高性能应用服务 HAI

实践教程

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

实践教程
快速使用 Hunyuan3D 模型

最近更新时间：2025-06-26 14:57:42

背景介绍

腾讯混元 3D 模型是一款先进的大规模 3D 资产创作系统，它可以用于生成带有高分辨率纹理贴图的高保真度 3D 模型。该系统包含两个基础组件：一个大规模几何生成模型——混元 3D-DiT，以及一个大规模纹理生成模型——混元 3D-Paint。几何生成模型基于扩散模型构建，旨在生成与给定条件图像精确匹配的几何模型，为下游应用奠定坚实基础。

快速使用

步骤一：创建 Hunyuan3D 应用

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。

- **选择应用：**选择社区应用，应用选择 **混元Hunyuan3D-2**。
- **地域：**建议选择靠近您实际地理位置的地域，降低网络延迟、提高您的访问速度。
- **算力方案：**选择合适的算力套餐。

> **说明：**
> 在单并发访问模型的情况下，建议配置如下：
>
> | 模型 | 推荐算力套餐 |
> | --- | --- |
> | Hunyuan3D-2 | GPU 性能型 |
>
> 具体算力套餐配置及参数可参考 套餐类型 。

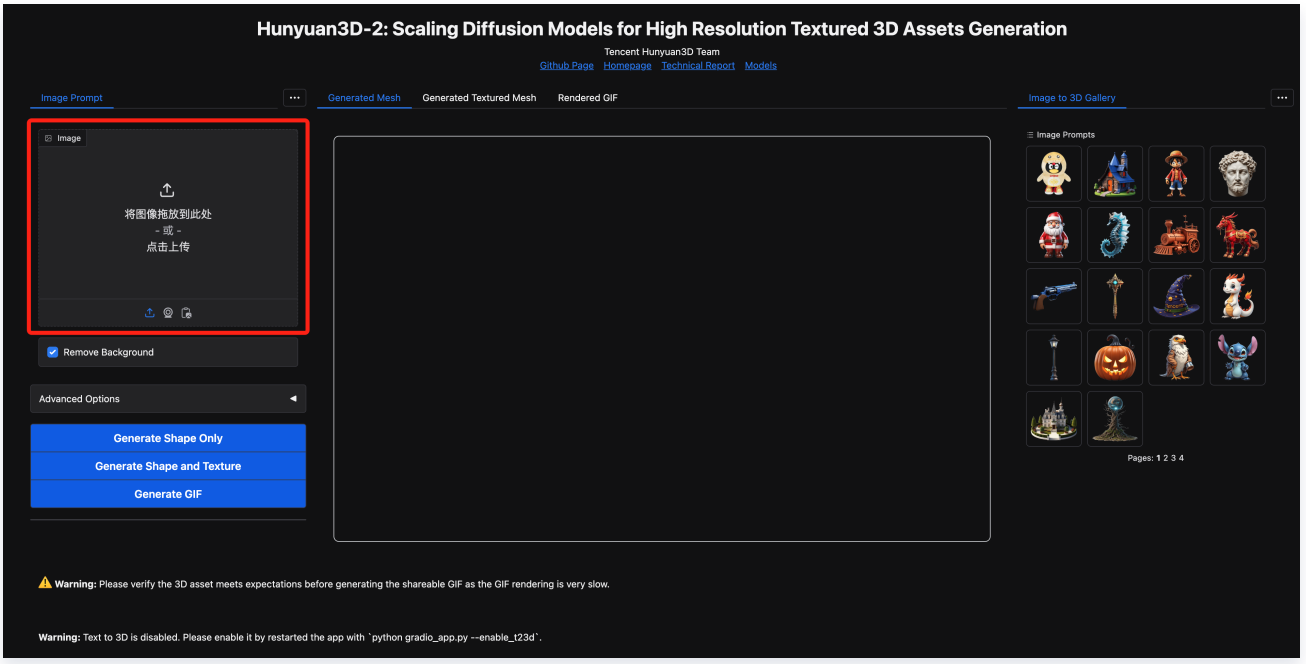
- **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
- **购买数量：**默认1台。

3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页面。同时您将在站内信中收到登录密码。此时，建议通过可视化界面（GUI）使用 Hunyuan3D 模型。
6. 您可以在此页面查看实例的详细配置信息，到此为止，说明您的 Hunyuan3D 应用实例购买成功。

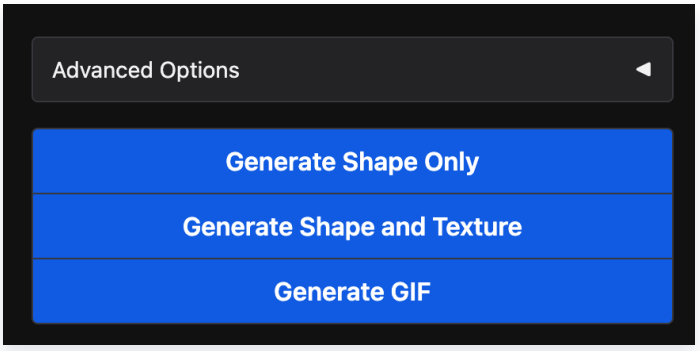
步骤二：使用 Hunyuan3D 模型

1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > Gradio WebUI。

2. 在新窗口中，在 Image Prompt 处**拖拽或点击上传图片**。非透明背景的图片建议勾选 **Remove Background**。



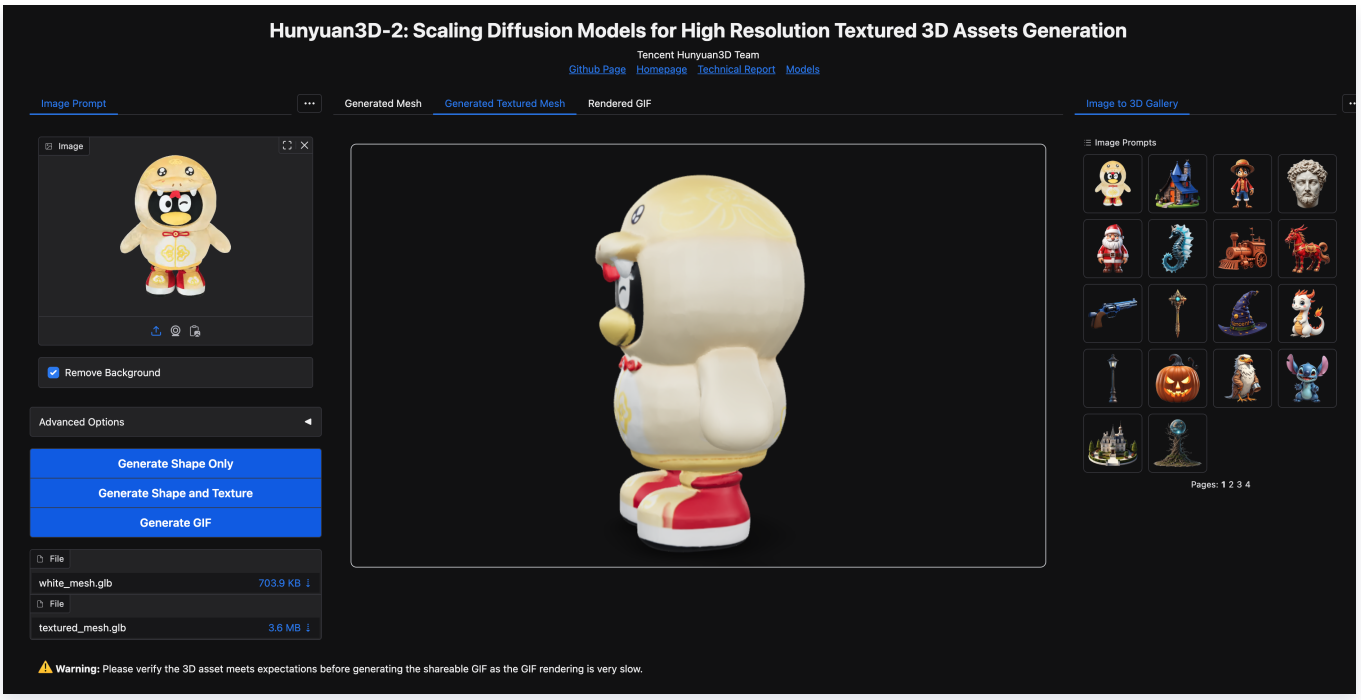
3. 选择需要生成的内容，目前支持**形状、纹理、GIF** 的生成。



- Generate Shape Only：仅生成形状
- Generate Shape and Texture：生成形状和纹理
- Generate GIF：生成 GIF 动图

> ⚠ **注意：**
> 由于 GIF 渲染非常耗时，请在生成 GIF 之前验证形状及纹理是否符合预期。

4. 等待 AI 生成完成，即可获得 3D 资产。



相关链接

- Github 地址：https://github.com/Tencent/Hunyuan3D-2
- Hugging Face 模型地址：https://huggingface.co/tencent/Hunyuan3D-2

快速使用 QwQ-32B 模型

最近更新时间：2025-06-26 14:57:42

背景介绍

QwQ-32B 是一款拥有 320 亿参数的模型，其性能可与具备 6710 亿参数（其中 370 亿被激活）的 DeepSeek-R1 媲美。此外，在 QwQ-32B 中集成了与 Agent 相关的能力，使其能够在使用工具的同时进行批判性思考，并根据环境反馈调整推理过程。
HAI 已提供 QwQ-32B 模型预装环境，用户可在 HAI 中快速启动，进行测试并接入业务。

快速使用

步骤一：创建 QwQ-32B 应用

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。
   - **选择应用：**选择社区应用，应用选择 **QwQ-32B**。
   - **地域：**建议选择靠近您实际地理位置的地域，降低网络延迟、提高您的访问速度。
   - **算力方案：**选择合适的算力套餐。

     说明：
     在单并发访问模型的情况下，建议最低配置如下：

模型	参数量级	推荐算力套餐
QwQ-32B	32B	GPU 进阶型

     具体算力套餐配置及参数可参考 套餐类型 。
   - **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
   - **购买数量：**默认1台。
3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页面。同时您将在站内信中收到登录密码。此时，可通过可视化界面（GUI）或命令行（Terminal）使用 QwQ-32B 模型。
6. 您可以在此页面查看实例的详细的配置信息，到此为止，说明您的 **QwQ-32B** 应用实例购买成功。

