一站式智算服务平台

用户使用手册

天翼云科技有限公司

目 录

1 产品介绍

1.1产品定义

一站式智算服务平台是大模型一站式、可视化、全流程 AI 开发训练平台，为用户提供 AI 建模的一站式解决方案。具备开箱即用、通用性强、大模型适用和安全可靠的优势。

【功能模块】

- 数据集管理：将训练模型所需要的各种数据，导入到数据集管理中，以便于更清晰、方便地管理训练数据，加快训练速度。支持数据集共享，在线标注等。
- 模型开发管理：使用多种方式设计模型和训练，启动训练任务并为训练任务分配算力资源。
- 训练任务管理：查看和管理启动的所有训练任务。从已完成的训练中，挑选满意的训练结果发布为模型。
- 模型管理和评估：导入和管理所有模型，对模型进行版本管理、导入导出、评估、分享。
- 模型压缩：在保证模型效果的前提下压缩模型大小，进而提升模型在推理调用时的性能。
- 线上服务管理：将模型部署为在线服务，供应用方调用。

【功能特性】

- 简化训练和部署的复杂流程
- 开箱即用，降低调优成本

- 平台化全流程管理

1.2产品优势

- 简单易用

开通后无需额外的配置或调试，3 步操作实现零代码多机多卡微调，减少安装组件、下载模型和数据的重复操作。

- 功能全面

集成多种加速及并行技术，满足模型训练推理业务需求。支持断点续训、优雅容错等管理功能。

- 性能优异

万卡算力集群纳管，训推综合性能提升 30%。相对于裸机运行，稳定运行时长提升 50%。

- 生态开放

引入 20+生态大模型、闭源模型和电信大模型。专业团队开发模型及算子库，助力客户完成昇腾迁移适配。

- 全流程开发工具

提供训练数据管理、模型开发（代码式开发工具、快速微调、预置大模型、预置开发环境）、模型训练、模型管理、服务部署、服务管理到模型服务调用的全链路功能。集成分布式训练调度技术、多种训练加速方法和高性能存储，支撑大模型训练，并极大降低训练和应用模型成本、缩短训练时长。

- 兼顾各类用户需求

面向需要开发复杂模型的用户，提供完整的代码式开发工具、预置大模型、预置开发环境等，满足用户的各种复杂模型开发需求。面向希望能快速、便

捷建模的用户，则充分利用大模型微调训练的特点，提供快速微调工具，只需选择数据、配置参数即可完成大模型微调，降低大模型训练的使用门槛。

- 部署快捷，适配广泛

集成分布式算力调度、模型并行推理和多种运算加速能力，提升模型推理性能，实现推理服务的快捷部署。同时，适配多种模型结构，灵活支持用户各类复杂推理应用需求。

- 集成多种 AI 框架

集成多种 AI 框架，包括国产 AI 框架，支持各种主流大模型。

- 安全可信

符合数据监管要求，不设置数据埋点，不收集存储用户的入参和出参数据，从根本上保证了用户的数据隐私安全。

- 卓越的客户服务

31 省本地化的销售网络体系，提供家门口的精细化客户服务。7*24 小时的免费运维服务，全力保障客户业务稳定运行。

1.3功能特性

- 简化训练和部署的复杂流程

在传统的 AI 模型研发流程中，科研人员需要经历一系列繁琐的环节，包括数据准备、模型构建、模型训练、模型评估、模型优化以及模型部署等。这些环节不仅涉及数据工程、模型框架、算法开发、模型加速等多个技术领域，还要求科研人员熟练使用数据治理工具、数据标注工具、数据管理工具、数据读取工具等一系列专业工具组件。同时，他们还需处理这些工具与硬件环境、操作系统环境的适配问题，以及管理众多的依赖环境包。这一复杂过程不仅耗时耗力，而且大大提高了模型研发的使用成本和复杂程度。

一站式智算服务平台通过整合全链路的工具组件，实现了训练与部署流程的极大简化，为科研人员提供了一站式解决方案。用户无需再为繁杂的工具和环境配置而烦恼，只需专注于模型的核心研发工作。智算开发平台不仅降低了大模型开发的使用门槛，更让 AI 技术的普及和应用变得更加便捷和高效。

- 开箱即用，降低调优成本

大模型场景下训练数据处理和使用的过程尤为复杂。硬件层面，需确保编译环境、框架工具、依赖资源包等与硬件完美适配。软件层面，需保障操作系统、深度学习框架、编译器等软件工具的顺畅运行。针对大模型的训练和调优更是加剧了整个过程的复杂程度，同时伴随着大量的时间和算力资源的消耗。传统训练调优工具往往无法满足要求。

一站式智算服务平台为用户带来了便利，通过平台，用户无需进行任何额外的配置或调试，开箱即用。平台预置了丰富的预训练模型和镜像环境，针对不同场景提供了多样化预置数据集，确保用户能够迅速投入工作。同时，平台集成了大模型微调训练工具，适用于专属大模型的快速训练。此外，平台还支持分布式训练和 Deepspeed 加速框架，提供断点续训功能，支持小样本微调，使用户能够轻松定制专属模型，极大地降低了调优成本，提高了研发效率。

- 平台化全流程管理

AI 训练的高效执行，依赖于大数据团队、数据标注团队、算法开发团队、性能优化团队以及算法工程化团队等多个专业角色的紧密协作。

一站式智算服务平台，一个集成化的平台化工具，将以上所有角色都汇聚于一个统一的平台之上，提供从数据处理、模型开发、模型训练到最终模型部署应用的全栈服务。

管理者能够在平台上实现统一管理和查看，确保各环节的无缝衔接，让各角色参与者能借助平台完美协同工作，实现数据互通、环境互通，确保数据和模型安全，全程不出平台实现训练开发资产的一站式沉淀与管理，能显著提升企业整体工作效率，实现 AI 生产的流水线化运作。

1.4应用场景

- 模型训练

向下纳管智算硬件资源，提供技术运维及训练加速。向上通过模型开发平台提供大模型训练全链路功能，简化操作，提升效率。封装训练所需的底层技术，缩小训练者所需掌握的技术范围，降低大模型开发技术门槛。

主要用户包括各基础大模型厂商，各种拥有行业和场景专业知识与数据的行业客户，如科研院所、大专院校和教育机构、政府、金融机构、工业企业、科技单位、医院等。

- 模型推理

向下纳管智算硬件资源，提供技术运维服务及推理加速。向上通过模型服务平台提供部署好的模型服务，并集成丰富配套工具，提供模型推理一站式部署服务。

主要用户包括各种软件开发商，特别是行业软件开发商，以及科研院所、大专院校和教育机构、政府、金融机构、工业企业、科技单位、医院等行业客户。

- 算力运营

智算平台可部署在客户的智算资源上，对算力资源进行统一管理、统一调度，赋能客户算力运营能力，帮助客户通过算力运营和销售取得收益。

主要客户包括各种算力运营商，如各行业大型企业集团、政府旗下的基建投资公司等。

1.5术语解释

- 预置模型

是指平台提供的原始模型，您可以通过选择预置模型进行训练从而得到行业或细分场景模型，不同的基础模型的参数和能力不同，我们将持续推出不同能力方向的模型。

- 模型微调

是指利用预先训练好的神经网络模型，并针对特定任务在相对较少量的监督数据上进行重新训练的技术。这种方法能够充分利用预训练模型在大型数据集上学到的通用特征和知识，从而加速在新任务上的训练过程，并通常能够取得较好的性能表现。

- 迭代轮次

是指模型训练过程中模型学习数据集的次数，可理解为学习几遍数据，可依据需求进行调整。

- 批处理大小

是指在模型训练过程中，每次处理的数据样本的数量，可理解为模型每看多少数据即更新一次模型参数，在选择批处理大小时需要综合考虑各种因素。

- 学习率

是指更新模型参数的系数，它决定了在每次迭代中，模型参数应该沿着梯度下降的方向更新多少，需要根据具体情况来仔细选择和调整学习率。

- 训练数据集

是机器学习或深度学习模型训练过程中的重要组成部分。训练数据集是一组已知输入和对应输出的数据，用于训练模型以学习从输入到输出的映射关系。构建合适训练集，通过模型调优可增强模型能力，提升预测效果。

- 测试数据集

在机器学习和深度学习中扮演着至关重要的角色，它用于评估模型在未见过的数据上的性能。与训练数据集不同，测试数据集在模型训练过程中是不可见的，也就是说，模型在接触到测试数据之前已经完成了所有的训练和调整。

1.6使用限制

- 数据隔离：数据是按照存储桶和文件目录做逻辑隔离。
- 集群隔离：集群是通过 VPC 网络隔离，完全互相独立。
- 多资源池：多资源池之间，数据、模型、镜像、集群是无法共享的，独立使用独立训练。
- 多集群：同资源池内、多集群之间，数据集、模型、镜像是共享的，任务实例是互相隔离的。

2 计费说明

2.1包周期计费方式

包周期付费指按订单的购买周期计费，是一种预付费模式，即先付费再使用。您可以通过包周期计费提前预留资源，同时享受比按量计费更大的价格优

惠。包周期计费适用于多种场景，尤其是需要稳定资源并长期使用的情况。

产品名称	规格	CPU	内存（GB）	显卡类型	显卡数	说明	标准价格（含税）	单位
训练服务DO-LC-标准型包周期	16C|128G|1*910B-64G（液冷）或以上	16	128	910B	1	共享集群	24772.87	元/服务*月
训练服务DO-LC-扩展型包周期	184C|1520G|8*910B-64G（液冷）或以上	184	1520	910B	8	共享集群	198182.95	元/服务*月
训练服务DO-LC-独立型包周期	184C|1520G|8*910B-64G（液冷）或以上	184	1520	910B	8	独立集群	196000.00	元/服务*月

