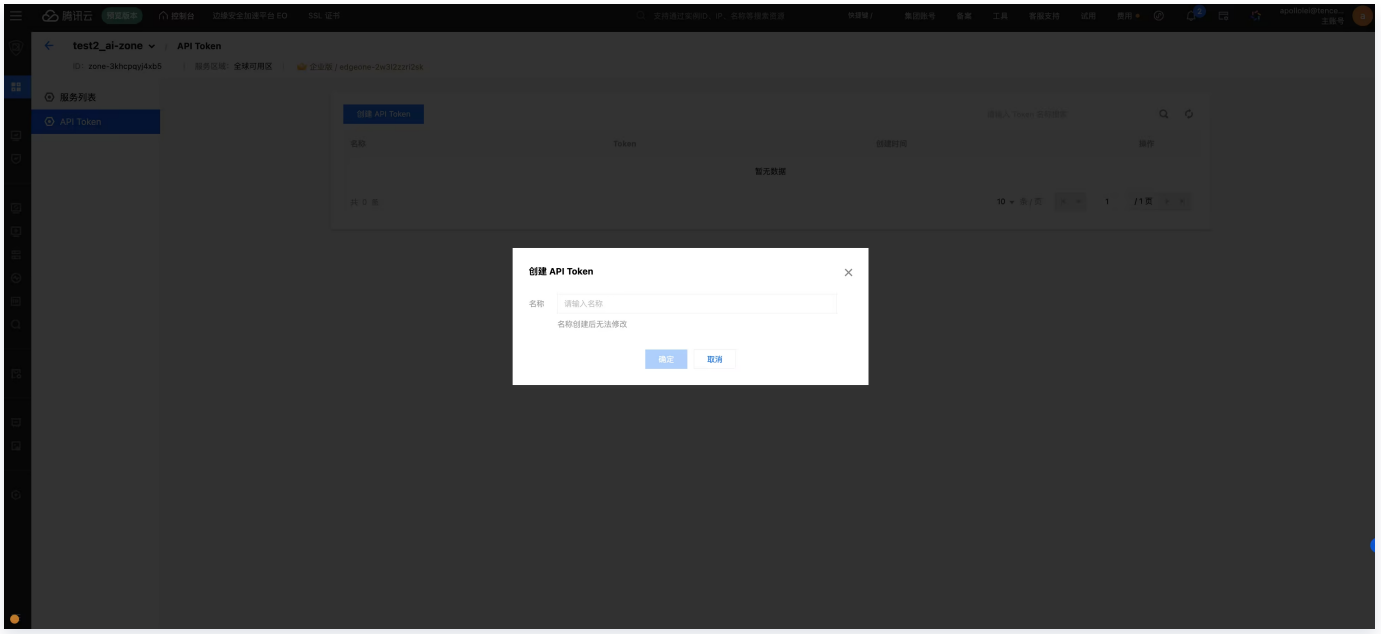
7.2 创建 API Token

边缘推理的所有请求都需要通过 Bearer Token 进行鉴权，因此需要先创建 API Token。

1. 单击服务详情中的**请求示例**。



2. 单击**创建 API Token**。输入 Token 名称（如 `my-first-token`），系统将自动生成一个 API Token。



3. 创建完成后，Token 会自动填入请求示例中。您也可以在左侧菜单的 **API Token** 页面中集中管理所有 Token。

> **说明：**
> API Token 是访问推理服务的重要凭证，请妥善保管。Token 默认以掩码形式显示，点击可复制完整内容。请勿在公开场所泄露您的 Token。

步骤八：调用模型服务

服务部署成功并获取 API Token 后，您可以通过 HTTP API 调用模型。请将以下示例中的 `YOUR_SERVICE_URL` 和 `YOUR_BEARER_TOKEN` 替换为您的实际信息：

```
curl https://YOUR_SERVICE_URL/v1/chat/completions \
  -H "Authorization: Bearer YOUR_BEARER_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "/data/models/LLM-Research/Llama-3.2-3B-Instruct",
    "messages": [
      {"role": "system", "content": "You are a helpful assistant."},
      {"role": "user", "content": "Hello, who are you?"}
    ],
    "max_tokens": 256,
    "temperature": 0.7
  }'
```

预期响应示例：

```
{
  "id": "cmpl-xxxxxxxx",
  "object": "chat.completion",
  "created": 1739260800,
  "model": "/data/models/LLM-Research/Llama-3.2-3B-Instruct",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! I'm Llama, a helpful AI assistant developed
by Meta. I'm here to help you with questions, provide information, and
assist with various tasks. How can I help you today?"
      },
      "finish_reason": "stop"
```

```
    }
  ],
  "usage": {
    "prompt_tokens": 25,
    "completion_tokens": 42,
    "total_tokens": 67
  }
}
```

请求参数说明

参数	类型	必填	说明
`model`	string	是	模型路径，与容器内模型文件路径一致
`messages`	array	是	对话消息列表，包含 `role`（system/user/assistant）和 `content`
`max_tokens`	integer	否	最大生成 token 数，默认由模型决定
`temperature`	float	否	生成温度，范围 0-2，值越高输出越随机，默认 1.0
`top_p`	float	否	核采样参数，范围 0-1，默认 1.0
`stream`	boolean	否	是否启用流式输出，默认 `false`
`stop`	string/array	否	停止生成的标记词

常见问题排查

问题	可能原因	解决方法
服务状态一直显示「部署中」	镜像过大，拉取时间较长	等待 15-30 分钟；检查镜像是否正确上传至 TCR
服务状态显示「部署失败」	GPU 显存不足或镜像启动异常	检查资源规格是否满足模型要求；查看部署日志排查错误

API 请求返回 401	Token 无效或过期	检查 Authorization 头是否正确；确认 Token 格式为 `Bearer <token>`
API 请求返回 502/504	服务未就绪或请求超时	确认服务状态为「运行中」；适当增加超时时间
返回 OOM 错误	显存不足	降低 `--max-model-len` 参数值；选择更高规格的 GPU 资源