步骤二：使用 QwQ-32B 模型

等待几分钟创建完成后，您将在站内信中收到登录密码。此时，可通过可视化界面 (GUI) 或命令行 (Terminal) 使用 QwQ-32B 模型。

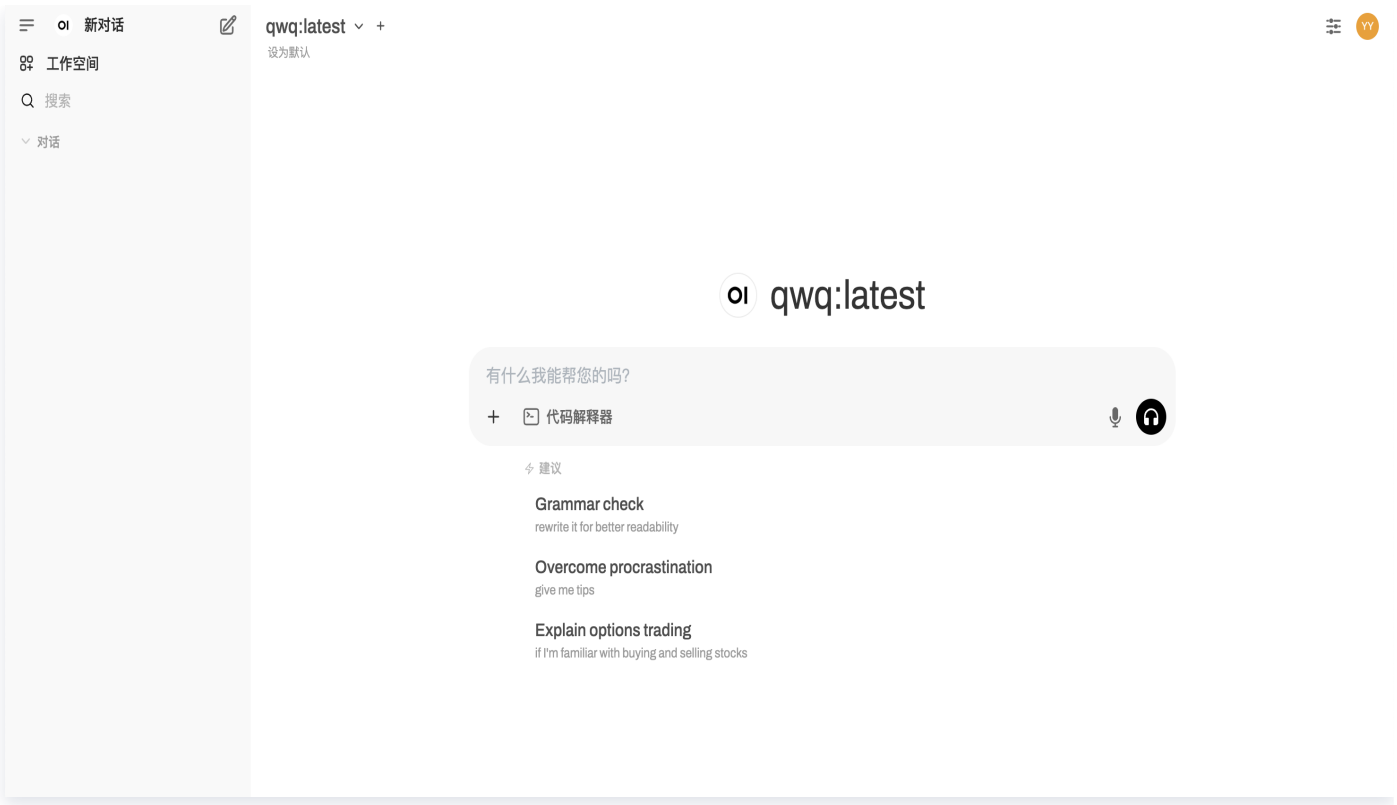
通过 OpenWebUI 可视化界面使用（推荐）

1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **OpenWebUI**。

2. 在新窗口中，单击**开始使用**。

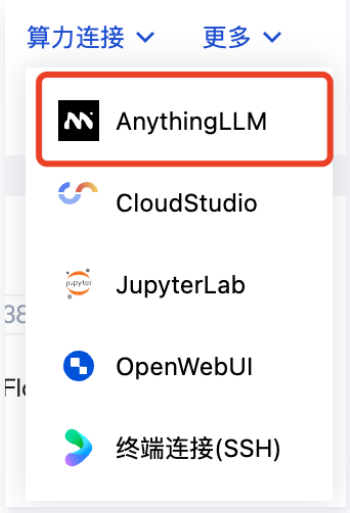
3. 自定义**名称**、**电子邮箱**、**密码**，创建管理员账号。

4. 完成管理员账号创建后，即可开始使用。

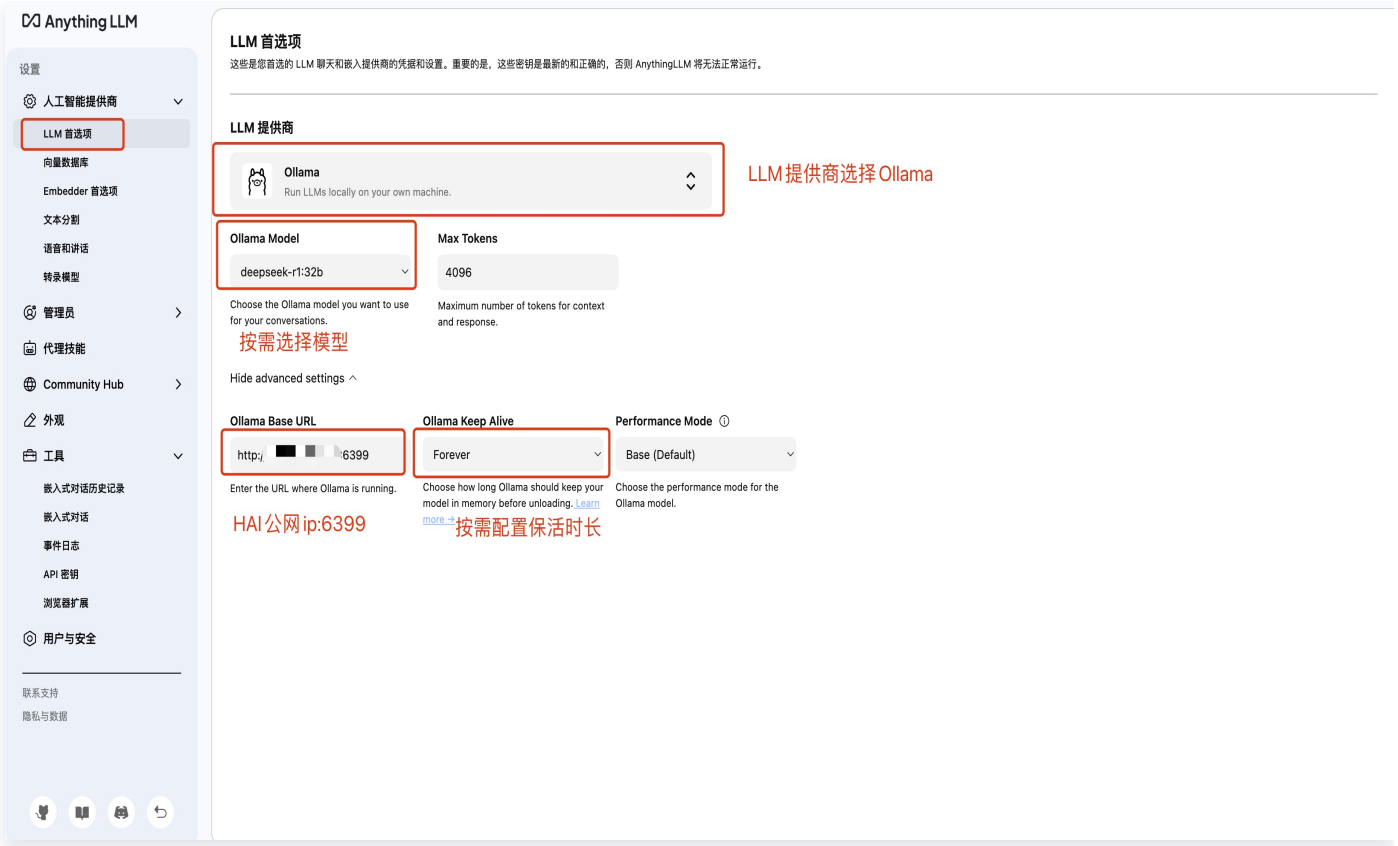


通过 AnythingLLM 可视化界面使用（推荐）

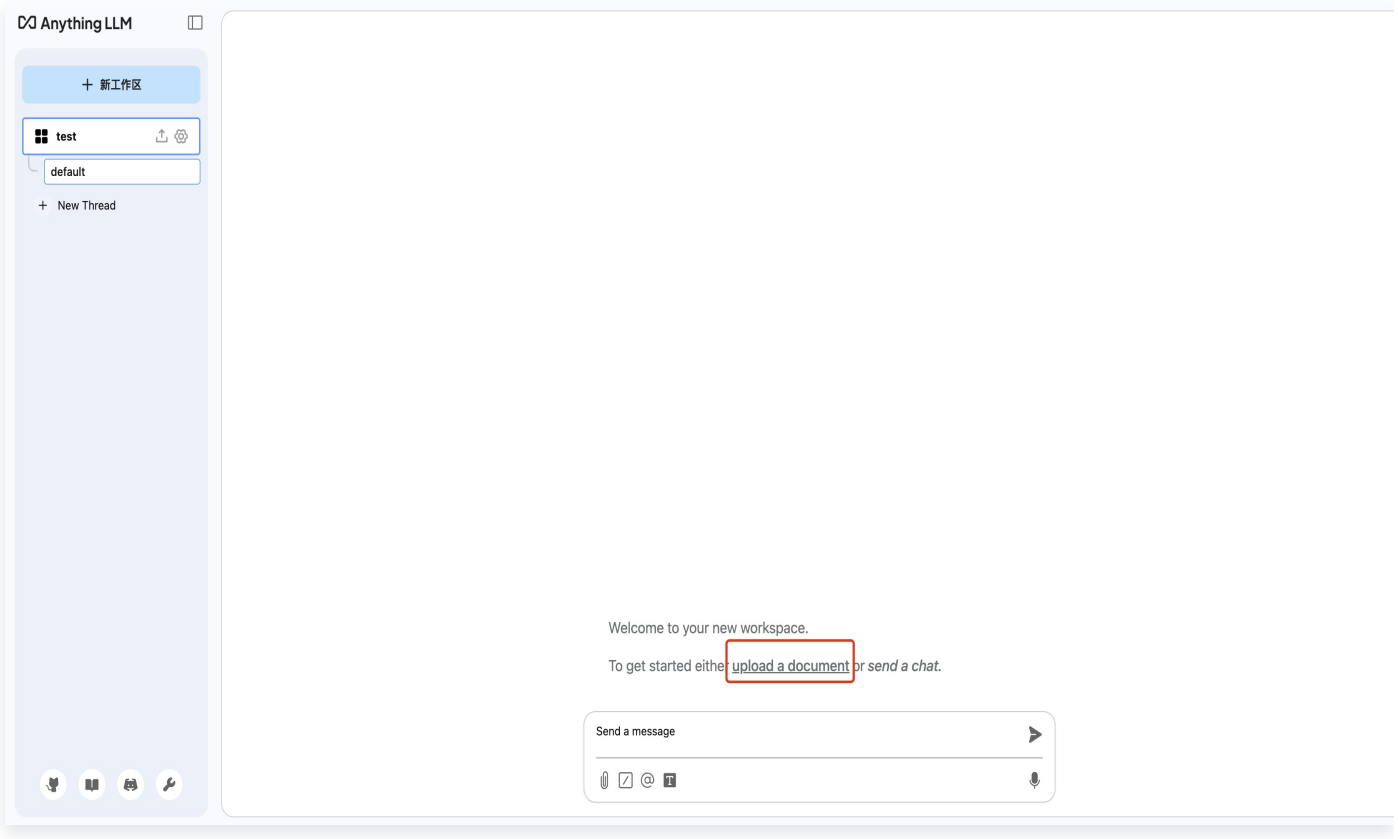
1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **AnythingLLM**。



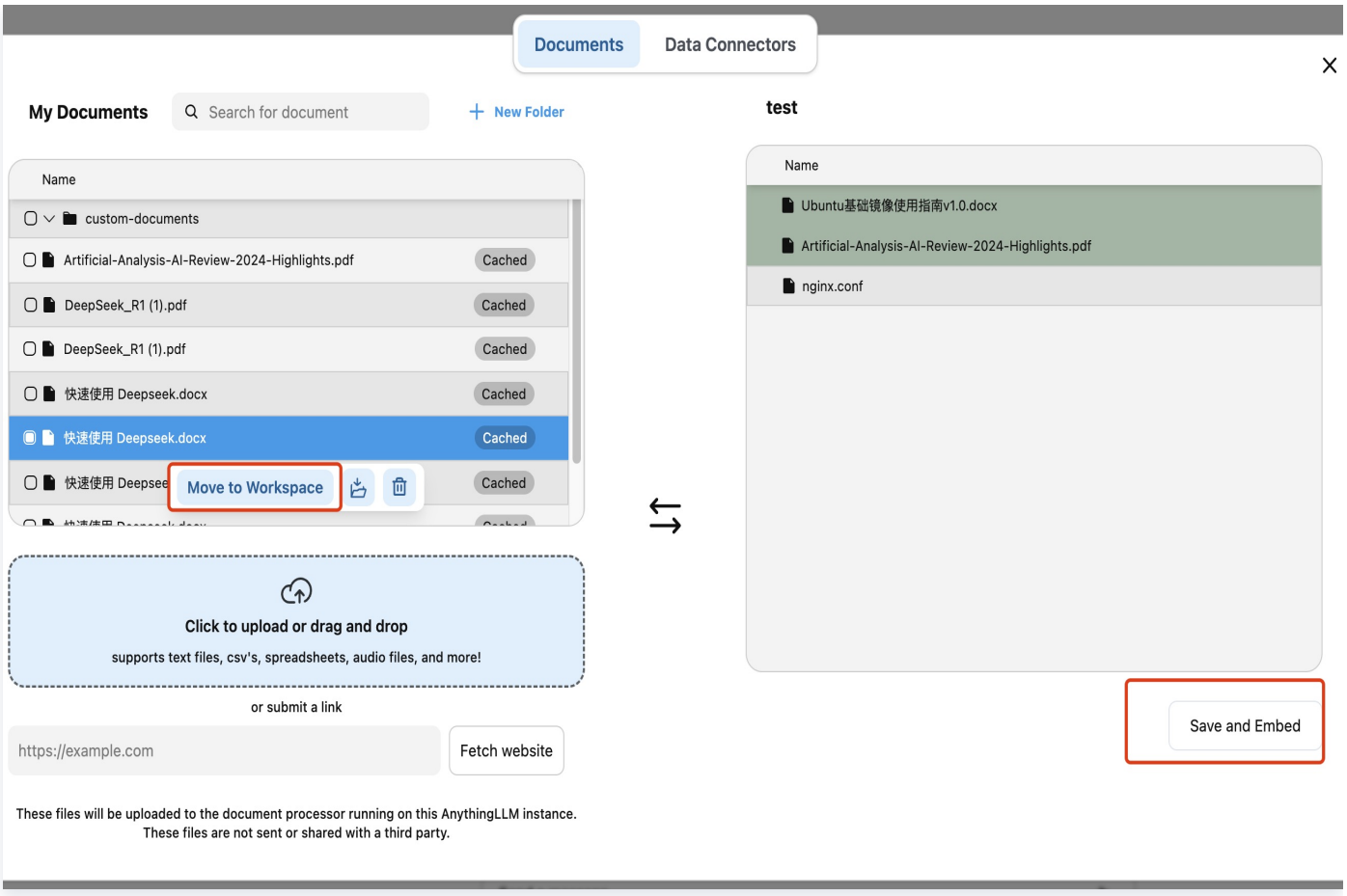
2. 新建窗口后，单击页面左下角**设置**，进入设置页面。单击左侧导航栏 **LLM 首选项**进入配置。
   - 将 **LLM 提供商**选择为 **Ollama**。
   - 将 **Ollama Base URL** 修改为：该台 HAI 实例的公网 IP:6399。
   - 在 **Ollama Model** 处选择需要使用的模型，例如：QwQ-32B。
   - 在 **Ollama Keep Alive** 处按需配置保活时长。（模型在每次超过保活时长后会被移除，再次使用时需重新载入模型，耗时较久，若不存在频繁切换模型诉求，建议将保活时长尽可能调大）



3. 配置完成后，回到项目页面，单击 **upload a document** 上传本地文件。



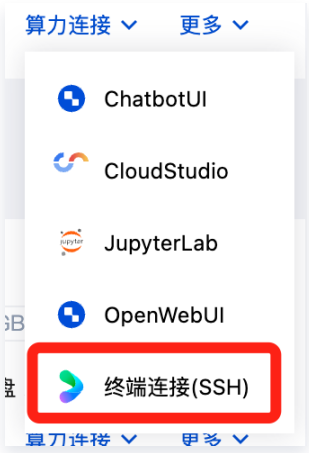
4. 上传文件后，选中希望使用的文件，单击 **Move to Workspace** 将文件添加至项目。单击 **Save and Embed**，完成配置。



5. 您可直接与模型进行对话，模型会根据对话内容智能调用本地知识库内容。

通过终端连接命令行使用

1. 在 高性能应用服务 HAI 控制台，选择**算力连接 > 终端连接(SSH)**。



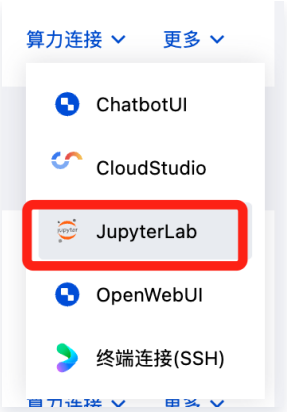
2. 在弹出的 OrcaTerm 登录页面中，输入站内信中的登录密码，单击**登录**。

3. 登录成功后，输入以下命令加载默认模型：

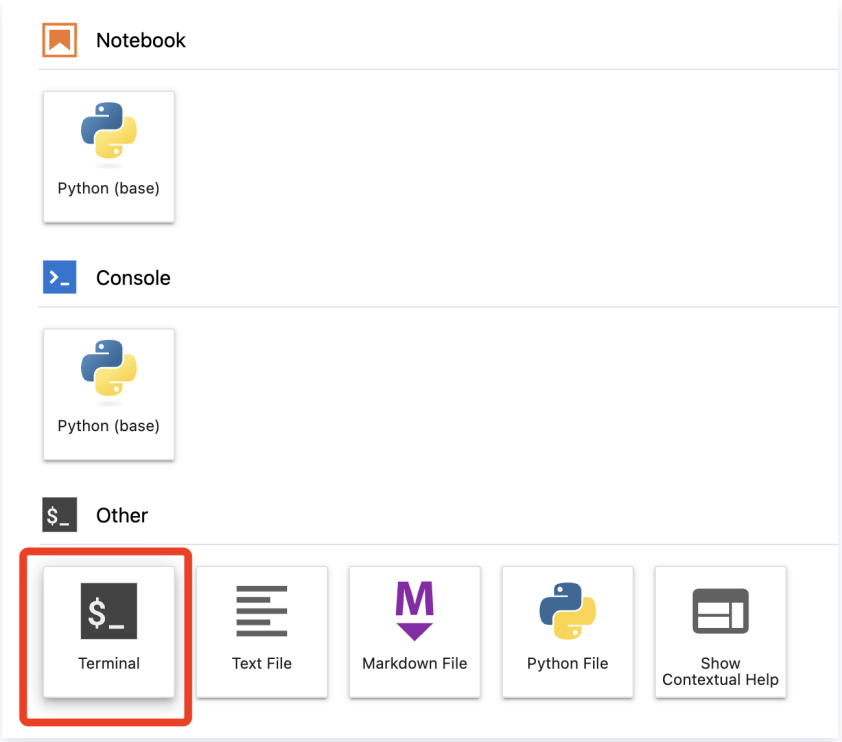
```
ollama run qwq
```

通过 JupyterLab 命令行使用

1. 在 高性能应用服务 HAI 控制台，选择**算力连接 > JupyterLab**。



2. 新建一个 **Terminal**。



3. 输入以下命令加载默认模型：

```
ollama run qwq
```

进阶使用

API 调用

实例环境中已预装并启动 Ollama serve，该服务支持通过 REST API 进行调用。您可以参考 Ollama API 文档，以了解具体的调用方式和方法。

常见问题

Ollama/API 的端口号是哪个?