训推一体服务 FO-L2	224C|2048G|8*H800-640G 或以上	224	2048	H800	8	独立集群	241212.00	元/服务*月

2.2按需计费模式

按需付费是一种灵活的计费模式，您可以通过按量计费灵活开通和释放资源，无需提前购买。按需付费的优势在于灵活性和节约成本，适用于需要灵活调整资源、业务不稳定或资金有限的场景。在选择计费模式时，应结合业务需求和实际情况来做出合适的选择。

产品名称	规格	CPU	内存（GB）	显卡类型	显卡数	说明	标准价格（含税）	单位
训练服务 DO-LC-标准型	16C|128G|1*910B-64G（液冷）或以上	16	128	910B	1	共享集群	41.29	元/服务*小时

训练服务 DO-LC-扩展型	184C|1520G|8*910B-64G（液冷）或以上	184	1520	910B	8	共享集群	330.30	元/服务*小时

2.3产品退订

服务开通后 7 天内如未使用则支持退订，退订后即可关闭。

退订地址：我的—费用中心—订单管理—退订管理

另外，您可通过天翼云官网工单或者客服电话【400-810-9889】沟通申请退款，款项会原路退回。

3 快速入门

3.1准备工作

- 注册天翼云账号

在开通和使用一站式智算服务平台之前，您需要先注册天翼云门户的账号。本节将介绍如何进行账号注册，如果您拥有天翼云的账号，可登录后使用一

站式智算服务平台。

1. 打开天翼云门户网站，点击【注册】。

2. 在注册页面，请填写【邮箱地址】、【登录密码】、【手机号码】，并点击【同意协议并提交】 按钮，如 1 分钟内手机未收到验证码，请再次点击【免费获取短信验证码】按钮。

3. 注册成功后，可到邮箱激活您的账号，即可体验天翼云。

4. 如需实名认证，请参考会员服务-实名认证。

- 为账户充值

* 使用一站式智算服务平台之前，请保证您的账户有充足的余额，账户余额需要大于 100 元。

* 关于如何为账户充值，请参考费用中心-账户充值。

* 一站式智算服务平台支持按卡时后计费和包周期预付费。

4 用户指南

4.1 模型广场

4.1.1模型查看

进入模型广场模块，点击【模型卡片】，查看平台预置模型的模型介绍（含使用场景、版本列表等）、API 文档、任务记录等内容。

4.1.2一键精调

支持对平台预置的模型进行一键精调，可点击模型卡片上的【精调】按钮直接发起精调。目前支持对 Llama2-13B-Chat、Qwen2-7B-Instruct 等模型发起精调。

4.1.3一键评估

支持对平台预置的模型进行一键评估，可点击模型卡片上的【评估】按钮直接发起评估。目前支持对 Llama2-13B-Chat、Qwen2-7B-Instruct 等模型发起评估。

4.1.4一键部署

支持对平台预置的模型进行一键部署，可点击模型卡片上的【部署】按钮直接发起部署。选择默认资源池，目前支持对 Llama2-13B-Chat、Qwen2-7B-Instruct 等模型发起部署。

4.1.5 API调用

支持通过 API 调用模型广场预置模型的推理服务，详情操作请参考模型服务相关内容。

4.1.6 任务记录

点击【任务记录】，可查看该模型的任务历史，包括模型精调、模型评估、模型部署等任务。

4.2体验中心

本模块提供在线测试的功能，您可以即刻体验模型效果。

4.2.1 体验中心工作台

进入体验中心工作台，需要先选择服务类型，当前可选择：文本对话、文本生图、图像理解。左下方支持通过测试台选择服务/应用进行参数配置。

4.2.1.1文本对话类模型体验

1.参数配置：可以在左侧测试工作台选择服务进行参数配置。

温度：Temperature 控制生成文本的多样性。较高的温度值会使生成的文本更加随机和多样化，而较低的温度值会使生成的文本更加确定和一致。

多样性：TopP 影响输出文本的多样性，取值越大，生成文本的多样性越强。

重复惩罚：Frequency_penalty 影响模型生成重复词汇的倾向。通过增加重复词汇的惩罚权重，降低模型逐字重复的可能性。

系统人设：设定模型的行为和背景，告知模型需要扮演的角色。例如： “假如你是一个 AI 助手”。

2.输入框中可直接输入问题，系统将根据输入的问题及配置的参数进行实时回答。

4.2.1.2文本生图类模型体验

1.参数配置：选择服务后，支持配置：绘画描述（Prompt）、图片风格、图片比例。

2.配置参数后，点击“生成图片”，即可在页面右侧查看生成的结果。

4.2.1.3图像理解类模型体验

1.参数配置：选择服务后，支持配置：Prompt、图片。

2.配置描述及图片后，发送对话，即可在页面右侧查看生成的结果。

4.2.2查看历史记录

点击右上角【查看历史记录】，系统会展开历史的对话记录，可以查看统计大模型的回答质量，保存上限为 200 条。

4.3模型定制

4.3.1模型精调

4.3.1.1 创建调优任务

- 进入模型定制模块，选择模型精调，进入调优任务列表，点击【新建调优任务】，进入创建页面。
- 选择已导入的数据集，选择基础大模型，设置调参方式、迭代轮次、批处理大小、学习率等指标，配置资源选择算力规格。不同的算力规格对应不同的价格，单节点下卡数越多训练越快。

4.3.1.2 监控调优任务

- 返回模型调优训练任务列表，列表中可以看到每个任务的运行进度、预估时长。
- 点击任务名称，可进入调优任务详情页，右上角可对任务进行停止和删除操作。
- 详情页可依次查看任务基础信息、日志、监控、Tensorboard 看板。监控中运行进度可以看到每一次迭代是否完成，资源监控看板可以查看 CPU 使用率、内存使用率、NPU 使用率等。

4.3.2开发机

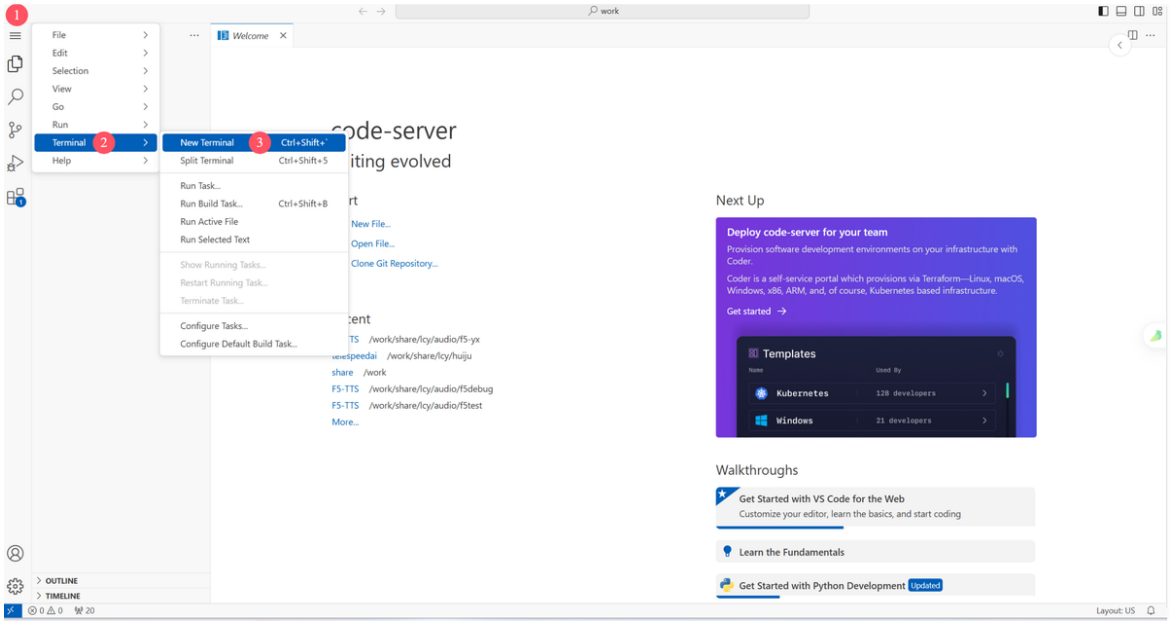
4.3.2.1环境登录和使用

- 使用注册时的手机号登录平台 https://huiju.ctyun.cn/modelSquare/?regionId=200000001852

- 先创建开发机IDE任务，按照如下步骤展开操作：

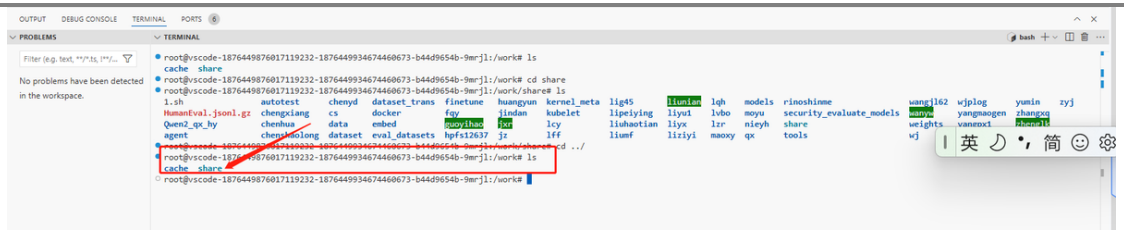
- 提交任务后，任务状态将依次显示启动中->环境准备中->运行中（如果长时间（>1min）界面状态未更新，可以使用F5手动刷新界面），当状态显示为运行中后，点击操作栏【打开】按钮。