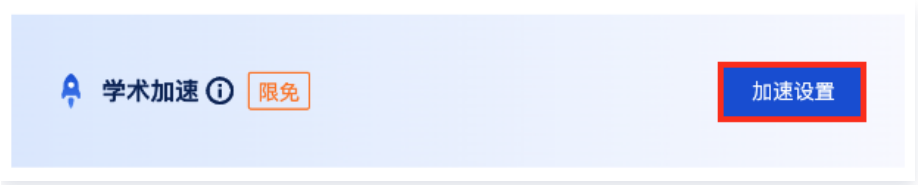
HAI 调用 Ollama 的 API 端口使用 6399，OpenWebUI 端口使用 6699，其他端口详情请参见 常用端口 。

如何通过 API 使用模型?

实例环境中已预装并启动 Ollama serve，该服务支持通过 REST API 进行调用。您可以参考 Ollama API 文档，以了解具体的调用方式和方法。

中国大陆地域通过 Ollama 下载模型速度慢怎么办?

目前北京、上海、广州的资源，可通过 高性能应用服务 HAI 控制台 单击**加速设置**，开启学术加速后，提高资源访问速度。相关能力介绍可参考 开启学术加速 。



提示资源紧张，排队人数过多，如何处理?

由于使用火热，部分地域可能出现售罄情况，无法成功创建实例。已付款项将会原路退回。建议更换地域重新购买或稍后重试。

快速使用 DeepSeek-R1 模型

最近更新时间：2025-06-26 14:57:42

背景介绍

DeepSeek-R1 在后训练阶段大规模使用了强化学习技术，在仅有极少标注数据的情况下，极大提升了模型推理能力。在数学、代码、自然语言推理等任务上，性能比肩 OpenAI o1 正式版。
HAI 已提供 DeepSeek-R1 模型预装环境，用户可在 HAI 中快速启动，进行测试并接入业务。

快速使用

步骤一：创建 DeepSeek-R1 应用

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。
   - **选择应用：**选择社区应用，应用选择 **DeepSeek-R1 AnythingLLM**。
   - **地域：**建议选择靠近自己实际地理位置的地域，降低网络延迟、提高您的访问速度。
   - **算力方案：**选择合适的算力套餐。

     说明：
     在单并发访问模型的情况下，建议最低配置如下：

模型	参数量级	推荐算力套餐
DeepSeek-R1	1.5B/7B/8B/14B	GPU基础型
DeepSeek-R1	32B	GPU进阶型

     具体算力套餐配置及参数可参考 套餐类型 。
   - **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
   - **购买数量：**默认1台。
3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页面。同时您将在站内信中收到登录密码。此时，可通过可视化界面（GUI）或命令行（Terminal）使用 DeepSeek 模型。

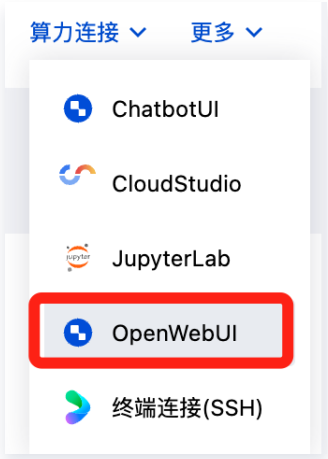
6. 您可以在此页面查看 DeepSeek-R1 详细的配置信息，到此为止，说明您的 DeepSeek-R1 应用实例购买成功。

步骤二：使用 DeepSeek-R1 模型

等待几分钟创建完成后，将在站内信中收到登录密码。此时，可通过可视化界面 (GUI) 或命令行 (Terminal) 使用 DeepSeek 模型。

通过 OpenWebUI 可视化界面使用（推荐）

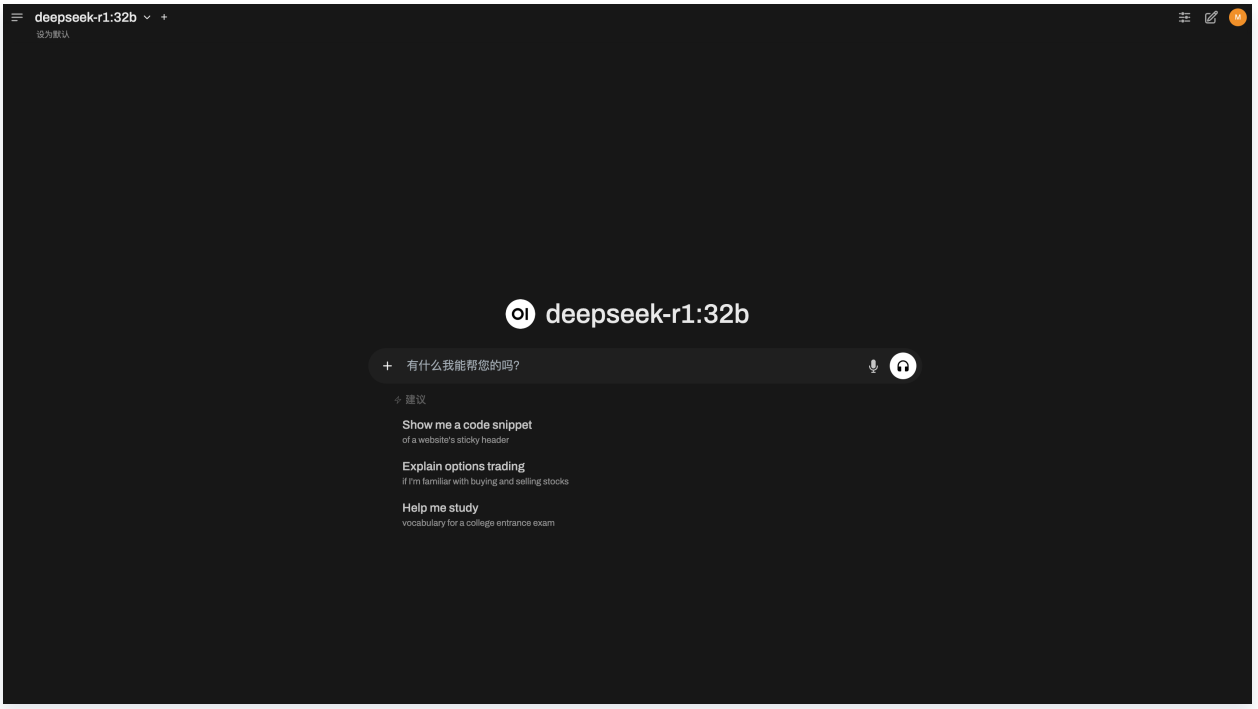
1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **OpenWebUI**。



2. 在新窗口中，单击**开始使用**。

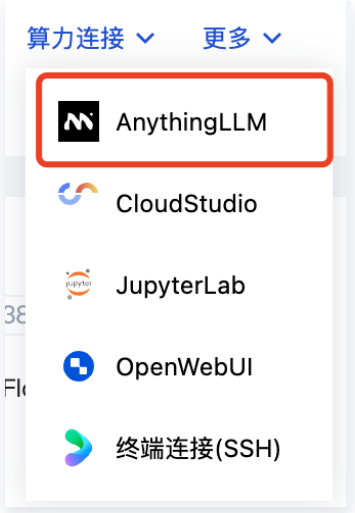
3. 自定义**名称**、**电子邮箱**、**密码**，创建管理员账号。

4. 开始使用

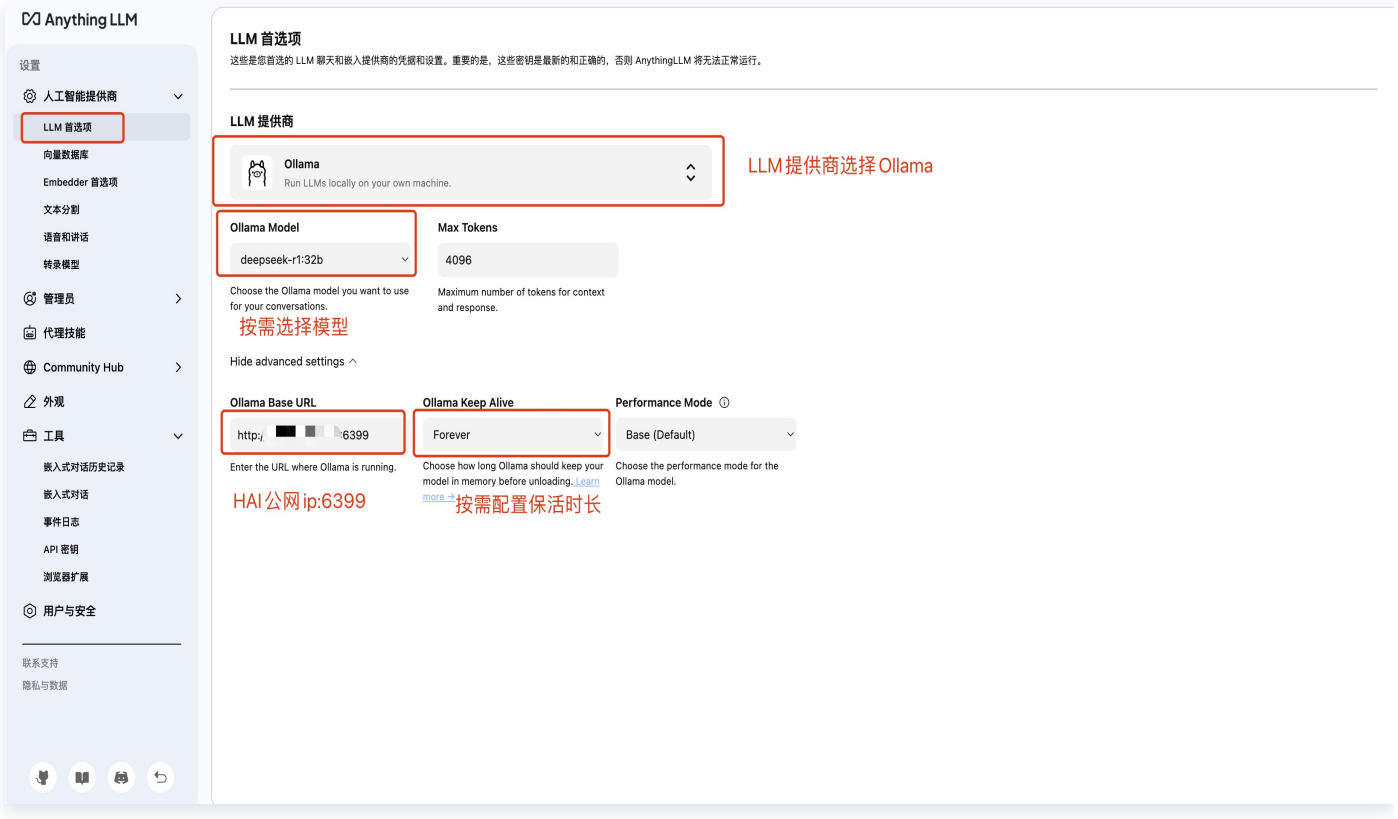


通过 AnythingLLM 可视化界面使用（推荐）

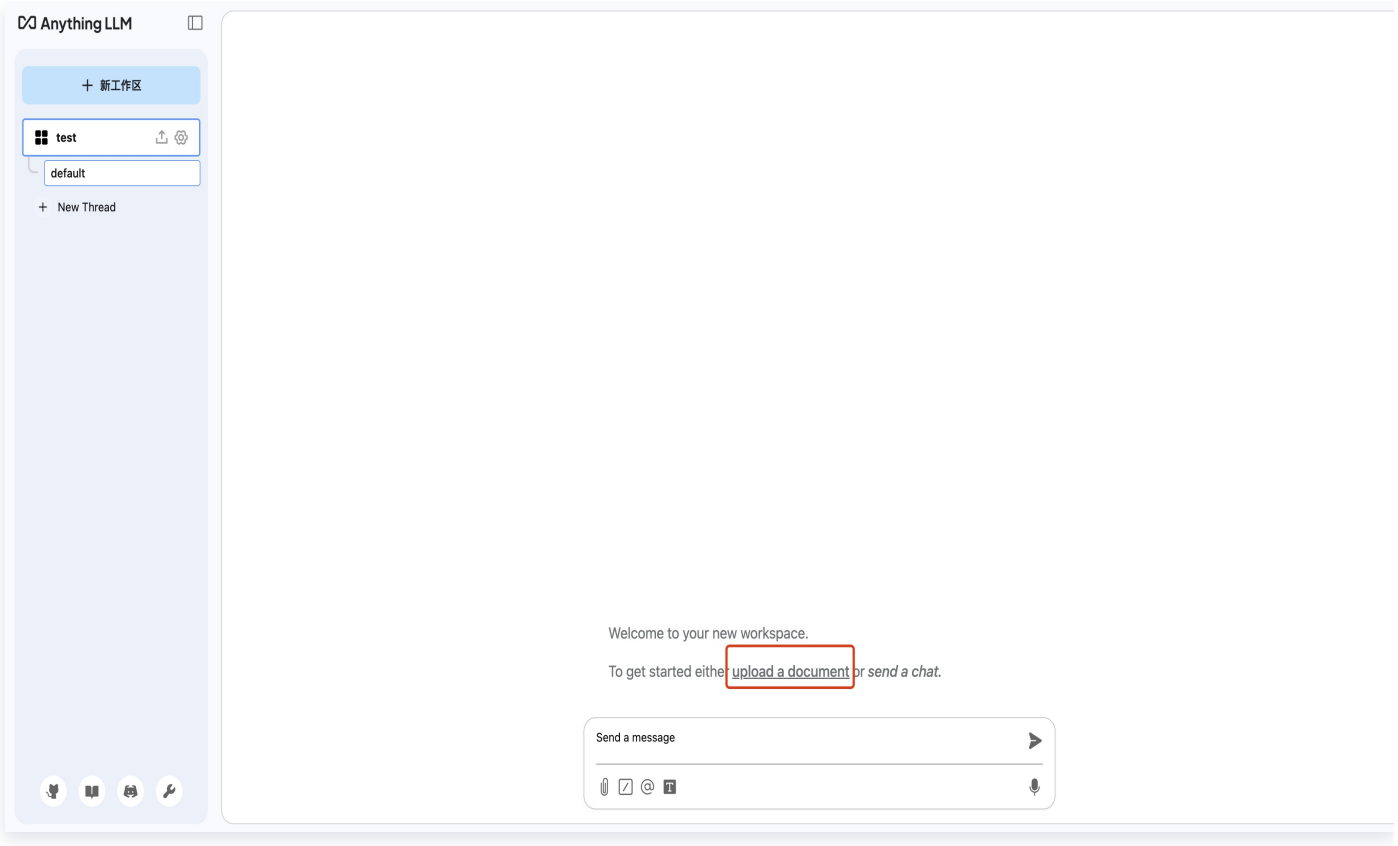
1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **AnythingLLM**。



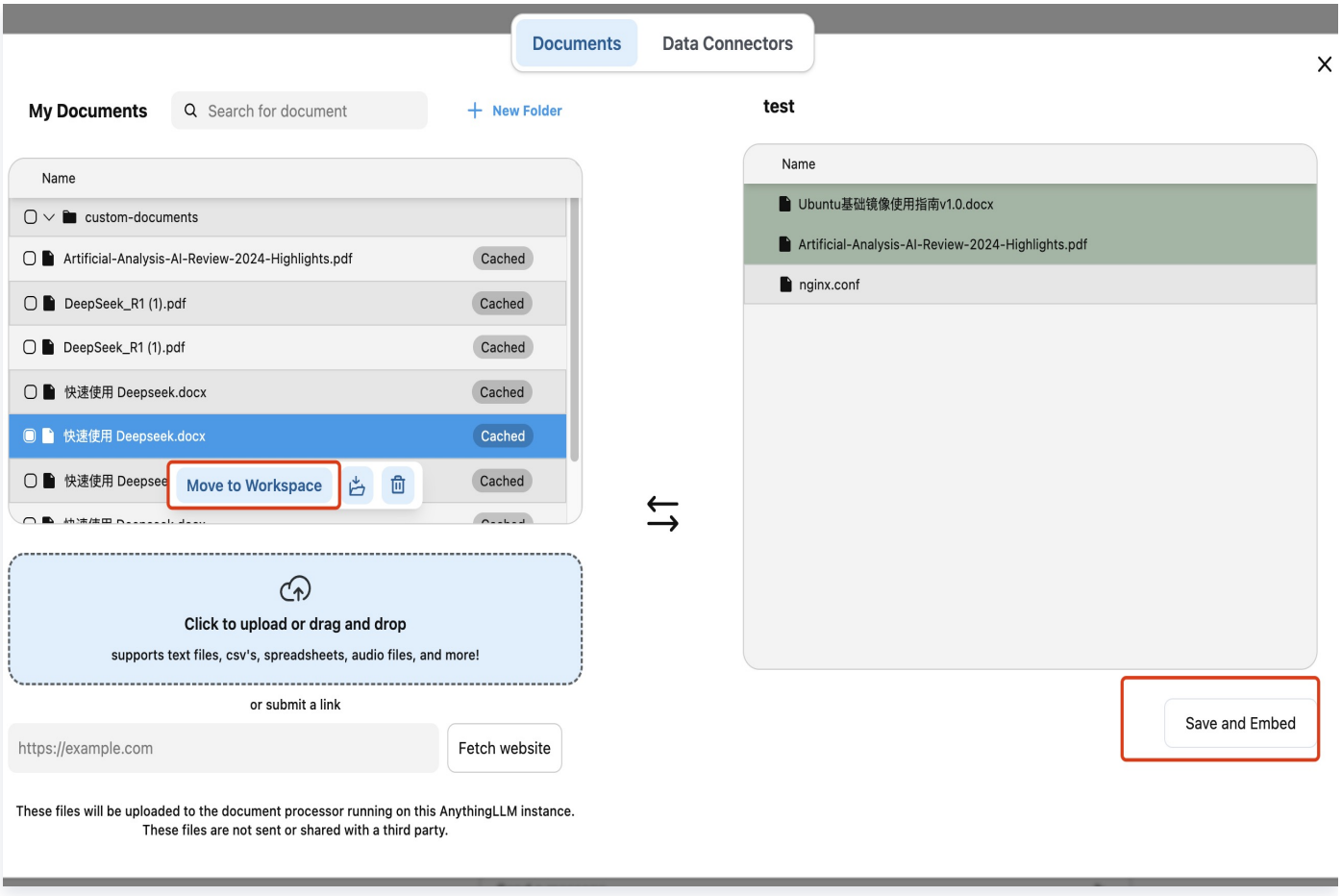
2. 新建窗口后，单击页面左下角**设置**，进入设置页面。单击左侧导航栏 **LLM 首选项**进入配置。
   - 将 **LLM 提供商**选择为 **Ollama**。
   - 将 **Ollama Base URL** 修改为：该台 HAI 实例的公网 IP:6399。
   - 在 **Ollama Model** 处选择需要使用的模型，例如：deepseek-r1:32b。
   - 在 **Ollama Keep Alive** 处按需配置保活时长。（模型在每次超过保活时长后会被移除，再次使用时需重新载入模型，耗时较久，若不存在频繁切换模型诉求，建议将保活时长尽可能调大。）



3. 配置完成后，回到项目页面，单击 **upload a document** 上传本地文件。



4. 上传文件后，选中希望使用的文件，单击 **Move to Workspace** 将文件添加至项目。单击 **Save and Embed**，完成配置。



5. 您可直接与模型进行对话，模型会根据对话内容智能调用本地知识库内容。

通过终端连接命令行使用

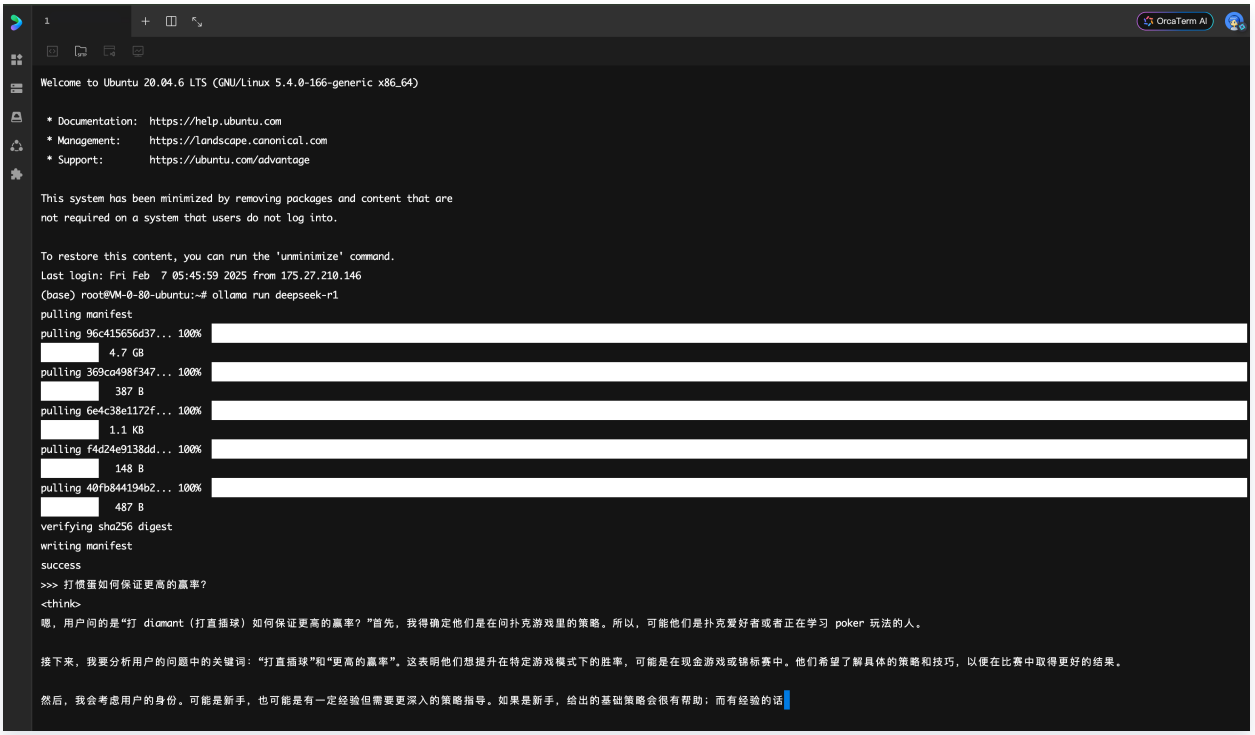
1. 在 高性能应用服务 HAI 控制台，选择**算力连接 > 终端连接(SSH)**。

2. 在弹出的 OrcaTerm 登录页面中，输入站内信中的登录密码，单击**登录**。

3. 登录成功后，输入以下命令加载默认模型：

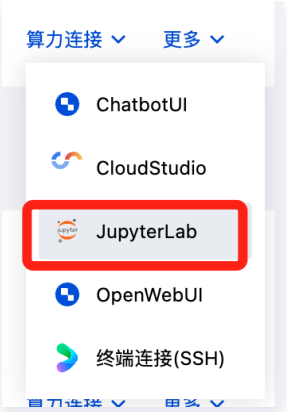
```
ollama run deepseek-r1
```

运行结果如下：

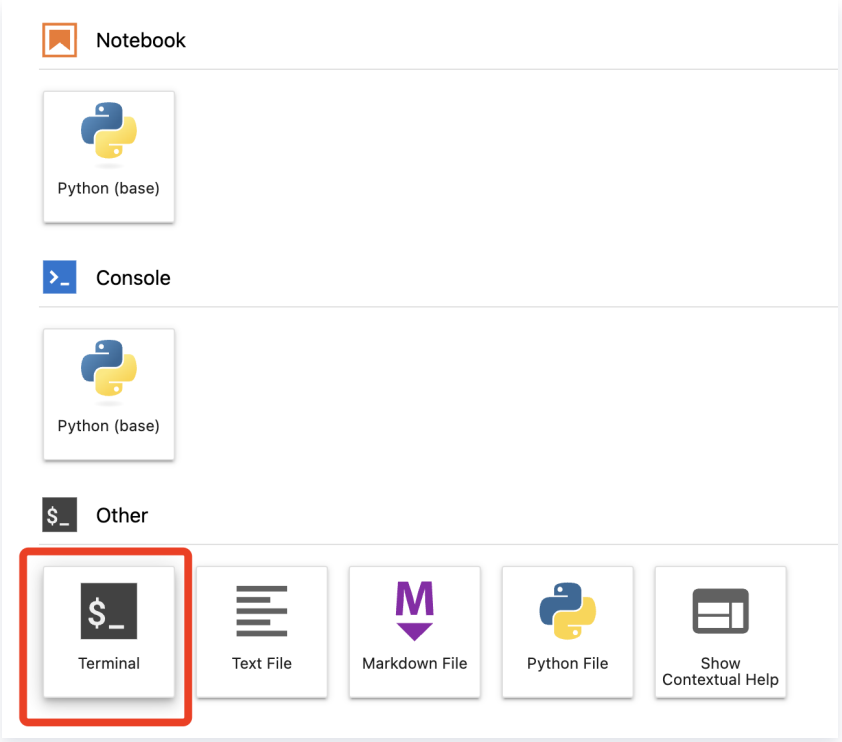


通过 JupyterLab 命令行使用

1. 在 高性能应用服务 HAI 控制台，选择**算力连接 > JupyterLab**。



2. 新建一个 **Terminal**。



3. 输入以下命令加载默认模型：

```
ollama run deepseek-r1
```

运行效果如下：

进阶使用

切换不同参数量级

若默认的模型无法满足需求，可通过以下命令自定义模型参数量级：

- DeepSeek-R1-Distill-1.5B

  ```
  ollama run deepseek-r1:1.5b
  ```

- DeepSeek-R1-Distill-7B

  ```
  ollama run deepseek-r1:7b
  ```

- DeepSeek-R1-Distill-8B

  ```
  ollama run deepseek-r1:8b
  ```

- DeepSeek-R1-Distill-14B

  ```
  ollama run deepseek-r1:14b
  ```

- DeepSeek-R1-Distill-32B

  ```
  ollama run deepseek-r1:32b
  ```

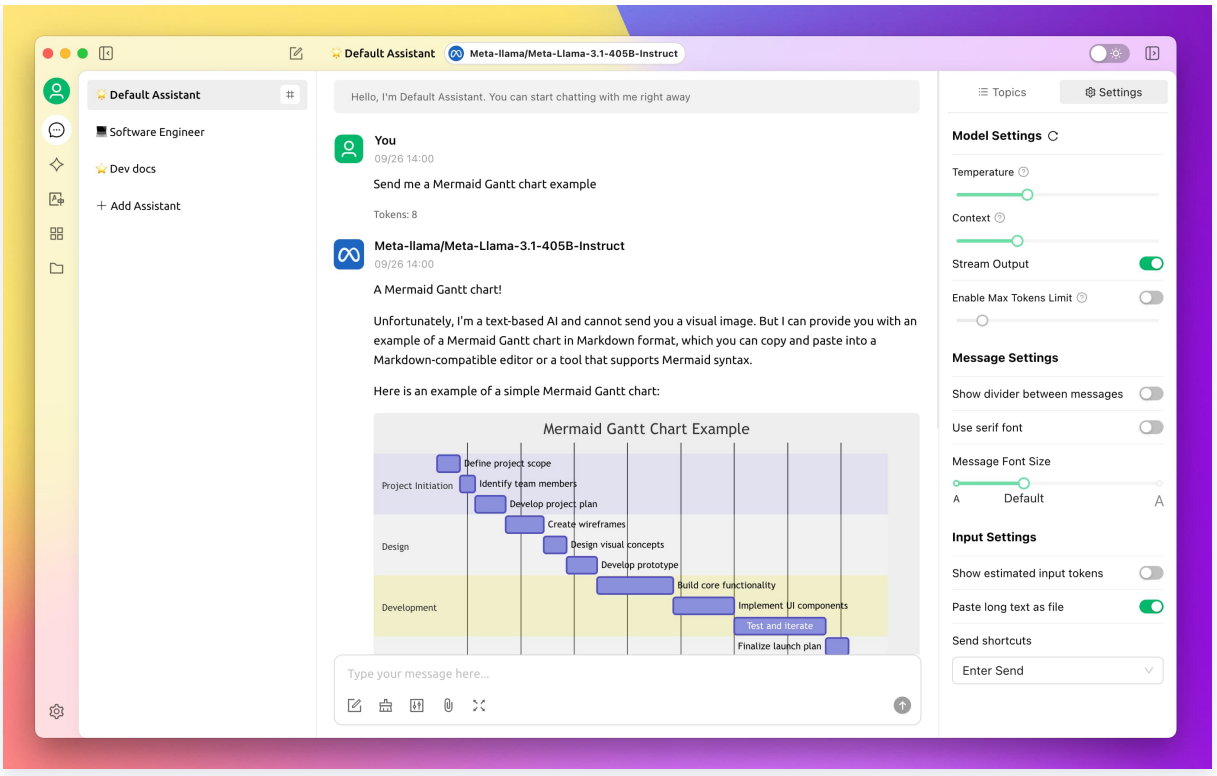
API 调用

实例环境中已预装并启动 Ollama serve，该服务支持通过 REST API 进行调用。您可以参考 Ollama API 文档，以了解具体的调用方式和方法。