- 打开后，显示vscode界面如下，按照下图步骤打开terminal运行终端：



4.3.2.2 启动训练任务

在/work/share/ 目录下创建自己的工作空间



- **vscode启动单机训练任务**

terminal 终端目前只支持单机训练任务，训练脚本跟tensorflow和pytorch 裸金属训练模式一样。

下方是在本机执行的单机多卡torchrun分布式参数示例：

```
DISTRIBUTED_ARGS="
    --nproc_per_node 8 \
    --nnodes 1 \
    --node_rank 0 \
    --master_addr localhost \
    --master_port 65500
"
torchrun $DISTRIBUTED_ARGS telespeed/run/llama31/pretrain_gpt.py
```

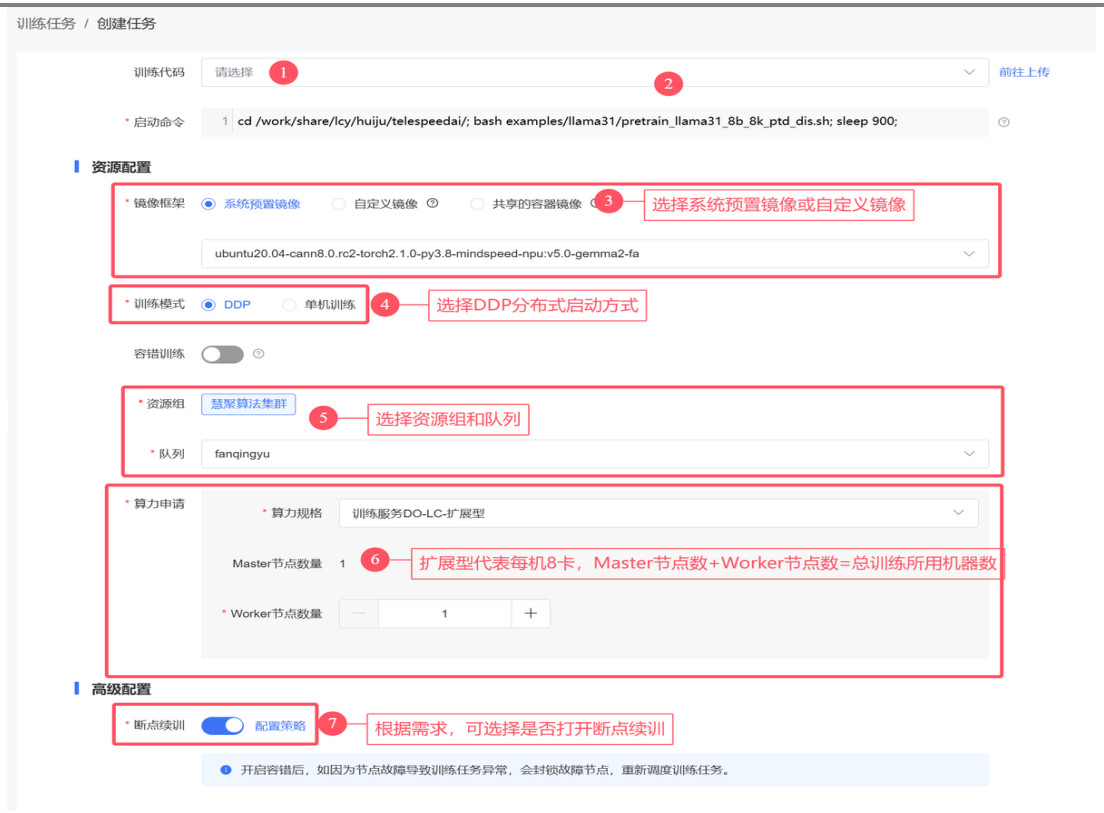
- **网页启动多机训练**

下方是在平台执行多机多卡训练任务的torchrun分布式参数示例：

```
GPU_NUM_PER_NODE=8
DISTRIBUTED_ARGS="
    --nproc_per_node $GPU_NUM_PER_NODE \
    --nnodes $PET_NNODES \
    --node_rank $PET_NODE_RANK \
    --master_addr $PET_MASTER_ADDR \
    --master_port $PET_MASTER_PORT
"
torchrun $DISTRIBUTED_ARGS telespeed/run/llama31/pretrain_gpt.py
```

按照下图的步骤启动训练任务


基本信息
* 任务名称
test_distributed
描述
请输入描述信息
1
拟定训练任务名称
0/128
数据集配置
训练数据集
+ 增加数据集
您代码中读取数据的相对路径需要改为本地挂载路径。
测试数据集
+ 增加数据集
您代码中读取数据的相对路径需要改为本地挂载路径。
预置模型配置
模型来源
我的模型
预置模型
我的模型文件
+ 增加我的模型文件
将模型管理中的模型文件挂载到容器本地路径，您的代码可以直接读取此相对路径。
环境配置
训练代码
请选择
前往上传
切换至代码工程目录下，执行训练脚本
* 启动命令
1 cd /work/share/lcy/huiju/telespeedai/;
2 bash examples/llama31/pretrain_llama31_8b_8k_ptd_dis.sh;
2



- **启动多个训练任务**

训练日志可以通过如下日志按钮进行查看，也可以在vscode 开发机本地目录查看。

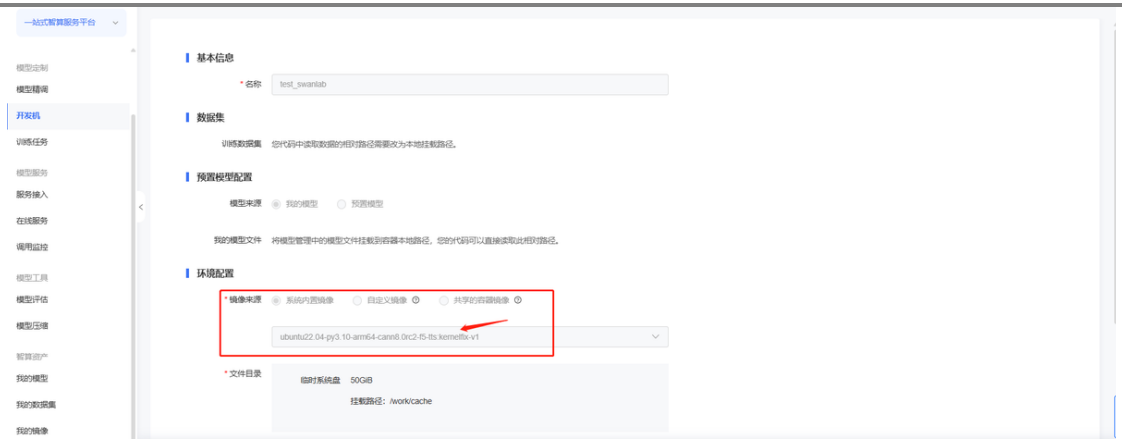
- **自定义镜像**

系统预置镜像往往不能满足开发需求，需要在预置镜像中进行环境安装然后重新打包镜像进行使用，打包后的镜像放在自定义镜像中。自定义镜像只能在自己账号内使用且不能组员分享使用，如果需要分享使用需要联系天翼云开发团队将自定义镜像迁移到系统预置镜像里面。

- 点击【制作镜像】按钮

- 按照规则进行镜像命名，镜像命名建议参考历史镜像的名称微改，历史镜像名称的获取方式如下：
  - 点击查看配置

  - 箭头处即为历史镜像名称



4.3.3训练任务

4.3.3.1 配置训练任务

进入模型定制模块，选择【开发机】，进入 JupyterLab 或 VSCode 列表，点击操作列【开始训练】，进入分布式训练配置页面，输入启动命令>选择镜像框架>配置算力资源，即可启动训练。

说明：
算法配置：
打开代码所在的文件路径：cd /work/代码所在路径
启动命令
如果是 sh 文件，启动命令写：sh 文件名.sh
如果是 py 文件，启动命令写：python 文件名.py
资源配置：
如果镜像中有 deepspeed，需要开启 deepspeed
单节点【184C|1520G|8*910B-64G（液冷）或以上】指：184 核 CPU，1520G 内存，单节点 8 张 910B 的 NPU 卡，每张卡的显存大小为 64GB。。
节点指物理机数量，Master+Worker 的数量为多机多卡总节点数，等于训练脚本中指定的 WORKER_CNT 的数量，算力规格中 GPU 的数量等于训练脚本中指定的 GPUS_PER_NODE 的数量。

4.3.3.2 命令行启动训练任务

执行安装命令：pip install /mnt/public/job_submit/task_submission-2.0-py3-none-any.whl

```
说明：
使用样例参考：/mnt/public/job_submit 目录下的样例文件 submit.sh
关键命令：
提交 pytorch 任务的帮助说明：submit pytorch --help
提交其他任务的帮助说明：submit --help
```

4.3.3.3 监控训练任务

- 进入【训练任务】，可以看到训练任务的状态、日志，可对训练任务进行启动、停止等。
- 操作列点击【日志】进入详情页，日志 tab 可以查看到运行的日志，支持搜索。监控 Tab 可查看硬件使用率。

4.4模型服务

4.4.1服务接入

进入模型服务模块，点击【服务接入】，服务接入模块可以将预置服务及我的服务打包为服务组，生成 AppKey 供用户调用；

●点击【创建服务组】，填写服务名称和服务描述。选择需要关联的服务（支持选择预置服务、我的服务），提交完成创建。

●完成创建后，系统会自动创建一个调用服务的密钥，即生成该服务的密钥，即“AppKey”。

●点击服务组卡片【查看详情】，可进入服务组详情页，查看该服务组关联的服务，点击【了解更多】可查看模型详情。

●点击服务组卡片对应按钮，可以支持对“AppKey”进行复制和重置，可以支持对服务进行停用/启用、编辑、删除、服务组监控。

4.4.2在线服务

平台支持将用户精调后的模型发布为在线服务，同时也支持直接调用预置模型的在线服务。

●点击【预置服务】，可查看目前平台预置的所有服务，目前可免费试用，每个模型免费试用上限为 1 万 tokens。点击 API 文档，可以查看模型的调用方式。

●点击【我的服务】，可以选择已有模型创建模型服务并进行服务管理。

1、部署我的模型

●镜像环境选择支持：系统内置镜像、从 JupyterLab/VSCode 中制作的自定义镜像、容器镜像服务共享过来的镜像。

●模型可选择智算资产-我的模型模块导入成功的模型。

●代码包可选择在智算资产-我的代码包模块中已上传的一个代码包。

●三方库配置支持选择三方库列表、requirements.txt 文件目录，指定三方库列表，格式与 requirements.txt 一致，输入内容以换行符分隔。