场景案例

搭建个人知识库

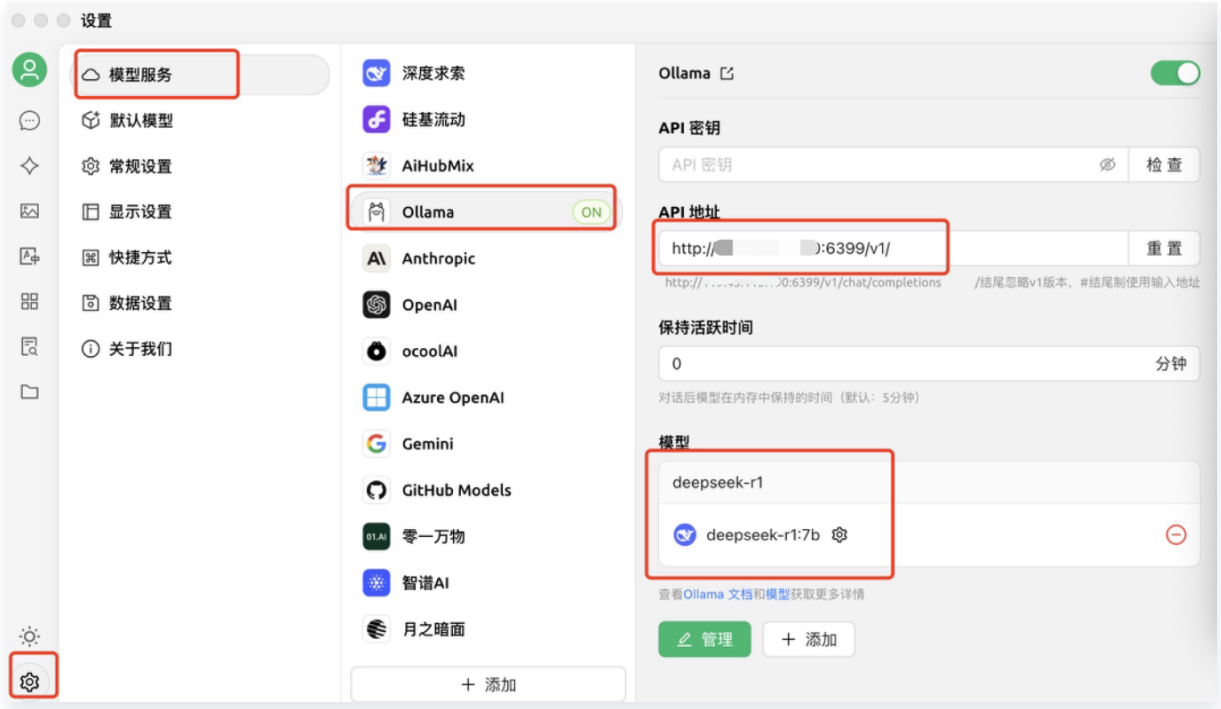
1. 下载 Cherry Studio：一款支持多个大语言模型（LLM）服务商的桌面客户端。



2. **配置 API**：进入设置界面，选择**模型服务**中的 **Ollama**，填写 API 地址及模型名称。

   2.1 API 地址：将默认的 localhost 替换为 HAI 实例的公网 IP，将端口号由11434修改为6399。

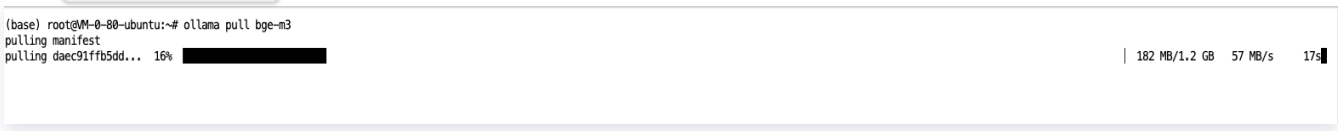
   2.2 单击下方的**添加**按钮添加模型，模型 ID 输入“deepseek-r1:7b”或“deepseek-r1:1.5b”



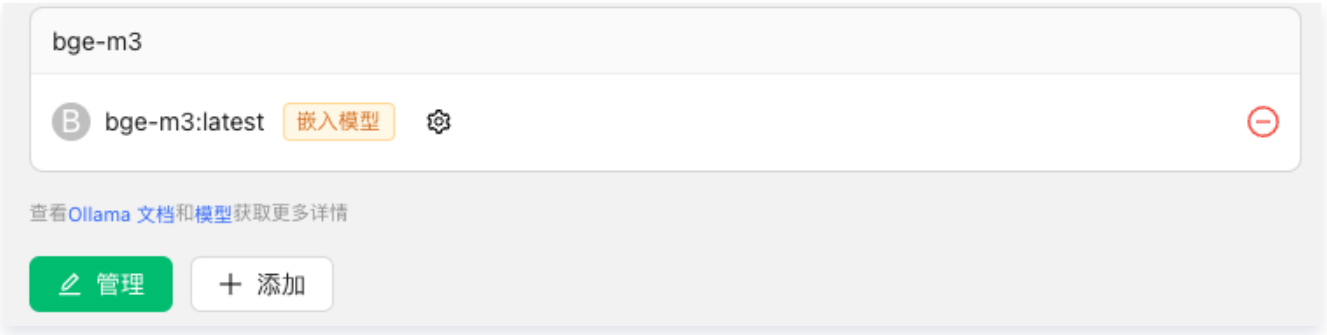
3. **检查连通性**：单击 API 密钥右侧的**检查**，API 密钥不需填写，页面显示“连接成功”即可完成配置。

4. **添加本地知识库文件并使用：**若需使用本地知识库，您可按如下步骤进行配置（以 bge-m3 嵌入模型为例）。

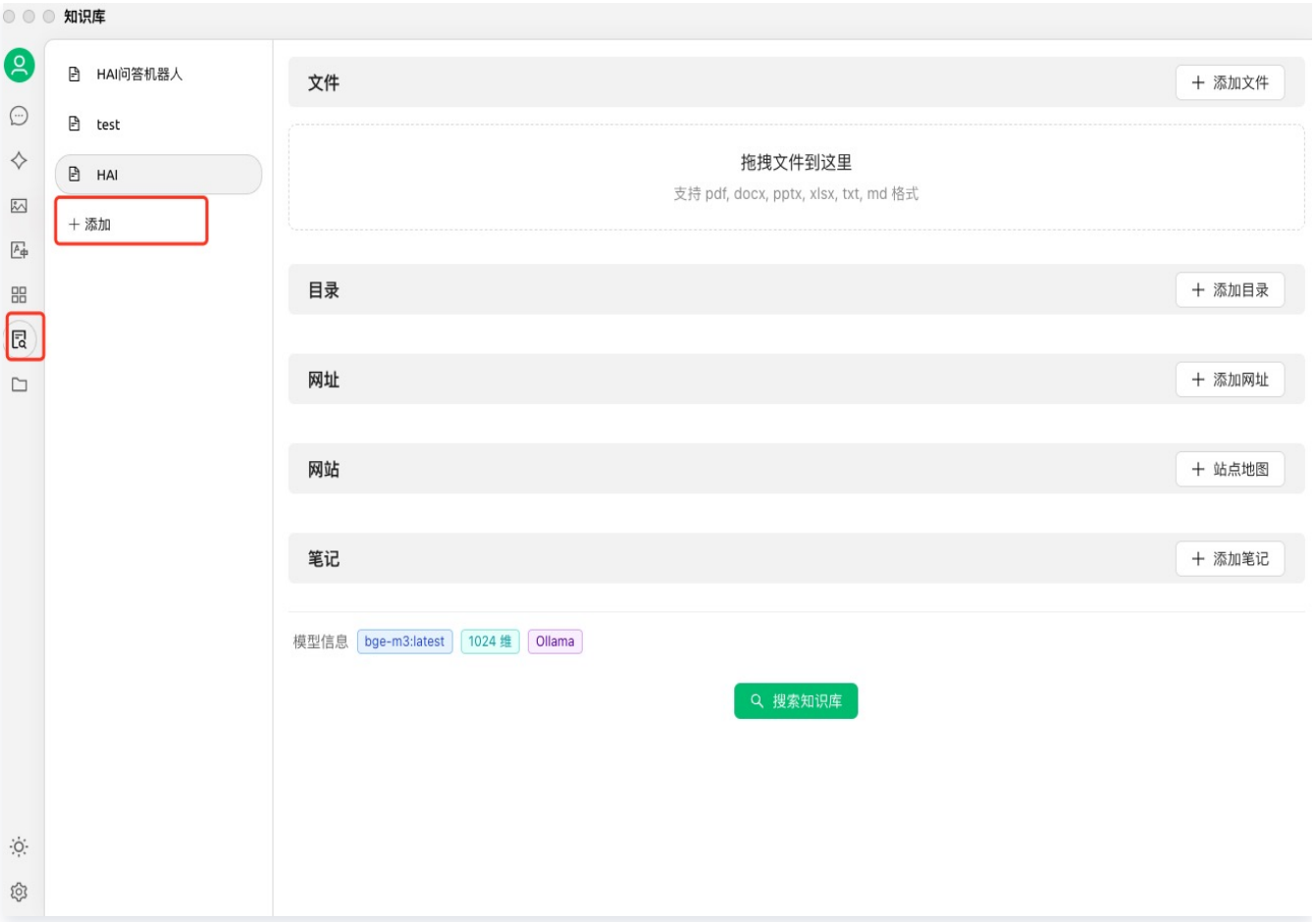
   4.1 下载嵌入模型：单击**算力连接**，选择 **JupyterLab**。进入后打开 terminal，输入 `ollama pull bge-m3` 。

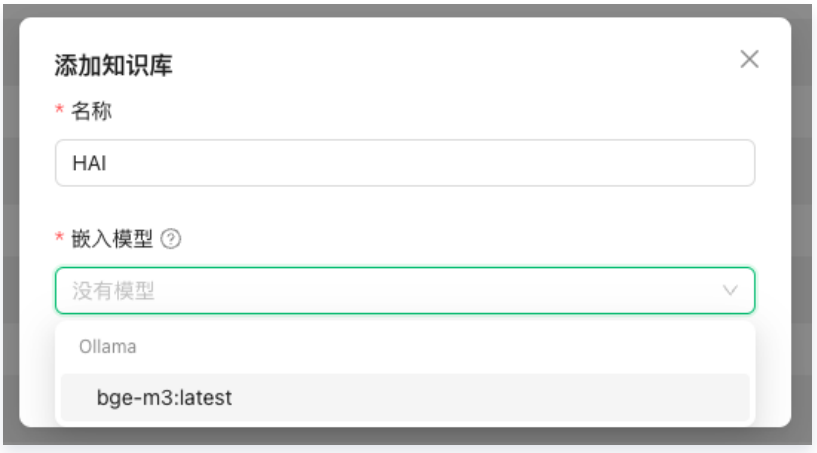


   4.2 添加嵌入模型：下载完成后，返回 cherry studio。进入 ollama 模型服务页面，单击下方**添加**按钮添加模型，模型 ID 输入“bge-m3:latest”。



   4.3 添加知识库：添加完成后，进入“知识库”页面，单击**添加**，嵌入模型选择“bge-m3:latest”，完成后即可上传本地文件，进行知识库管理。


知识库
HAI问答机器人
test
HAI
+ 添加
文件
+ 添加文件
拖拽文件到这里
支持 pdf, docx, pptx, xlsx, txt, md 格式
目录
+ 添加目录
网址
+ 添加网址
网站
+ 站点地图
笔记
+ 添加笔记
模型信息
bge-m3:latest
1024 维
Ollama
搜索知识库


添加知识库
* 名称
HAI
* 嵌入模型
没有模型
Ollama
bge-m3:latest

常见问题

目前支持哪些参数量级的模型?

目前 HAI 已支持1.5B、7B、8B、14B、32B 的 DeepSeek-R1。70B、671B 将在近期推出，欢迎持续关注。

Ollama/API 的端口号是哪个?

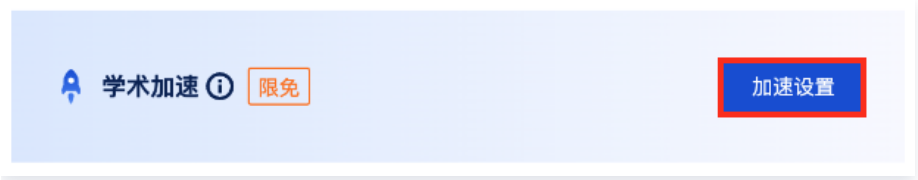
HAI 调用 Ollama 的 API 端口使用 6399，OpenWebUI 端口使用 6699，ChatbotUI 端口使用 6889。其他端口详情请参见 常用端口 。

如何通过 API 使用模型?

实例环境中已预装并启动 Ollama serve，该服务支持通过 REST API 进行调用。您可以参考 Ollama API 文档，以了解具体的调用方式和方法。

中国大陆地域通过 Ollama 下载模型速度慢怎么办?

目前北京、上海、广州的资源，可通过 高性能应用服务 HAI 控制台 单击**加速设置**，开启学术加速后，提高资源访问速度。相关能力介绍可参考 开启学术加速 。



提示资源紧张，排队人数过多，如何处理?

由于 DeepSeek 使用火热，部分地域可能出现售罄情况，无法成功创建实例。已付款项将会原路退回。建议更换地域重新购买或稍后重试。

社区交流

如有使用问题，欢迎加入腾讯云 DeepSeek 部署交流群。我们期待您的建议与反馈。

快速使用 TACO 加速版 DeepSeek-R1 32B

最近更新时间：2025-06-26 14:57:42

背景介绍

DeepSeek-R1 在后训练阶段大规模使用了强化学习技术，在仅有极少标注数据的情况下，极大提升了模型推理能力。在数学、代码、自然语言推理等任务上，性能比肩 OpenAI o1 正式版。
DeepSeek-R1 32B TACO 加速版环境预装**腾讯云自研 TACO 推理加速框架**及 DeepSeek-R1 32B 模型，大幅提升推理性能，实测在多个场景下较 vLLM 有80%的性能提升。

> ⚠ **注意：**
> DeepSeek-R1 32B TACO 加速版环境目前仅对白名单客户开放，若您有体验需求，可在本文档下方填写问卷提交试用申请。

快速使用

步骤一：创建 DeepSeek-R1 32B TACO 加速版应用

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。
   - **选择应用：**选择社区应用，应用选择 **DeepSeek-R1 32B TACO 加速版**。
   - **地域：**建议选择靠近自己实际地理位置的地域，降低网络延迟、提高您的访问速度。
   - **算力方案：**选择合适的算力套餐。
   - **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
   - **购买数量：**默认1台。
3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页面。同时您将在站内信中收到登录密码。此时，可通过可视化界面（GUI）使用 DeepSeek 模型。