●输入环境变量。

●输入镜像的启动运行命令，如 python/mount/code/{codeid}/run.py（须提供 OAI 兼容的推理服务接口服务）。

●选择资源部署信息，包括队列、资源规格和实例数量，系统会回显对应计费信息。

●完成部署，并开始计费。

2、管理我的服务

●在列表可查看模型是否部署成功，在操作列可进行模型查看、更新、停止、重启、修改、上下线、删除等操作。停止服务后计费也会停止，再次启动服务即可开通计费。

●操作列点击【查看】可进入该服务的详情页，查看部署的模型列表、服务监控、配置历史、运行记录、事件日志、服务日志。

●状态为运行中的模型服务可正常调用。需要使用 location+modelId+appKey 请求调用。具体调用方式如下：

1. 点击【查看】进入该服务的详情页，可查看该服务的 API 文档，通过详情页中的“modelId”和“接口地址”条目获取 modelId 和 location。

2. 创建或编辑服务组，选择对应服务并提交，通过服务组卡片上的“App Key”条目获取 AppKey。

3. 根据平台规范构造请求，调用对应服务，目前支持部署 Chat 类型的模型，请求样例如下：

```
curl --location '{$location}' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer {$appKey}' \
--data '{
    "model": "{$modelId}",
    "messages": [
        {
            "role": "user",
            "content": [
                {
                    "type": "text",
                    "text": "xxx"
                }
            ]
        }
    ]
}'
```

4.4.3调用监控

进入模型服务模块，点击【调用监控】，支持查看预置服务的调用数据。

- 在页面顶部选择统计时间，筛选后，即可查看该时间段内全部服务的调用监控概览，包含调用总量、调用失败量、调用总 tokens 等指标。
- 在模型列表点击【查看监控】，进入该模型的调用数据详情页，可以查看具体的模型在特定服务组、特定服务中的调用监控情况。
- 点击【调用失败明细】，可以查看调用失败的次数、占比、错误信息等数据。
- 点击【导出】，可以直接导出相关数据到本地。

4.5模型工具

4.5.1模型评估

模型评估旨在对平台精调生成的大模型输出效果进行评估，当前支持对“模型调优”运行完成的模型进行评估。

4.5.1.1评估数据准备

准备用于评估模型能力的数据集，并在数据集管理中导入和发布。

4.5.1.2新建评估任务

在模型评估菜单页面中，点击【新建评估任务】，选择一个用于评估的 Benchmark 数据集，选择对应的评估标准，以及评估用到的资源，即可完成评估任务新建。

- 准确率：正确预测(标注与预测完全匹配)的样本数与总样本数的比例。
- ROUGE-1：将模型生成的结果和标准结果按 unigram 拆分后，计算出的召回率。
- ROUGE-2：将模型生成的结果和标准结果按 bigram 拆分后，计算出的召回率。
- ROUGE-L：衡量了模型生成的结果和标准结果的最长公共子序列，并计算出召回率。
- BLEU-4：用于评估模型生成的句子和实际句子的差异的指标，值为 unigram，bigram，trigram，4-grams 的加权平均。

4.5.1.3查看评估任务

在评估详情页，可以查看评估任务的详细内容，包括基座模型、模型评估结果、评估日志等信息。

4.5.2模型压缩

模型压缩旨在帮助客户在尽量不减少模型效果的前提下压缩模型大小，进而提升模型在推理调用时的性能。

4.5.2.1 创建模型压缩任务

在菜单中选择模型压缩，进入模型压缩主任务界面，点击“创建压缩任务”按钮，进入新建压缩任务页面。由用户填写压缩任务所需的任务信息、模型信息、压

缩配置、资源配置。

基本信息

填写压缩任务名称、压缩任务描述。

压缩配置

- 选择源模型: 此处支持选择用户希望压缩的模型，支持从『模型管理』中选择（不支持选择预置模型）。
- 模型创建方式：选择压缩后模型的保存方式，支持保存为已有模型新版本（默认为最新版本）或保存为新模型（默认 V1 版本）。
- 选择已有模型：同一模型各版本的基础模型需保持一致，已自动过滤不符合要求的模型。
- 压缩策略-量化压缩：

WxAxCx 中 W、A、C 分别代表模型权重（weight）、激活（activation）和键值缓存（kv cache），数字 x 代表模型压缩后相应部分的比特数。模型压缩过程后，高比特浮点数会映射到低比特量化空间，从而达到降低显存占用、提升推理性能等目的。 模型的推理性能收益均需要通过实际测试获得，表中策略类型仅做参考。

4.5.2.2 获取压缩结果

压缩任务运行完成后，压缩后的模型会自动保存到用户指定的模型管理中的位置。可以通过开发机挂载模型、或者下载模型来查看模型文件本身。压缩后的模型可以直接通过我的服务进行服务部署，部署为推理效果更优的大模型服务。

4.6智算资产

4.6.1我的模型

全面管理用户从开发、训练到评估完成的模型生命周期，该模块不仅提供模型文件的安全存储功能，还具备精细化的版本管理，确保每一阶段的模型变更都有迹可循。

4.6.1.1新建模型

在我的模型菜单页面中，点击【新建模型】，输入模型名称、以及导入模型。支持 4 种导入方式，分别为当前平台导入、本地上传、口令导入、下载链接导入。

- 当前平台导入：支持从平台上运行完成的模型调优和训练任务中导入、也可以从 JupyterLab 和 VSCode 的目录中导入。
- 本地上传：支持从本地电脑导入不超过 2G 的模型文件。
- 口令导入：支持输入一站式智算服务平台其他账户分享的口令完成导入。
- 下载链接导入：支持输入互联网下载链接地址完成模型导入。

4.6.1.2模型列表

导入的模型可以在我的模型的列表中查看，每个模型可以导入多个版本。操作列点击【查看详情】可以查看模型的所有版本。

模型的每个版本都会显示导入状态，比较大的模型导入时间较长。

4.6.1.3模型分享与导出

1. 模型分享

模型列表和模型版本列表中，点击【分享】可生成分享口令，支持模型分享，可将模型在多个账号之间进行共享下载。

账户 1 要把模型文件分享给账户 2，需要账户 1 在模型列表或版本列表中点击【分享】获得一个分享口令，将分享口令线下给到账户 2。

账户 2 在新建模型中选择【口令导入】，输入账户 1 给到的分享口令即可完成模型导入。

2. 模型导出

进入模型详情页，在模型版本列表中支持模型导出，可以选择导出到本地，也可以选择导出到天翼云媒体存储中。

4.6.2我的数据集

4.6.2.1数据导入

进入我的数据集模块，点击【创建数据集】，录入数据集名称、数据类型、标注类型等。

1.本地数据导入：数据集操作列点击【导入数据】，导入方式选择“本地上传”或“上传压缩包”>导入方式“本地压缩包导入”。

2.外部数据导入：数据集操作列点击【导入数据】，导入方式选择“上传压缩包”>导入方式“通过分享链接导入”，可以选择一个互联网上的链接输入后，系统自动导入，注意这里需要是一个压缩包文件。

4.6.2.2数据标注

对导入成功的数据，点击操作列【标注】进入标注页面。

1.在标注详情页对数据进行微调和打标处理。

2.页面左侧可对导入数据内容进行修改和撰写。

- 指令微调数据标注：instruction、input、output 是指令微调的 3 个字段，instruction 代表指令要求，input 代表指令输入，output 代表模型根据指令和输入执行的结果。撰写完成点击【下一篇】按钮进行下一条数据的处理。
- Q&A 对数据标注：Text、Query、Answer、Match、File、Similar-Question 是 Q&A 对数据的 6 个字段，Text 代表文件名称，Query 代表查询问题，Answer 代表问题对应的回答，Match 代表查询条件与数据源中数据项的匹配结果，File 代表处理数据位置，Similar-Question 代表相似的问题。撰写完成点击【下一篇】按钮进行下一条数据的处理。
- 强化学习回复排序数据标注：无标注信息代表无任何标注和排序动作信息数据，有标注信息代表含相关度排序和安全度排序标注动作信息数据，无相关度排序是候选回复无相关排序动作数据，无安全度排序代表无安全度排序动作数据。撰写完成点击【下一篇】按钮进行下一条数据

  的处理。点击【恢复默认排序】可清除在线排序操作。

  3.页面右侧可对导入数据进行打标审核。

4.6.2.3数据管理

- 数据集加速：如果您希望训练过程中训练速度更快的话，可以选定数据集，点击【操作】，选择【推送到高速缓存】，该操作可将数据集从对象存储转存到快速存储中进行加速。
- 数据集发布：针对文本类数据集，标注完成后，可以选定数据集，点击【发布】，完成发布后的数据集才能供后续的训练使用。

- 数据集共享：选定数据集，点击【操作】，选择【共享数据】生成共享口令，对方点击数据集管理页面【添加共享数据集】输入口令即可将您共享的数据集添加至数据集列表。

4.6.3我的镜像

4.6.3.1制作镜像

- 启动在线制作环境：进入模型定制模块，选择开发机，点击【JupyterLab】>【创建 JupyterLab】或【VSCode】>【创建 VSCode】，选择一个系统内置镜像，选择运行环境，提交后操作列点击启动。
- 镜像制作：等待启动成功，当创建的 JupyterLab 或 VSCode 的状态显示【运行中】后即可点击操作列【打开】，在开发环境中安装自己需要的软件和环境，退出，选中创建的 JupyterLab 或 VSCode，操作列点【更多】>【制作镜像】，即可将容器中的操作环境打包成新的镜像，并出现在自定义镜像列表中。