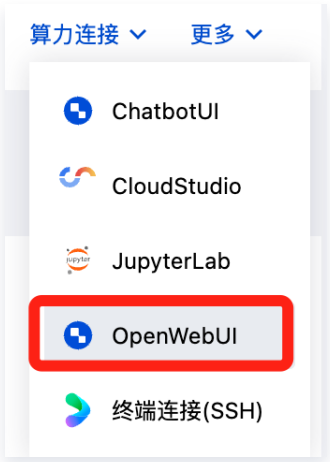
6. 您可以在此页面查看详细的配置信息，到此为止，说明您的应用实例购买成功。

步骤二：使用 DeepSeek-R1 模型

等待几分钟创建完成后，将在站内信中收到登录密码。此时，可通过可视化界面（GUI）使用 DeepSeek 模型。

通过 OpenWebUI 可视化界面使用（推荐）

1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **OpenWebUI**。

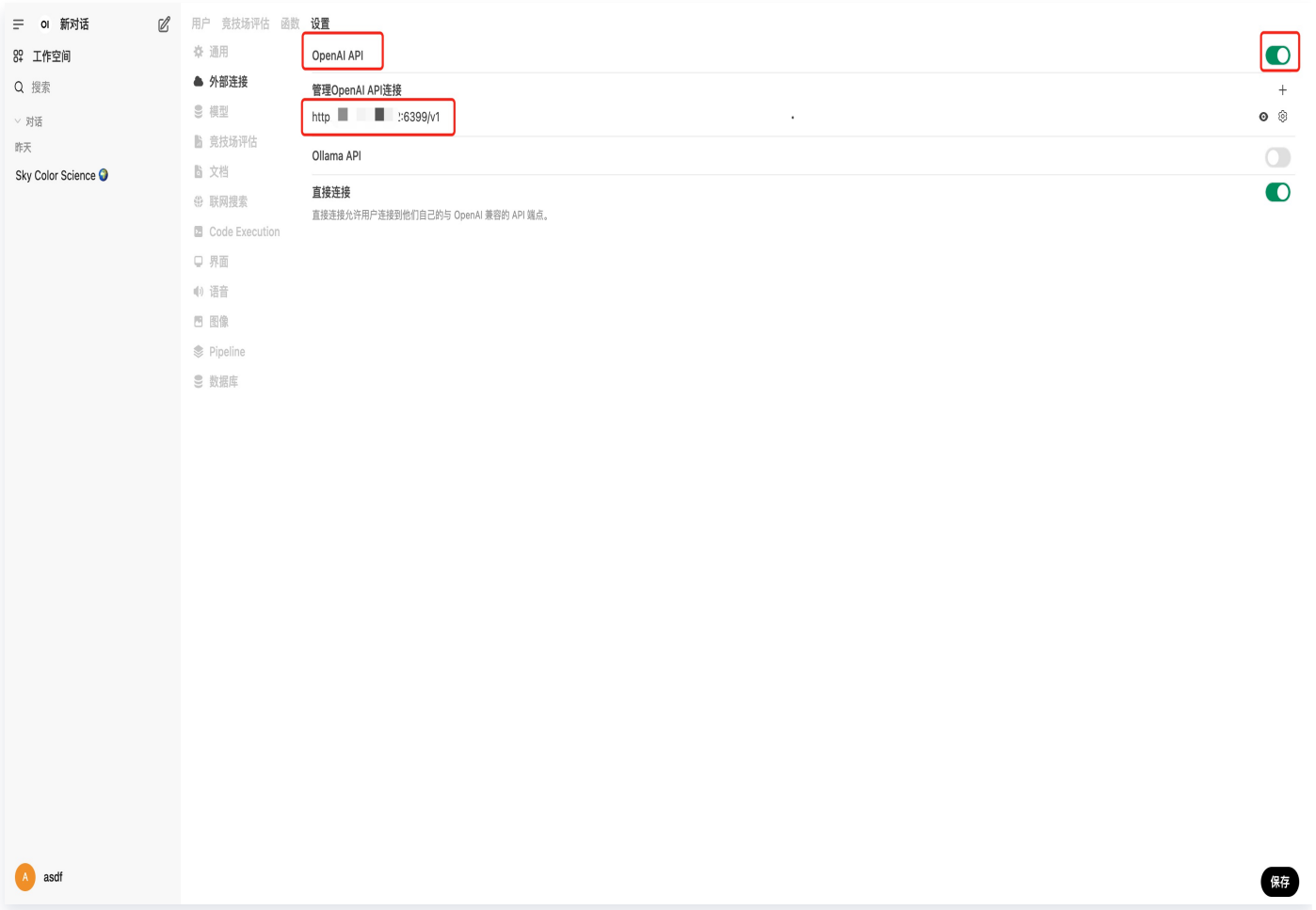


2. 在新窗口中，单击**开始使用**。

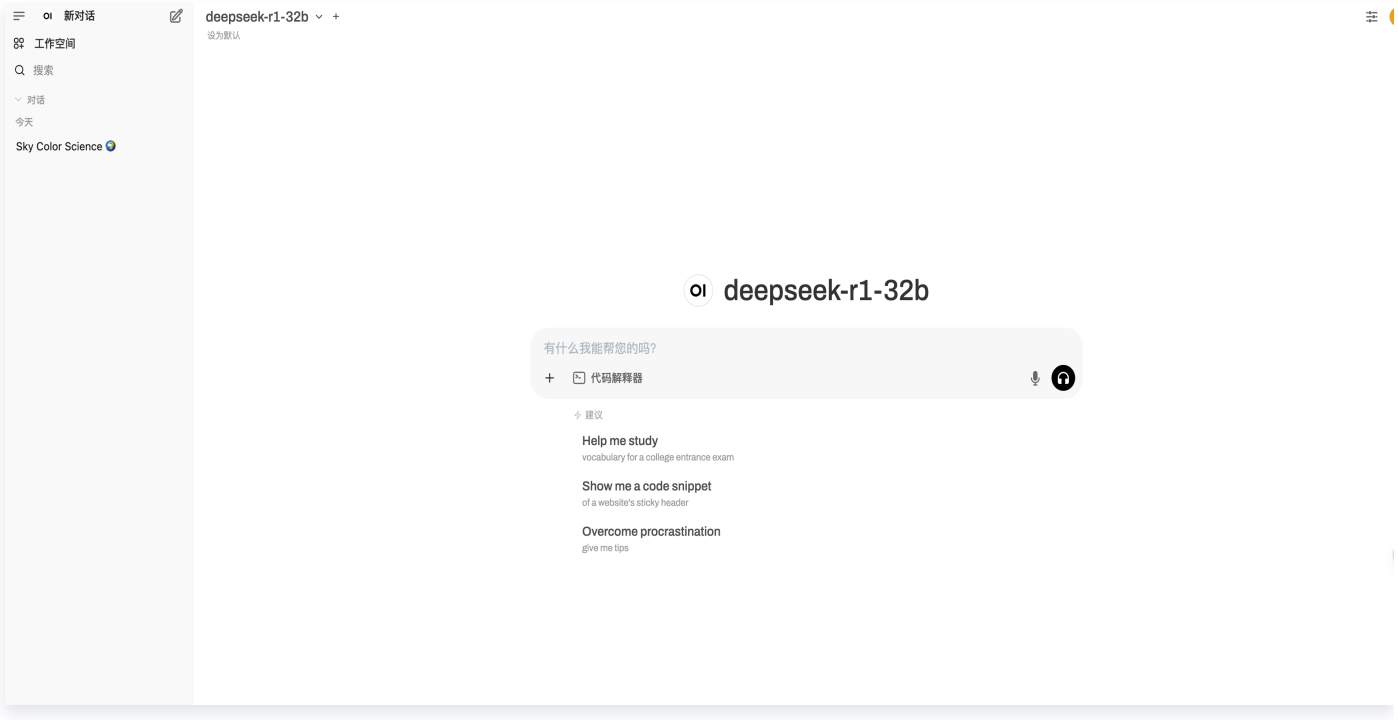
3. 自定义**名称**、**电子邮箱**、**密码**，创建管理员账号。

4. 配置 OpenWebUI：单击页面左下角**设置**-**管理员设置**-**外部连接**。

将 OpenAI API 连接修改为该台 HAI 实例的公网 IP:6399/v1。

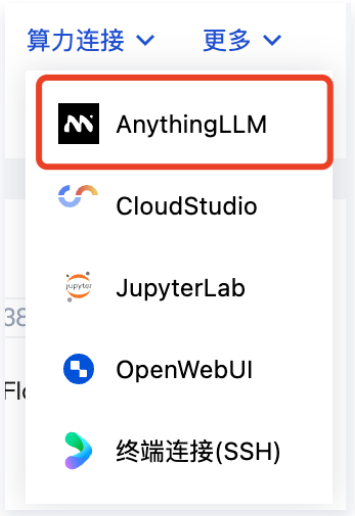


5. 配置完成后，即可返回聊天页面进行对话。

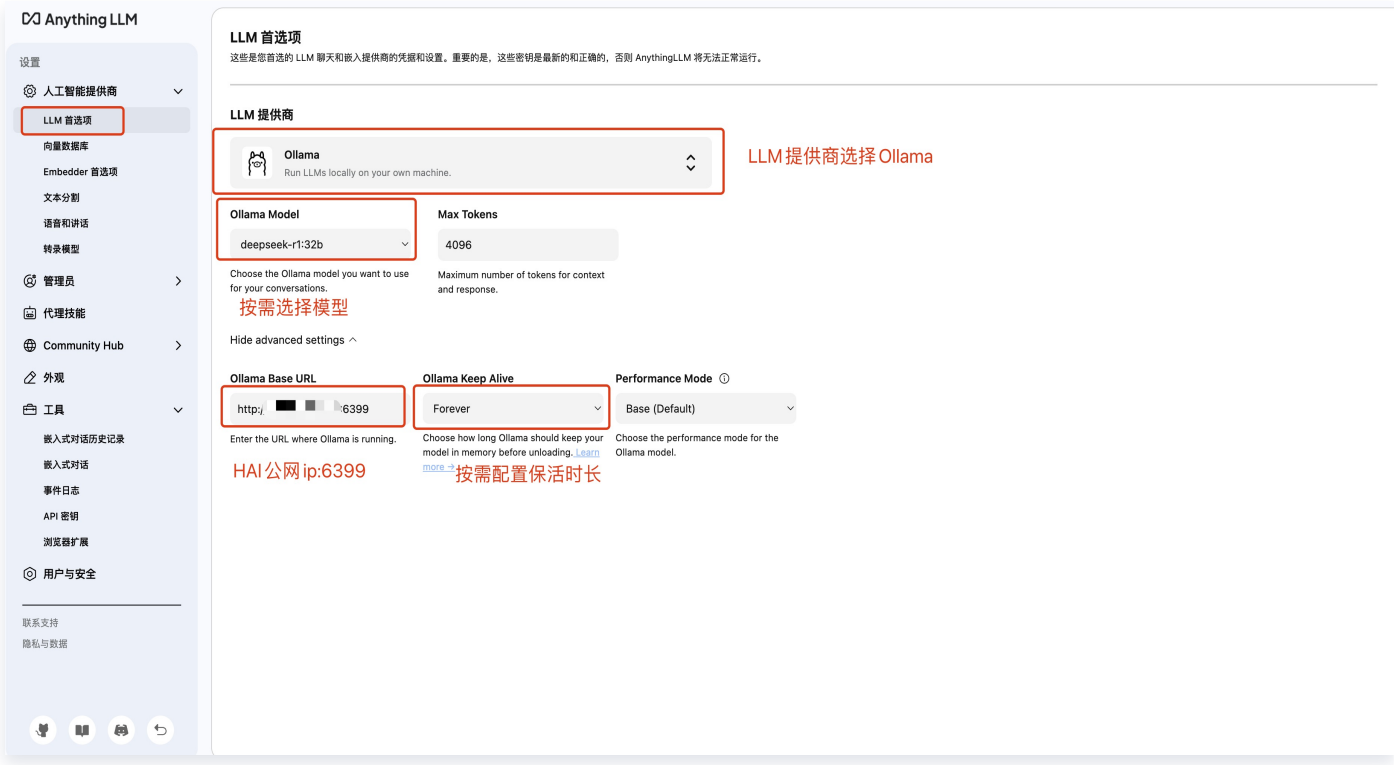


通过 AnythingLLM 可视化界面使用

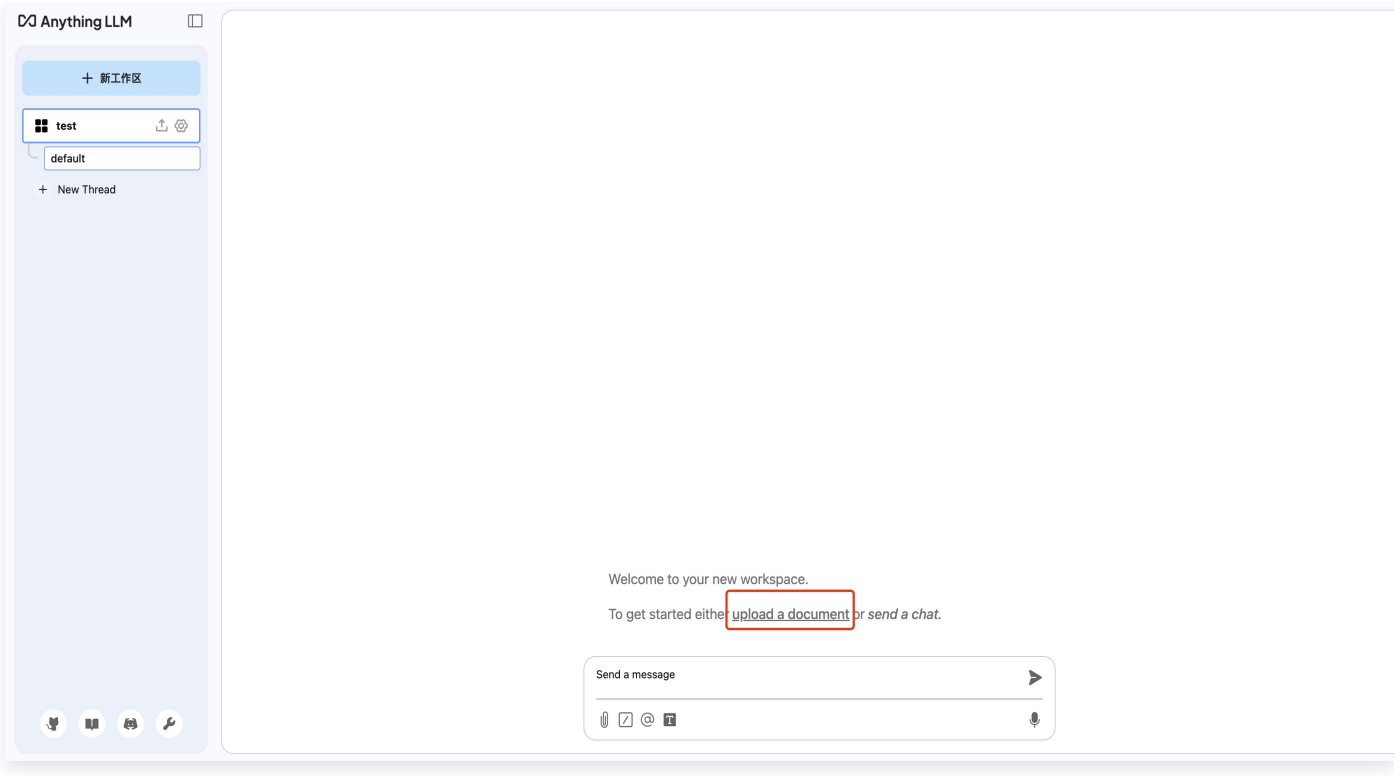
1. 登录 高性能应用服务 HAI 控制台，选择**算力连接** > **AnythingLLM**。



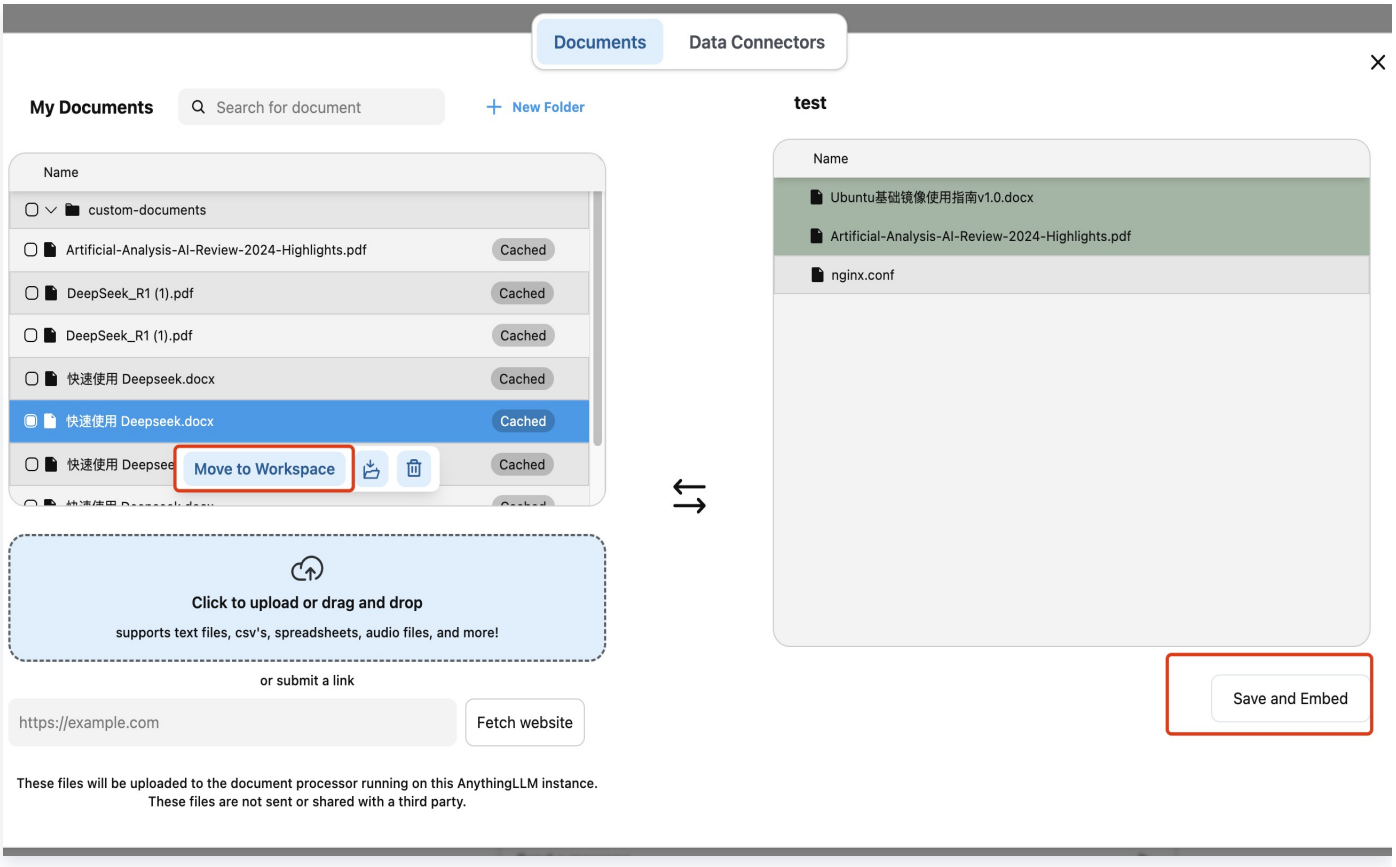
2. 新建窗口后，单击页面左下角**设置**，进入设置页面。单击左侧导航栏 **LLM 首选项**进入配置。
   - 将 **LLM 提供商**选择为 **Ollama**。
   - 将 **Ollama Base URL** 修改为：该台 HAI 实例的公网 IP:6399。
   - 在 **Ollama Model** 处选择需要使用的模型，例如：deepseek-r1:32b。
   - 在 **Ollama Keep Alive** 处按需配置保活时长。（模型在每次超过保活时长后会被移除，再次使用时需重新载入模型，耗时较久，若不存在频繁切换模型诉求，建议将保活时长尽可能调大。）



3. 配置完成后，回到项目页面，单击 **upload a document** 上传本地文件。



4. 上传文件后，选中希望使用的文件，单击 **Move to Workspace** 将文件添加至项目。单击 **Save and Embed**，完成配置。



5. 您可直接与模型进行对话，模型会根据对话内容智能调用本地知识库内容。

API 调用

该环境 API 兼容 OpenAI 调用规范，在实例启动后，您可使用下述调用方式对模型进行调用。

- 将 `ip` 修改为实例公网 IP，`port` 修改为6399。
- 将 `"Bearer $OPENAI_API_KEY"` 替换为任意字符。

```
curl http://<ip>:port/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "deepseek-r1-32b",
    "messages": [{"role": "user", "content": "为什么天空是蓝色的"}],
    "temperature": 0.7
  }'
```

开启申请

您可以填写 高性能应用服务 HAI-DeepSeek-R1 32B TACO 加速版体验申请 问卷，提交试用申请。

快速构建 Stable Diffusion 文生图 API 服务

最近更新时间：2025-06-26 14:57:42

本次我们使用 腾讯云高性能应用服务 HAI 体验快速搭建并使用 AI 模型 StableDiffusion，实现思路如下：

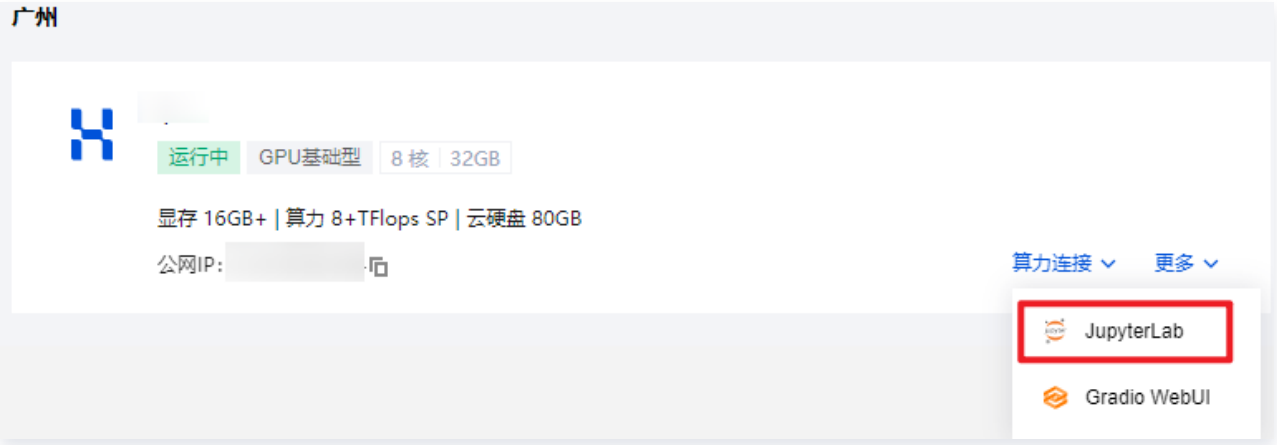
- 提前通过高性能应用服务 HAI 部署成功 StableDiffusion 应用。
- 基于部署好的应用，利用体验 JupyterLab 进行 StableDiffusion API 的部署。

前提

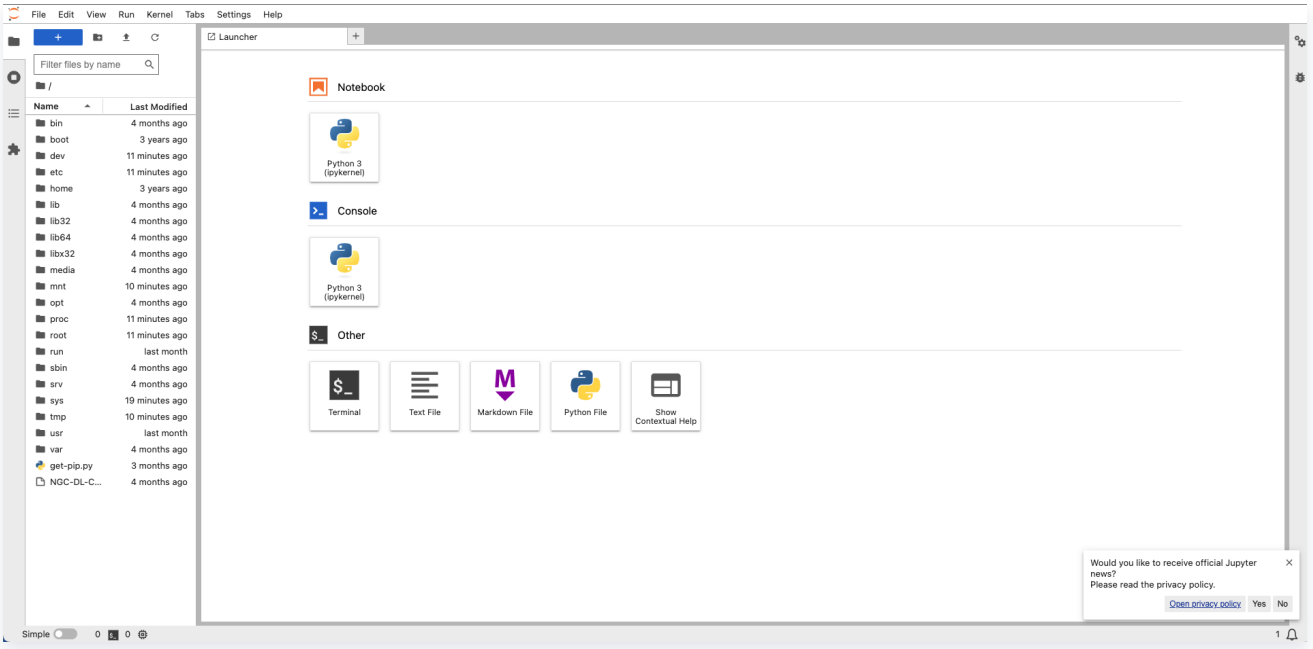
在部署 API 服务之前，请确保您已成功部署 StableDiffusion 应用。详细步骤可参见 快速使用 Stable Diffusion 文生图应用 。

部署 API 服务

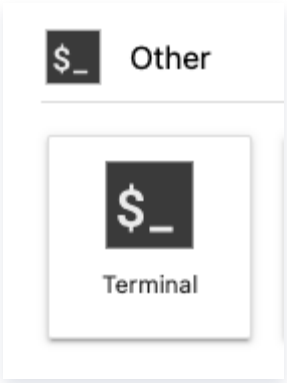
1. 进入 jupyter_lab 控制台 操作界面。
   1.1 在实例列表中选择**更多** > **JupyterLab**。



   1.2 初步认识并操作 JupyterLab。

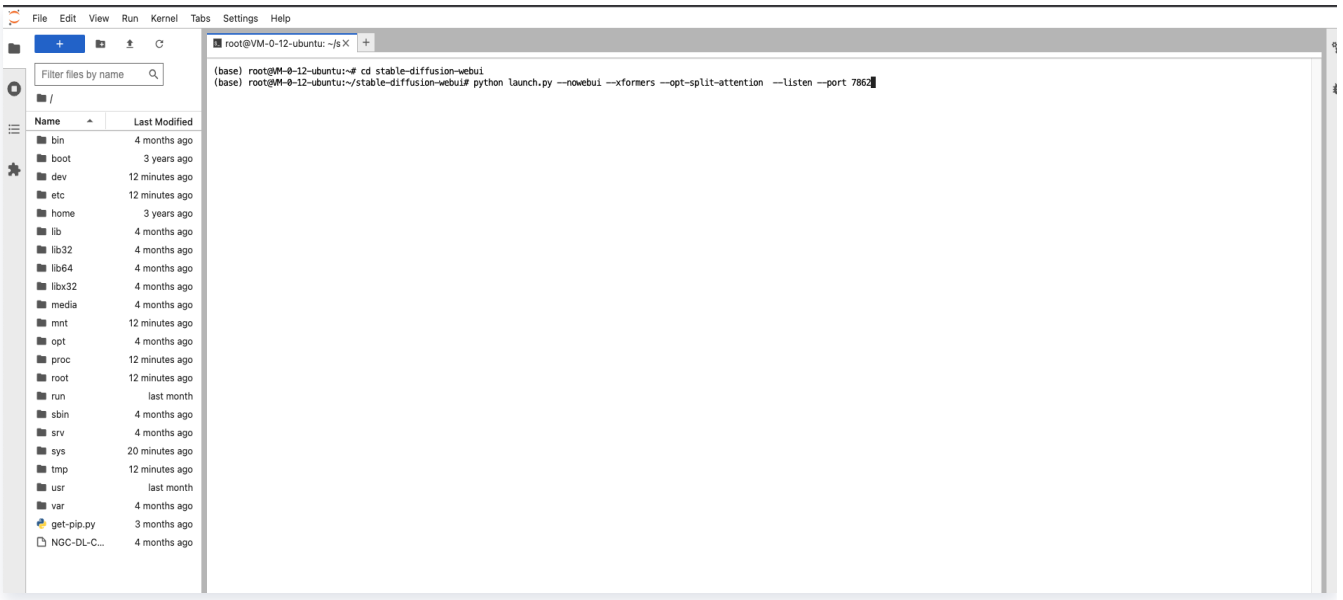


1.3 选择使用终端命令行操作。



输入代码：

```
cd stable-diffusion-webui
python launch.py --nowebui --xformers --opt-split-attention  --
listen --port 7862
```



命令参数描述如下图：

命令	描述
--nowebui	以 API 模式启动。
--xformers	改善内存消耗和速度。
--opt-split-attention	Cross attention layer optimization 优化显着减少了内存使用。
--listen	默认启动绑定的 IP 是 127.0.0.1。
--port	默认端口是7860，可以配置并修改该参数，例如：--port 7862。
--gradio-auth username:password	如果希望给 WebUI 设置登录密码，可以配置该参数，例如：--gradio-auth GitLqr:123456。