4.6.3.2镜像共享

- 登陆天翼云容器镜像服务，访问地址（https://crs.ctyun.cn/dy/crs/#/dashboard），在【同资源池】下，按需创建【个人版】或【企业版】，进入实例详情创建属于您自己的命名空间和镜像仓库。
- 有了镜像仓库后，根据实例详情访问凭证中的指引通过公网地址将您的镜像上传至仓库中。
- 进入镜像共享，创建镜像共享，将您希望使用的镜像共享至一站式智算服务平台，共享目

标用户填入【huijuprod】，共享后您就可以在一站式智算服务平台的自定义镜像中看到此镜像。

注意：

1、如果您想在模型开发 JupyterLab 和 VSCode 中使用自定义镜像，需要将对应的软件安装包打包进您的自定义镜像中。

方法 1：在 docker file 中具体执行命令：

```
# vscode
curl -fsSL https://code-server.dev/install.sh | sh
code-server --install-extension ms-python.python
# jupyterlab
pip install jupyterlab
```

方法 2：将打包好的镜像在本地起起来，然后运行如下命令安装软件，安装完成后，执行 docker commit {容器名称}，打包成新镜像后，即可上传。

```
# vscode
curl -fsSL https://code-server.dev/install.sh | sh
code-server --install-extension ms-python.python
# jupyterlab
pip install jupyterlab
```

2、如果在自定义镜像列表看不到分享过来的镜像，请检查：

- 容器镜像服务所选区域与平台是不是同资源。
- 截止时间是不是>=当前时间，超出截止时间后共享失效。
- 共享的镜像状态是不是启用。

4.6.4我的代码包

支持直接上传本地文件、本地压缩包。单次上传文件最多支持 5 个。

对于文件数量较多，建议使用压缩包上传。

上传完成后操作列点【在线编码】即可进入 JupyterLab 或 VSCode 进行编码。

说明：
存储目录：
/work 目录可以被用作统一的文件管理，同时开发机中不同的实例或容器任务可以共享这个目录。
/work 目录下中有 3 个子目录。3 个目录的区别如下：
/work/home：您独享的、永久的、高性能存储空间，关闭开发机和训练任务后存储内容始终保留。可用于存放代码和部分数据集等重要的文件，建议个人仅使用该目录。
/work/cache：您独享的临时高性能存储空间，但关闭开发机后存储内容仅保留 3 天。可用于存放临时的代码和部分数据集。
tensorboard：保存在/work/home/task/${MODEL_PATH}/model 下，保存后在页面上可以通过 tensorboard 查看。前提是需要先开通 home 目录。
获取脚本所在目录：
获取脚本所在目录的绝对路径 ：SCRIPT=$(readlink -f "$0")
获取该脚本所在目录的路径：SCRIPTPATH=$(dirname "$SCRIPT")
输出脚本所在的目录 ：echo "当前脚本所在目录为： $SCRIPTPATH"

4.7运营后台

4.7.1用户运营

旨在让平台管理员能够轻松查看并管理本租户下所有用户的平台使用情况。

- 进入用户运营模块，用户运营详情页分为用户数据大盘以及用户列表两大板块。
- 定位到用户数据大盘，设置时间范围，即可查看所选时间段内的总用户数、每日用户数、总付费用户数、每日付费用户数。付费用户指在平台使用了耗费算力的功能，比如模型训练。
- 定位到用户列表，可查看本租户下所有用户的基本信息如账号、名称，登录信息，任务信息，消耗资源信息以及消费金额信息，右侧操作列支持为每

个用户设置单任务配额，即最大可用 GPU 卡数/CPU 核数。用户列表支持按用户名称筛选。

- 若您在平台已开专属集群，您在【用户列表】右侧可以看到【队列管理】，滑动到队列管理，可以看到本租户所在集群的队列列表，可查看每个队列的基本信息、运行任务信息、用户数、资源占用等信息，支持创建新队列，修改已有队列的可使用用户、队列 GPU/CPU 数量信息，删除已有队列等操作。

4.7.2资源运营

面向在一站式智算服务平台已开通专属集群的租户，旨在让平台管理员能够轻松查看并管理专属集群的资源使用情况。

- 进入资源运营模块，资源运营详情页分为资源&任务大盘、资源利用曲线图、任务列表三大板块。
- 定位到资源&任务大盘，选择集群，设置时间范围，即可查看选定集群所选时间段内 GPU/CPU 总量、正在使用量、空闲量以及正在使用量/空闲量占比。可以查看当前训练中任务数、排队中任务数以及排队中任务所需 GPU 卡数。
- 定位到资源利用曲线图，设置时间范围，即可查看所选时间段内，GPU/CPU/显存/内存利用率曲线图，支持按每天、每小时查看，支持将数据下载到本地。可以查看 GPU/CPU 卡时耗时曲

线图，启动训练任务数/实例数曲线图，排队中任务所需 GPU/CPU 峰值数曲线图。

- 定位到任务列表，设有排队任务管理、运行任务管理、运行历史三个标签页，排队任务可以查看等待时长，可以调整其优先级，优先级越高越优先被调度。运行任务可以查看任务的运行状态及时长，运行历史可以查看运行结束的任务。

4.7.3监控调度

面向在一站式智算服务平台已开通专属集群的租户，旨在让平台管理员能够轻松查看并调度集群资源。

- 进入监控调度模块，监控调度详情页分为节点统计大盘、节点状态监控、节点列表三大板块。
- 定位到节点统计大盘，选择集群，即可查看选定集群节点维度的资源情况，包含总节点数、空闲节点数、污点节点数、异常 GPU 卡数、单节点最大空闲 GPU 卡数、正在使用/空闲 GPU 卡数。
- 定位到节点状态监控，可以通过不同颜色区分每个节点每块 GPU 卡的占用/空闲状态，以及是否出现硬件错误。
- 定位到节点列表，可以查看所有节点的状态、标签、资源规格、GPU/CPU/内存利用率等信息。
- 将标签页从节点列表切换到 GPU 列表，可以查看所有 GPU 卡运行的实例、运行时长、GPU/显存利用率等信息。

4.7.4配置设置

旨在让平台管理员能够轻松查看并设置本租户下所有用户对资源使用的限额。

进入配置设置模块，可支持设置单用户最大同时使用的 GPU/CPU 数量以及并行文件存储初始分配额度。

5 推理服务 API

5.1 如何调用API

5.1.1 终端节点

- 主要作用：用户信息的发送和接收，信令信息的控制处理、安全保护等作用。
- 终端节点：https://wishub-x1.ctyun.cn

5.1.2接口构造

请求域名

1. 终端请求地址：https://wishub-x1.ctyun.cn

通信协议

- 接口通过 HTTPS 进行通信，保护用户数据的机密性和完整性，确保网络通信的安全性。

版本管理

- 为区分接口版本，在http请求路径中加入版本信息，目前版本为v1，因而请求的url前缀为：**https://wishub-x1.ctyun.cn/v1**

请求鉴权

- 请求header中需要填入Authorization鉴权信息，Authorization对应值应为Bearer + 平台上获取的服务组App Key
  - AppKey获取方法：登录平台门户，一站式智算服务平台->模型服务->服务接入->创建服务组->绑定服务->获取appKey
- 为避免被安全护栏拦截，建议在http请求header中填入User-Agent信息
  - 浏览器或客户端标识：Chrome/58.0.3029.110、Mozilla/5.0、AppleWebKit/537.36、Safari/537.36、Windows NT 10.0、

PostmanRuntime-ApipostRuntime/1.1.0等

请求示例

```
请求路径：https://wishub-x1.ctyun.cn/v1/xxx/yyy，其中/xxx/yyy 为具体的功能路径，如/chat/completions
请求方式：POST
请求header必填项：
Authorization: Bearer AppKey
Content-Type: application/json
其他header：
User-Agent: PostmanRuntime-ApipostRuntime/1.1.0
```

5.2 接口类型列表

功能分类	支持模型类别	请求路径后缀	请求完整路径	功能描述
chat	文本生成、图像理解	/chat/completions	https://wishub-...	针对描述会话的消息列表，模型将返回响应。
image	文本生图	/images/generations	https://wishub-...	基于提示创建图像。

5.3 API列表

模型	模型简介	模型ID
Baichuan2-Turbo	Baichuan-Turbo系列模型是百川智能推出的大语言模型，采用搜索增强技术实现大模型与领域知识、全网知识的全面链接。	43ac83747cb34730a00b7cfe590c89ac
Llama2-13B-Chat	Llama2是预先训练和微调的生成文本模型的集合，其规模从70亿到700亿个参数不等。这是13B微调模型的存储库，针对对话用例进行了优化。	96dc8f33609d4ce6af3ff55ea377831a
Qwen-7B-Chat	通义千问-7B（Qwen-7B）是阿里云研发的通义千问大模型系列的70亿参数规模的模型。Qwen-7B是基于Transformer的大语言模型, 在超大规模的预训练数据上进行训练得到。预训练数据类型多样，覆盖广泛，包括大量网络文本、专业书籍、代码等。同时，在Qwen-7B的基础上，使用对齐机制打造了基于大语言模型的AI助手Qwen-7B-Chat。	fc23987da1344a8f8bdf1274e832f193
Llama2-7B-Chat	Llama2-7B-Chat是Meta AI开发的大型语言模型Llama2家族中最小的聊天模型。该模型有70亿个参数，并在来自公开来源的2万亿token数据上进行了预训练。它已经在超过一百万个人工注释的指令数据集上进行了微调	e30f90ca899a4b1a9c25c0949edd64fc

	。	
Llama2-70B-Chat	Llama 2 是预训练和微调的生成文本模型的集合，规模从 70 亿到 700 亿个参数不等。这是 70B 微调模型的存储库，针对对话用例进行了优化。	bafbc7785d50466c89819da43964332b
Qwen1.5-7B-Chat	通义千问1.5（Qwen1.5）是阿里云研发的通义千问系列开源模型，是一种基于 Transformer 的纯解码器语言模型，已在大量数据上进行了预训练。该系列包括Base和Chat等多版本、多规模，满足不同的计算需求，这是Qwen1.5-7B-Chat版本。	bfc0bdbf8b394c139a734235b1e6f887
Qwen2-72B-Instruct	Qwen2 是 Qwen 大型语言模型的新系列。Qwen2发布了5个尺寸的预训练和指令微调模型，包括Qwen2-0.5B、Qwen2-1.5B、Qwen2-7B、Qwen2-57B-A14B以及Qwen2-72B。这是指令调整的 72B Qwen2 模型，使用了大量数据对模型进行了预训练，并使用监督微调和直接偏好优化对模型进行了后训练。	2f05789705a64606a552fc2b30326bba
ChatGLM3-6B	ChatGLM3-6B 是 ChatGLM 系列最新一代的开源模型，在保留了前两代模型对话流畅、部署门槛低等众多优秀特性的基础上，ChatGLM3-6B 引入了更强大的基础模型、更完整的功能支持、更全面的开源序列几大特性	7450fa1957784203935 42c7fa13c6640

	。	
TeleChat-12B	星辰语义大模型TeleChat是由中电信人工智能科技有限公司研发训练的大语言模型，TeleChat-12B模型基座采用3万亿 Tokens中英文高质量语料进行训练。TeleChat-12B-bot在模型结构、训练数据、训练方法等方面进行了改进，在通用问答和知识类、代码类、数学类榜单上相比TeleChat-7B-bot均有大幅提升。	fdc31b36028043c48b15131885b148ce
Qwen1.5-14B-Chat	通义千问1.5（Qwen1.5）是阿里云研发的通义千问系列开源模型，是一种基于 Transformer 的纯解码器语言模型，已在大量数据上进行了预训练。该系列包括Base和Chat等多版本、多规模，满足不同的计算需求，这是Qwen1.5-14B-Chat版本。	acfe01f00b0c4ff49c29c6c77b771b60
Llama3-8B-Instruct	Meta 开发并发布了 Meta Llama 3 系列大型语言模型 （LLM），包含 8B 和 70B 两种参数大小，Llama3-8B-Instruct 是经过指令微调的版本，针对对话用例进行了优化，在常见的行业基准测试中优于许多可用的开源聊天模型。	bda59c34e4424598bbd5930eba713fbf
Llama3-70B-Instruct	Meta 开发并发布了 Meta Llama 3 系列大型语言模型 （LLM），包含 8B	6192ed0cb6334302a2c

	和 70B 两种参数大小，Llama3-70B-Instruct 是经过指令微调的版本，针对对话用例进行了优化，在常见的行业基准测试中优于许多可用的开源聊天模型。	32735dbbb6ce3
Qwen1.5-72B-Chat	通义千问1.5（Qwen1.5）是阿里云研发的通义千问系列开源模型，是一种基于 Transformer 的纯解码器语言模型，已在大量数据上进行了预训练。该系列包括Base和Chat等多版本、多规模，满足不同的计算需求，这是Qwen1.5-72B-Chat版本。	9d140d415f11414aa05c8888e267a896
Qwen1.5-32B-Chat	Qwen1.5-32B 是 Qwen1.5 语言模型系列的最新成员，除了模型大小外，其在模型架构上除了GQA几乎无其他差异。GQA能让该模型在模型服务时具有更高的推理效率潜力。这是Qwen1.5-32B-Chat版本。	12d5a37bf1ed4bf9b1cb8e446cfa60b3
InternLM2-Chat-7B	InternLM2-Chat-7B 是书生· 浦语大模型系列中开源的 70 亿参数库模型和针对实际场景量身定制的聊天模型。InternLM2相比于初代InternLM，在推理、数学、代码等方面的能力提升尤为显著，综合能力领先于同量级开源模型。	50beebff68b34803bd71d380e49078f5
Qwen2-7B-Instruct	Qwen2-7B-Instruct是 Qwen2大型语言模型系列中覆盖70亿参数的指令	0e97efbf3aa042ebbaf0

	调优语言模型，支持高达 131,072 个令牌的上下文长度，能够处理大量输入。	b2d358403b94
Qwen-VL-Chat	Qwen-VL-Chat模型是在阿里云研发的大规模视觉语言模型 Qwen-VL 系列的基础上，使用对齐机制打造的视觉AI助手，该模型有更优秀的中文指令跟随，支持更灵活的交互方式，包括多图、多轮问答、创作等能力。	e8c39004ff804ca699d47b9254039db8
StableDiffusion-V2.1	StableDiffusion-V2.1是由 Stability AI 公司推出的基于深度学习的文生图模型，它能够根据文本描述生成详细的图像，同时也可以应用于其他任务，例如图生图，生成简短视频等。	40f9ae16e840417289ad2951f5b2c88f
Deepseek-V2-Lite-C...	Deepseek-V2-Lite-Chat是一款强大的开源专家混合（MoE）语言聊天模型，具有16B参数，2.4B活动参数，使用5.7T令牌从头开始训练，其特点是同时具备经济的训练和高效的推理。	0855b510473e4ec3a029569853f64974
Qwen2.5-72B-Instruct	Qwen2.5系列发布了许多基本语言模型和指令调整语言模型，参数范围从0.5到720亿个参数不等。Qwen2.5-72B-Instruct模型是Qwen2.5系列大型语言模型指令调整版本。	d9df728b30a346afb74d2099b6c209aa

Gemma2-9B-IT	Gemma2-9B-IT是Google最新发布的具有90亿参数的开源大型语言模型的指令调优版本。模型在大量文本数据上进行预训练，并且在性能上相较于前一代有了显著提升。该版本的性能在同类产品中也处于领先地位，超过了Llama3-8B和其他同规模的开源模型。	4dae2b9727db46b7b86e84e8ae6530a9
Llama3.2-3B-Instruct	Meta Llama3.2多语言大型语言模型（LLMs）系列是一系列预训练及指令微调的生成模型，包含1B和3B参数规模。Llama3.2指令微调的纯文本模型专门针对多语言对话应用场景进行了优化，包括代理检索和摘要任务。它们在通用行业基准测试中超越了许多可用的开源和闭源聊天模型。这是Llama3.2-3B-Instruct版本。	f7d0baa95fd2480280214bfe505b0e2e
ChatGLM3-6B-32K	ChatGLM3-6B-32K模型在ChatGLM3-6B的基础上进一步强化了对于长文本的理解能力，能够更好的处理最多32K长度的上下文。具体对位置编码进行了更新，并设计了更有针对性的长文本训练方法，在对话阶段使用 32K 的上下文长度训练。	98b6d84f6b15421886d64350f2832782
CodeGemma-7B-IT	CodeGemma是构建在Gemma之上的轻量级开放代码模型的集合。CodeGemma-7B-IT模型是CodeGemma系列模型之一，是一种文本到文	fa8b78d2db034b6798c894e30fba1173

	本和文本到代码的解码器模型的指令调整变体，具有70亿参数，可用于代码聊天和指令跟随。	
Qwen2.5-Math-7B-I...	Qwen2.5-Math系列是数学专项大语言模型Qwen2-Math的升级版。系列包括1.5B、7B、72B三种参数的基础模型和指令微调模型以及数学奖励模型Qwen2.5-Math-RM-72B，Qwen2.5-Math-7B-Instruct的性能与Qwen2-Math-72B-Instruct相当。	ea056b1eedfc479198b49e2ef156e2aa
DeepSeek-Coder-V...	DeepSeek-Coder-V2-Lite-Instruct是一款强大的开源专家混合（MoE）语言聊天模型，具有16B参数，2.4B活动参数。该模型基于DeepSeek-V2进一步预训练，增加了6T Tokens，可在特定的代码任务中实现与GPT4-Turbo相当的性能。	f23651e4a8904ea589a6372e0e860b10

5.4 错误处理

- 请求处理过程中出现异常时，服务会对外抛出非200的http状态码，表明当前请求无法正常完成。
- 对于异常的请求响应，请求体中会返回error结构，error中返回具体的错误信息。
- 特殊地，在流式请求中：

- 如果在流式请求接收处理之前发生了异常，如鉴权、参数校验等问题，与普通的非流式一样返回http code，并带有error结构。
- 如果在流式请求已经接收，会先对外返回流式请求连接建立的信息，此时http code为200，而在后续模型流式返回过程中发生了异常，会在流失返回的chunk返回error结构，并终止当前的流式请求。

错误error结构

字段名称	二级字段	类型	必选	描述
error		object	是	错误信息
-	code	string	是	平台错误码
-	type	string	是	平台错误类型
-	message	string	是	平台错误详情

错误结果示例

```
{
   "error" : {
     "code" : "500001",
     "type" : "INVOKE_MODEL_ERROR",
     "message" : "服务接口异常，请联系管理员"
   }
}
```

错误码

错误码	错误类型	http code	错误详情	备注
500001	INVOKE_MODEL_ERROR	500	服务接口异常，请联系管理员	
500002	PARAM_ERROR	400	参数不正确，请联系管理员	参数校验失败，需建返回的具体报错信息
500003	SYSTEM_ERROR	500	系统错误，请联系	

			管理员	
500004	INCORRECT_API_KEY_PROVIDED	401	AppKey不正确，请使用正确的AppKey	
600001	INVOKE_TEXT_AUDIT_ERROR	400	很抱歉，关于这个问题我无法提供相应的信息。如果您有其他问题，我将很愿意为您回答。	内容审核失败
600002	TEXT_AUDIT_NOT_PASS	400	很抱歉，关于这个问题我无法提供相应的信息。如果您有其他问题，我将很愿意为您回答。	内容审核失败
600003	TEXT_AUDIT_QUESTION_NOT_PASS	400	很抱歉，关于这个问题我无法提供相应的信息。如果您有其他问题，我将很愿意为您回答。	问题未通过审核
600004	TEXT_AUDIT_ANSWER_NOT_PASS	400	很抱歉，关于这个问题我无法提供相	问题未通过审核

			应的信息。如果您有其他问题，我将很愿意为您回答。	
600005	RESOURCES_TIPS	429	达到免费试用上限	免费试用限制
700001	AK_RPM_RATELIMITING	429	请求RPM超限，请稍后重试	每分钟请求数超限
700002	AK_TPM_RATELIMITING	429	请求TPM超限，请减少tokens后重试	每分钟tokens数超限
700003	AK_BANNED	429	您的请求AK已被封禁，请更换AK或联系后台人员解封后重试	App Key被封禁
700004	MODEL_RPM_RATELIMITING	429	请求RPM超限，请稍后重试	每分钟请求数超限
700005	MODEL_TPM_RATELIMITING	429	请求TPM超限，请减少tokens后重试	每分钟tokens数超限
700006	AK_IPM_RATELIMITING	429	请求IPM超限，请稍后重试	每分钟图片/音频数超限

700007	MODEL_IPM_RATELIMITING	429	请求IPM超限，请稍后重试	每分钟图片/音频数超限

5.5 API

5.5.1 chat 对话API

1 接口描述

-	**描述**
接口名称	对话
请求路径	https://wishub-x1.ctyun.cn/v1/chat/completions
功能描述	针对描述会话的消息列表，模型将返回响应

2 请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer AppKey	鉴权信息填入AppKey。
Content-Type	application/json	

2.2 请求参数

备注：此参数为全平台模型通用，每个模型支持的参数、参数范围可能因模型不同而有所差异，详细可见模型广场内每个模型的API文档。

参数名称	二级参数	三级参数	四级参数	类型	必选	描述
model				string	是	模型ID。
messages				array	是	用户当前输入的期望模型执行指令。一个列表内多个字典，支持多轮对话。 对话列表，每个列表项为一个message object，message object中包含用户role和content两部分信

						息： role可选值为user、assistant、system； role为system时，不校验content空值，且message中system只能位于开头，即messages[0]位置； role为user时说明是用户提问，role为assistant时说明是模型回答，而content为实际的对话内容； 单轮/多轮对话中，最后一个message中role必须为user，content为用户输入的最新问题，其余结果除system角色外都为历史信息拼接送入messages中，assistant和user的role只能交替出现，assistant后只能跟user，user后只能跟assistant。
-	role			string	否	对话角色，role类型枚举值：user、assistant、system。
-	content			string/array	是	对话内容，内容目前有两种格式： string,array。 string类型：表示文本对话内容。

						array类型：表示多个对话内容列表，每个列表项为一个content object，每个content object包含type、image_url、text等信息。 type可选值为text、image_url。 type为text时，取text字段作为对话内容。 type为image_url时，取image_url字段作为对话内容。
-	-	type		string	否	对话内容类型，type类型枚举值：text,image_url。
-	-	text		string	否	文本对话内容，type为text时传入。
-	-	image_url		object	否	图片对话内容，type为image_url时传入。
-	-	-	url	string	否	图片对话内容中的图片地址，目前可以为二进制数据的base64编码。

frequency_penalty				float	否	频率惩罚。它影响模型如何根据文本中词汇token的现有频率惩罚新词汇token。值大于0，会根据新标记在文本中的现有频率来惩罚新标记，从而降低模型逐字重复同一行的可能性。 一般取值范围[-2, 2]，**具体取值范围、默认值需见对应模型。**
max_tokens				int	否	最大生成长度。控制最大生成长度，超过该值则截断。 一般取值范围(0, 2048]，**具体取值范围、默认值需见对应模型。**
n				int	否	1-n个choices。
presence_penalty				float	否	存在惩罚。用户控制模型生成时整个序列中的重复度。 一般取值范围[-2.0, 2.0]，**具体取值范围、默认值需见对应模型。**
response_format				object	否	返回格式。

-	type			string	否	返回格式枚举值：text,json_object。
seed				int	否	随机种子。用于指定推理过程的随机种子，相同的seed值可以确保推理结果的可重现性，不同的seed值会提升推理结果的随机性。 一般取值范围(0, 9223372036854775807]，**具体取值范围、默认值需见对应模型。**
stop				string/ array	否	生成停止标识。当模型生成结果以stop中某个元素结尾时，停止文本生成。
stream				bool	否	是否以流式接口的形式返回数据。默认为False，非流式。
stream_options				object	否	流式选项，stream为True有效。
-	include_usage			bool	否	是否在返回中包含usage，stream为True有效。 **取值为True时，会在流式返回的最后一个chunk里返回usage信息**

						，并该chunk中choices列表为空 •
temperature				float	否	温度采样。该值越高生成文本的多样性越高，该值越低生成文本的确定性越高。 一般取值范围(0, 2)，**具体取值范围、默认值需见对应模型。**
top_k				int	否	top-k 采样。取值越大，生成的随机性越高；取值越小，生成的确定性越高。 一般取值范围[1, 100]，**具体取值范围、默认值需见对应模型。**
top_p				float	否	top-p 采样。该值越高生成文本的多样性越高，该值越低生成文本的确定性越高。该值为 0 时没有随机性。 一般取值范围(0, 1]，**具体取值范围、默认值需见对应模型**

user				string	否	用户唯一身份ID。

请求参数示例

```
{
  "model": "1234567890",   // 模型ID
  "messages": [
    {
      "role": "user",
      "content": "Hello!"
    }
  ]
}
```

3 请求返回

3.1非流式返回

3.1.1非流式正常返回

字段名称	二级字段	三级字段	字段类型	描述
id			string	唯一标识符
choices			string	choices列表
-	index		int	choice索引
-	message		object	模型生成的消息
-	-	role	string	对话角色
-	-	content	string	对话消息内容
-	finish_reason		string	模型停止生成标记的原因。 stop: 模型生成遇到自然停止点或提

				供的停止序列； length: 达到请求中 指定的最大标记数 ； content_filter：如 果由于内容过滤器 中的标志而省略了 内容 tool_calls/function _call： 模型调用 了函数。
created			int	Unix时间戳（以秒为单位）。
model			string	调用的模型名称。
object			string	返回的对象类型。 非流式返回始终为 ：chat.completion

usage			object	请求使用情况的统计信息。
-	completion_tokens		int	生成token数。
-	prompt_tokens		int	输入token数。
-	total_tokens		int	使用的token总数（prompt + completion）。

返回结果示例

```
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "xxx-chat",
  "choices": [{
    "index": 0,
    "finish_reason": "stop",
    "message": {
      "role": "assistant",
      "content": "\n\nHello there, how may I assist you today?"
    }
  }],
  "usage": {
    "prompt_tokens": 9,
    "completion_tokens": 12,
```

3.1.2 非流式异常返回

非流式异常返回时：

- http code 返回非200。
- http body 中返回 error 结构，error结构中包含code、type、message、param等信息，具体可见OpenAPI接口文档中的error结构描述及错误码部分介绍。

错误结果示例

```
{
  "error" : {
    "code" : "500001",
    "type" : "INVOKE_MODEL_ERROR",
    "message" : "服务接口异常，请联系管理员"
  }
}
```

3.2 流式返回

3.2.1 流式正常返回

字段名称	二级字段	三级字段	字段类型	描述
id			string	唯一标识符
choices			string	choices列表
-	index		int	choice索引
-	delta		object	模型生成的消息
-	-	role	string	对话角色
-	-	content	string	对话消息内容
-	finish_reason		string	模型停止生成标记的原因。 stop: 模型生成遇到自然停止点或提供的停止序列； length: 达到请求中指定的最大标记数；

				content_filter：如果由于内容过滤器中的标志而省略了内容 tool_calls/function_call： 模型调用了函数。
created			int	Unix时间戳（以秒为单位）。
model			string	调用的模型名称。
object			string	返回的对象类型。 流式返回始终为：chat.completion.chunk
usage			object	请求使用情况的统 计信息。 仅在stream_options: {"include_usage": true}设置时显示。如果存在，则它包含一个 null 值，但

				最后一个块包含整个请求的token使用情况的统计信息。
-	completion_tokens		int	生成token数。
-	prompt_tokens		int	输入token数。
-	total_tokens		int	使用的token总数（prompt + completion）。

返回结果示例

```
1        {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"role":"assistant"},"finish_reason":null}]}
2        {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"content":"Hello"},"finish_reason":null}]}
3        ....
4        {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{},"finish_reason":"stop"}]}
5
```

stream_options.include_usage 为 True 时多返回一条包含usage流式消息。

```
1        {"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268199,"model":"xxx-chat", "choices":[], "usage":
{"prompt_tokens": 9,"completion_tokens": 120,"total_tokens": 129}}
```

3.2.2 流式异常返回

流式异常分为两种：

- 如果在流式请求接收处理之前发生了异常，如鉴权、参数校验等问题，与普通的非流式一样返回http code，并带有error结构。
- 如果在流式请求已经接收，会先对外返回流式请求连接建立的信息，此时http code为200，而在后续模型流式返回过程中发生了异常，会在流失返回的chunk返回error结构，并终止当前的流式请求。

流式请求建立后的异常返回示例

```
1	{"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"role":"assistant"},"finish_reason":null}]}
2	{"id":"chatcmpl-123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"content":"Hello"},"finish_reason":null}]}
3	....
4	{"error":{"code":"500001","type":"INVOKE_MODEL_ERROR","message":"服务接口异常，请联系管理员"}}
```

4 请求示例代码

```
1	假设平台用户组AppKey=884c8fc4054548a7b1ca1123592f5b7，模型ID=96dcaaaaaaaaaaa5ff55ea377831a，以此为例进行说明。
```

4.1 curl方式请求

```
curl --request POST \
  --url https://wishub-x1.ctyun.cn/v1/chat/completions \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
  --data '{
      "model": "96dcaaaaaaaaaaa5ff55ea377831a",
      "messages": [
          {
              "role": "user",
              "content": "Hello"
          }
      ]
}'
```

4.2 python方式请求

```
import json
import requests
URL = "https://wishub-x1.ctyun.cn/v1/chat/completions"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
data = {
    "model": "96dcaaaaaaaaaaa5ff55ea377831a",
    "messages": [
        {"role": "user", "content": "Hello"}
    ],
    "stream": True
}
try:
    response = requests.post(URL, headers=headers, json=data, stream=True)
```

4.3 openai 客户端示例代码

```
import openai
from openai import OpenAI
client = OpenAI(base_url="https://wishub-x1.ctyun.cn/v1", api_key="884c8fc4054548a7b1ca1123592f5b7")
messages = [
    {"role": "user", "content": "Hello"}
]
try:
    stream = client.chat.completions.create(
        model="96dcaaaaaaaaaaa5ff55ea377831a",
        messages=messages,
        stream=True
    )
    # 流式
    for chunk in stream:
        print(chunk.choices[0].delta.content or "", end="", flush=True)
except openai.APIStatusError as e:
```

5.5.2 image 文本生图API

1 接口描述

-	**描述**
接口名称	文本生图
请求路径	https://wishub-x1.ctyun.cn/v1/images/generations
功能描述	基于提示创建图像

2 请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer AppKey	鉴权信息填入AppKey。
Content-Type	application/json	

2.2 请求参数

备注：此参数为全平台模型通用，每个模型支持的参数、参数范围可能因模型不同而有所差异，详细可见模型广场内每个模型的API文档。

参数名称	参数类型	必选	描述
model	string	是	模型ID。
prompt	string	是	所需图像的文本描述，最大长度1000。
negative_prompt	string	否	避免生成负面图像的提示，最大长度1000。

n	int	否	要生成的图像数。 一般取值为1，**具体取值范围、默认值需见对应模型。**
num_inference_steps	int	否	去噪步骤的数量。去噪步骤越多，通常可以得到更高质量的图像，但推理速度会变慢。 一般取值范围 [1, 50]，**具体取值范围、默认值需见对应模型。**
quality	int	否	生成图像质量。openai参数，平台未启用。
size	string	否	输出的图像尺寸（高x宽）。 一般取值范围360x640,480x640,512x512,640x360,640x480,720x1280,1024x1024,1280x720，默认值为512x512，**具体取值范**

			围、默认值需见对应模型。
response_format	string	否	返回图像的格式。 一般取值范围url,b64_json，默认值为b64_json，**具体取值范围、默认值需见对应模型。**
style	string	否	生成图像风格。一般取值范围standard,portrait,landscape,cartoon,technology,photography,concept，默认值为standard，**具体取值范围、默认值需见对应模型。**
user	string	否	用户唯一身份ID。

请求参数示例

```
{
  "model": "bfetrcdggsdfsdf",
  "prompt": "A cute baby sea otter",
  "n": 1,
  "size": "512x512"
}
```

3 请求返回

3.1请求正常返回

字段名称	二级字段	字段类型	描述
created		string	Unix时间戳（以秒为单位）。
data		array	图像列表

-	b64_json	string	b64_json 格式图片
-	url	string	url 格式图片
-	revised_prompt	string	实际使用的修改后的 prompt

返回结果示例

```
{
  "id": "1714459832",
  "data": [{
    "b64_json": "xxxxxxxxxxxxxxxxxxx"
  }]
}
```

3.2异常返回

异常返回时：

- http code 返回非200。

- http body 中返回 error 结构，error结构中包含code、type、message、param等信息，具体可见OpenAPI接口文档中的error结构描述及错误码部分介绍。

错误结果示例

```
{
  "error" : {
    "code" : "500001",
    "type" : "INVOKE_MODEL_ERROR",
    "message" : "服务接口异常，请联系管理员"
  }
}
```

4 请求示例代码

```
假设平台用户组AppKey=884c8fc4054548a7b1ca1123592f5b7，模型ID=96dcaaaaaaaaaaaa5ff55ea377831a，以此为例进行说明。
```

4.1 curl方式请求

```
curl --request POST \
  --url https://wishub-x1.ctyun.cn/v1/images/generations \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
  --data '{
      "model": "96dcaaaaaaaaaaa5ff55ea377831a",
      "prompt" : "A cute baby sea otter",
      "n" : 1,
      "size" : "512x512"
}'
```

4.2 python方式请求

```
import json
import requests
URL = "https://wishub-x1.ctyun.cn/v1/images/generations"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
data = {
        "model": "96dcaaaaaaaaaaa5ff55ea377831a",
        "prompt" : "A cute baby sea otter",
        "n" : 1,
        "size" : "512x512"
}
try:
    response = requests.post(URL, headers=headers, json=data)
    if response.status_code != 200:
```

4.3 openai 客户端示例代码

```python
import openai
from openai import OpenAI
client = OpenAI(base_url="https://wishub-x1.ctyun.cn/v1", api_key="884c8fc4054548a7b1ca1123592f5b7")
try:
    response = client.images.generate(
        model="96dcaaaaaaaaaaa5ff55ea377831a",
        prompt="A cute baby sea otter",
    )
    print(f"{response.data[0].b64_json}")
except openai.APIStatusError as e:
    print(f"APIStatusError: {e.status_code}, {e.message}, {e.body}")
except openai.APIError as e:
    print(f"APIError: {e.body}")
except Exception as e:
    print(f"Exception: {e}")
```

6 平台功能 OpenAPI

6.1 平台功能API使用说明

OpenAPI门户提供了产品的API 文档、API调试、SDK中心等。

关于用户如何使用产品功能API的详细介绍，请参见API文档。您可以在OpenAPI门户了解到具体的调用前必知、API概览、如何调用API以及具体的API的接口详细说明。

7 常见问题

7.1 计费相关

1、 一站式智算服务平台支持哪些计费方式？

一站式智算服务平台支持包周期和按需计费两种计费模式。

2、后付费的账单是月结算还是日结算？

一站式智算服务平台是按照小时结算，每小时结算账单。

7.2 平台操作

1、平台已预置的模型有哪些？

进入模型服务模块，选择在线服务，点击【预置服务】，可以看到平台预置的模型，平台预置了多款等基础大模型，包括通义千问、Llama、ChatGLM 等系列，可以直接使用。不同的基础模型的参数和能力不同，我们将持续推出不同能力方向的模型。

2、平台提供的开发工具有哪些？

JupyterLab和Visual Studio Code (VSCode)。

3、GPU模型脚本如何迁移到昇腾NPU上？

- 新建脚本train.py，写入以下原GPU脚本代码。
- 添加以下库代码。

```
import time
import torch
.....
import torch_npu
from torch_npu.npu import amp # 导入 AMP 模块
from torch_npu.contrib import transfer_to_npu    # 使能自动迁移
```

4、IDE 无法打开图片或预览 MD 文件，该怎么办？

- 无法在 IDE 打开图片或预览 MD 文件，这是由于浏览器设置问题，需要开启 Chrome 浏览器 的 unsafely-treat-insecure-origin-as-secure 功能。
- 进入 Chrome Flag 管理界面配置：chrome://flags/#unsafely-treat-insecure-origin-as-secure

5、一站式服务平台预置的镜像有哪些？

进入智算资产模块，选择我的镜像，点击【系统内置镜像】，可以看到平台内置的镜像，包括 PyTorch、TensorFlow 等。

6、如果在自定义镜像列表看不到容器镜像服务分享过来的镜像，怎么办？

请进行以下检查：

- 容器镜像服务所选区域与平台是不是同资源池。
- 截止时间是不是大于等于当前时间，超出截止时间后共享失效。
- 共享镜像状态是不是启用。

7、如何在模型开发 JupyterLab 和 VSCode 中使用自定义镜像？

需要将对应的软件安装包打包进您的自定义镜像中，具体方式见下方。

- 在 docker file 中具体执行命令。

```
#VSCode
curl -fsSL https://code-server.dev/install.sh | sh
code-server --install-extension ms-python.python
```

```
#Jupyterlab
pip install jupyterlab
```

- 将打包好的镜像在本地起起来，然后运行如下命令安装软件，安装完成后，执行 docker commit {容器名称}，打包成新镜像后，即可上传。

````
    #VSCode
    curl -fsSL https://code-server.dev/install.sh | sh
    code-server --install-extension ms-python.python
```

```
   #Jupyterlab
pip install jupyterlab
```
````

8、我想基于自己的模型进行二次训练微调怎么做？
````

可以先在模型管理中导入自己的模型，在 JupyterLab 和 VSCode 创建训练任务，在挂载模型的选项中选择【模型管理】，选择已导入需要二次训练微调的模型，即可挂载自己的模型进行训练。

9、一站式智算服务平台是否支持 IB 和 NVlink？

当前昇腾集群暂不支持。

10、如何给子账号配置资源使用的限额？

主账号管理员进入运营后台，在配置设置模块，可支持设置单用户最大同时使用的 GPU/CPU 数量以及并行文件存储初始分配额度。

7.3 如何联系我们

1、产品使用方面的问题如何反馈？

您可通过天翼云官网工单或者客服电话【400-810-9889】进行反馈。