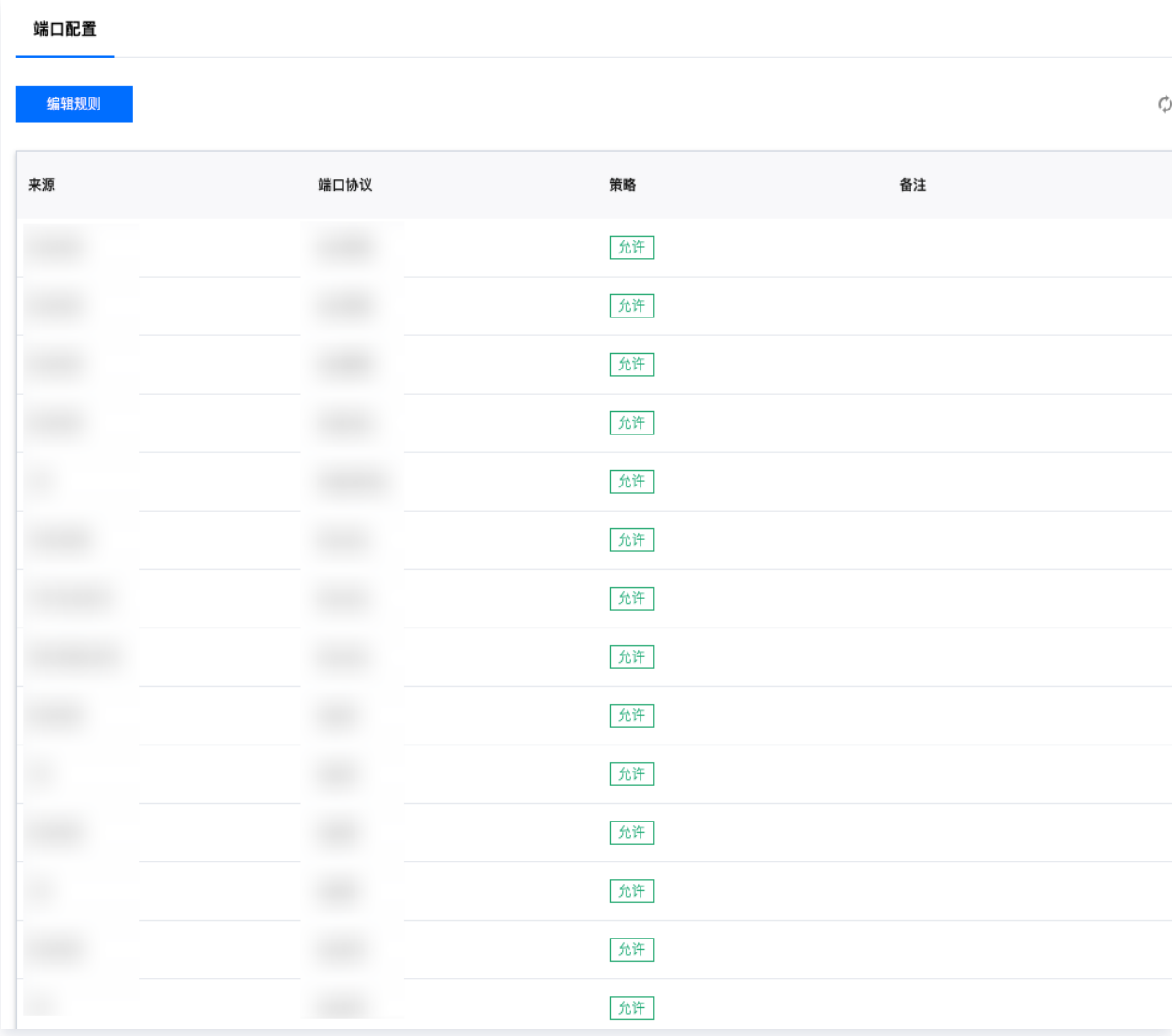
操作截图如下图所示：

```
(base) root@VM-0-12-ubuntu:~/stable-diffusion-webui# python launch.py --nowebui --xformers --opt-split-attention  --listen --port 7862
Python 3.8.10 (default, Jun  4 2021, 15:09:15)
[GCC 7.5.0]
Version: v1.5.2
Commit hash: c9c8485bc1e8720aba70f029d25cba1c4abf2b5c
Installing requirements
If submitting an issue on github, please provide the full startup log for debugging purposes.

Initializing Dreambooth
Dreambooth revision: cf086c536b141fc522ff11f6cffc8b7b12da04b9
Successfully installed accelerate-0.21.0 fastapi-0.94.1 gitpython-3.1.40 transformers-4.30.2

Does your project take forever to startup?
Repetitive dependency installation may be the reason.
Automatic1111's base project sets strict requirements on outdated dependencies.
If an extension is using a newer version, the dependency is uninstalled and reinstalled twice every startup.

[+] xformers version 0.0.17 installed.
[+] torch version 2.0.1 installed.
[+] torchvision version 0.15.2 installed.
[+] accelerate version 0.21.0 installed.
[+] diffusers version 0.19.3 installed.
[+] transformers version 4.30.2 installed.
[+] bitsandbytes version 0.35.4 installed.


Launching API server with arguments: --nowebui --xformers --opt-split-attention --listen --port 7862
[2023-11-01 06:34:58,839][DEBUG][git.cmd] - Popen(['git', 'version'], cwd=/root/stable-diffusion-webui, stdin=None, shell=False, universal_newlines=False)
[2023-11-01 06:34:58,867][DEBUG][git.cmd] - Popen(['git', 'version'], cwd=/root/stable-diffusion-webui, stdin=None, shell=False, universal_newlines=False)
================================================================================
You are running xformers 0.0.17.
The program is tested to work with xformers 0.0.20.
To reinstall the desired version, run with commandline flag --reinstall-xformers.

Use --skip-version-check commandline argument to disable this check.
================================================================================
2023-11-01 06:34:59,976 - ControlNet - INFO - ControlNet v1.1.410
ControlNet preprocessor location: /root/stable-diffusion-webui/extensions/sd-webui-controlnet/annotator/downloads
2023-11-01 06:35:00,089 - ControlNet - INFO - ControlNet v1.1.410
Loading weights [6ce0161689] from /root/stable-diffusion-webui/models/Stable-diffusion/v1-5-pruned-emaonly.safetensors
Creating model from config: /root/stable-diffusion-webui/configs/v1-inference.yaml
LatentDiffusion: Running in eps-prediction mode
DiffusionWrapper has 859.52 M params.
Model loaded in 2.6s (load weights from disk: 0.2s, create model: 0.5s, apply weights to model: 0.5s, apply half(): 0.3s, move model to device: 0.6s, calculate empty prompt: 0.4s).
[2023-11-01 06:35:03,462][DEBUG][api.py] - SD-Webui API layer loaded
Applying attention optimization: xformers... done.
[2023-11-01 06:35:03,818][DEBUG][api.py] - Loading Dreambooth API Endpoints.
Startup time: 48.4s (launcher: 37.5s, import torch: 2.5s, import gradio: 0.8s, setup paths: 1.2s, other imports: 1.9s, load scripts: 4.4s).
INFO:     Started server process [220]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:7862 (Press CTRL+C to quit)
```

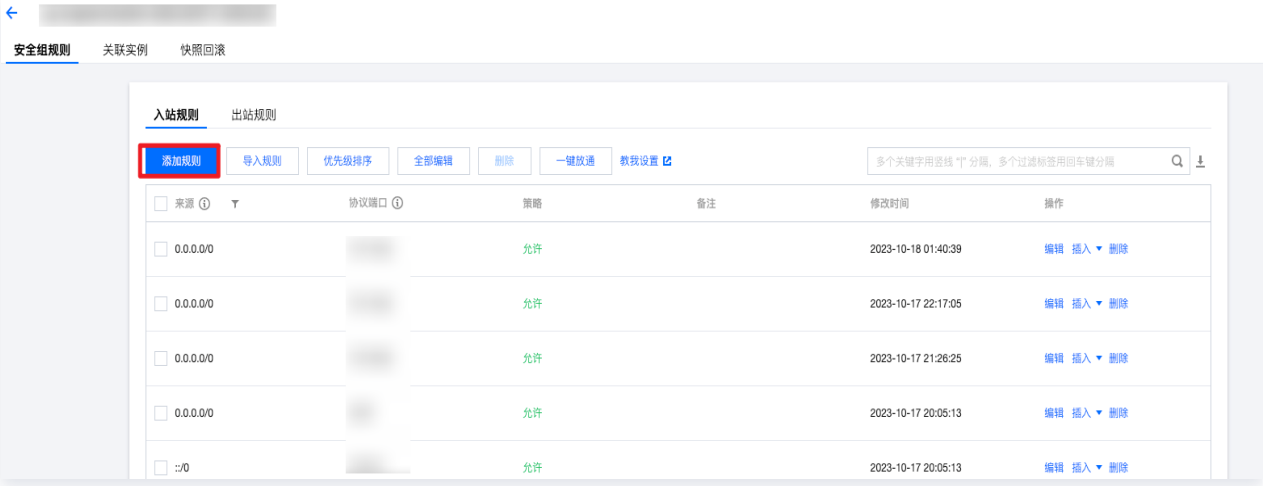
1.4 添加高性能应用服务 HAI 的端口配置，使外部网络能够顺利地访问该服务器提供的 API 服务。

1.4.1 在算力管理页面。单击实例空白进入详情设置页。

1.4.2 在端口配置弹窗中，单击**编辑规则**。



1.4.3 在**安全组规则**页面中，在**入站规则**页签单击**添加规则**。



配置参考如下：

来源： 0.0.0.0/0

协议端口：TCP:7862 （根据您配置的端口填写）

2. 启动 StableDiffusion API 接口使用指南

2.1 配置完成后，在浏览器地址栏输入服务器 IP 地址:端口号/docs 可查看相关的 API 接口使用指南。

官方提供的常用 API 如下：

```
/sdapi/v1/txt2img文字生图 POST
/sdapi/v1/img2img图片生图 POST
/sdapi/v1/options获取设置 GET | 更新设置 POST（可用来更新远端的模型）
/sdapi/v1/sd-models获取所有的模型 GET
```

2.2 查看相关接口示例（`/sdapi/v1/txt2img`）。

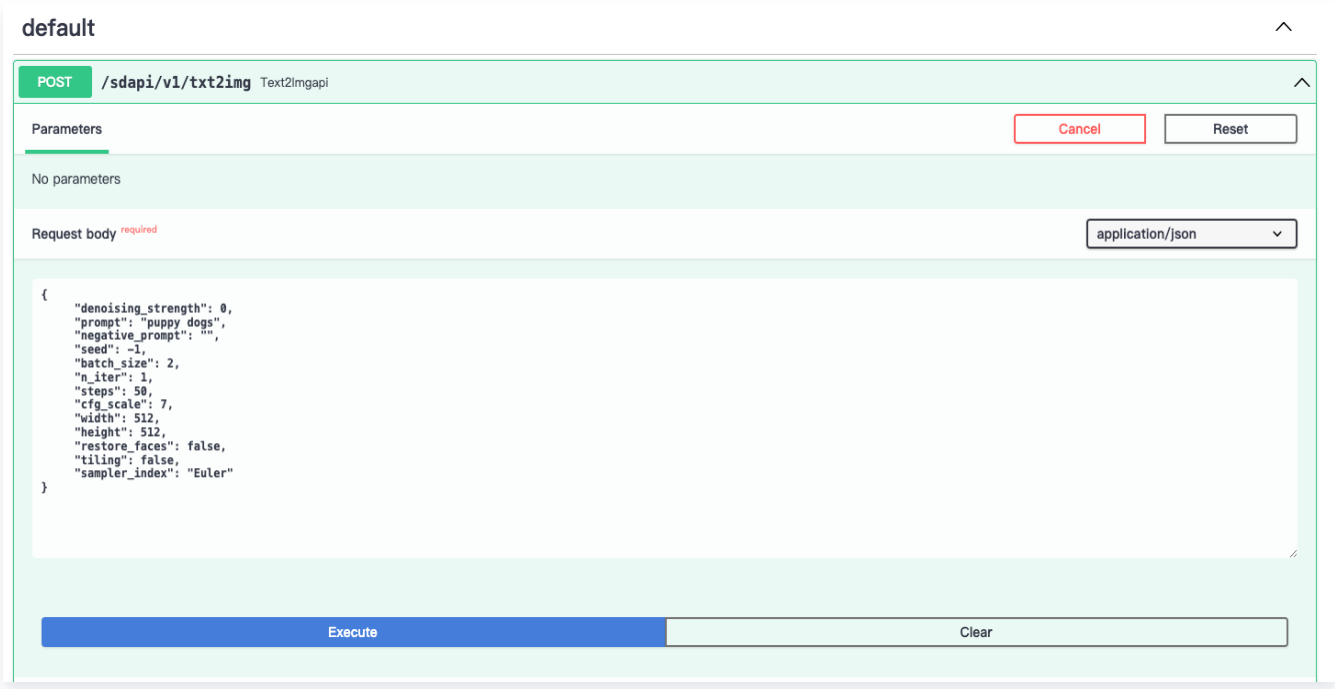
常用输入如下：

```
{
    "denoising_strength": 0,
    "prompt": "puppy dogs",
    "negative_prompt": "",
    "seed": -1,
    "batch_size": 2,
    "n_iter": 1,
    "steps": 50,
    "cfg_scale": 7,
    "width": 512,
    "height": 512,
    "restore_faces": false,
    "tiling": false,
```

```
    "sampler_index": "Euler"
}
```

可复制以上参数到 Request body 中。

名称	说明
prompt	提示词
negative_prompt	反向提示词
seed	种子，随机数
batch_size	每次张数
n_iter	生成批次
steps	生成步数
cfg_scale	关键词相关性
width	宽度
height	高度
restore_faces	脸部修复
tiling	可平铺
sampler_index	采样方法



返回的格式如下：

```
{
    "images": [...], // 这里是一个base64格式的字符串数组，根据请求的图片数量而定
    "parameters": {  ... },//此处为输入的body
    "info": "{...}"// 返回的图片的信息
}
```

当看到类似上图的消息时，说明已经成功与远端的服务器进行连接！如果希望验证结果的图片的实际展示效果，可以复制 images 中的其中一张图片的 base64 格式的字符串，到相关的网站下转换为 jpg 格式。

3. 使用 Python 向高性能应用服务 HAI 提供的 StableDiffusionAPI 发送请求。

以下演示如何使用 Python 向 StableDiffusion API 发出请求。向应用程序的 txt2img（即**文本到图像**）API 发送 POST 请求以简单地生成图像。

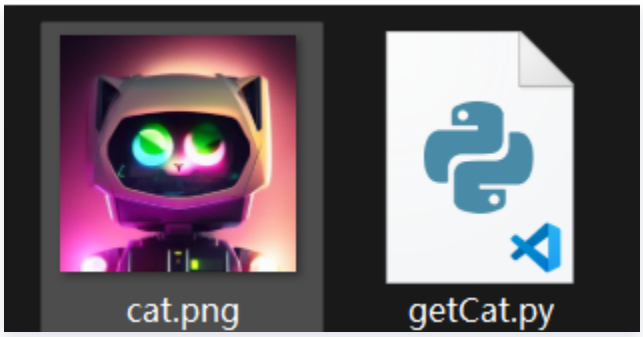
我们将使用 requests 包，如果您还没有安装，请使用安装脚本：

```
pip install requests
```

我们可以发送一个包含提示的请求作为一个简单的字符串。服务器将返回一个图像作为 base64 编码的 PNG 文件，我们需要对其进行解码。 要解码 base64 图像，我们只需使用 base64.b64decode(b64_image)。以下使用 Python 作为脚本代码测试：

```
import json
import base64
import requests

your_ip = '0.0.0.0' # HAI服务器IP地址
your_port =7862  # SD api 监听的端口

def submit_post(url: str,data: dict):
  """
  Submit a POST request to the given URL withthe given data.
  """
  return requests.post(url,data=json.dumps(data))

def save_encoded_image(b64_image: str,output_path: str):
  """
  Save the given image to the given outputpath.
  """
  with open(output_path,"wb") as image_file:
      image_file.write(base64.b64decode(b64_image))

if __name__ == '__main__':
  #/sdapi/v1/txt2img
  txt2img_url = f'http://{your_ip}:{your_port}/sdapi/v1/txt2img'
  data = {
     'prompt': 'a pretty cat,cyberpunk art,kerem beyit,verycute robot
zen,Playful,Independent,beeple |',

'negative_prompt':'(deformed,distorted,disfigured:1.0),poorlydrawn,bad
anatomy,wrong anatomy,extra limb,missing limb,floating limbs,
(mutatedhands and
fingers:1.5),disconnectedlimbs,mutation,mutated,ugly,disgusting,blurry
,amputation,flowers,human,man,woman',
     'Steps':50,
     'Seed':1791574510
  }
  response = submit_post(txt2img_url,data)
  save_encoded_image(response.json()['images'][0],'cat.png')
```

请记住，您的结果会与上述示例有所不同。 如果遇到问题，请仔细检查运行 StableDiffusionAPI 应用程序的终端的输出。 如果您遇到404 Not Found 的问题，请仔细检查 URL 是否输入正确并指向正确的地址（例如 127.0.0.1）。



服务端可查看每一次接口调用详情：

```
100%|██████████| 50/50 [00:07<00:00,  7.00it/s]
[2023-09-24 04:17:42,047][INFO][modules.shared] - Ending job scripts_txt2img (7.38 seconds)
INFO:     [redacted]:4207 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:18:32,754][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.01it/s]
[2023-09-24 04:18:40,121][INFO][modules.shared] - Ending job scripts_txt2img (7.37 seconds)
INFO:     [redacted]:4240 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:18:51,476][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  6.99it/s]
[2023-09-24 04:18:58,864][INFO][modules.shared] - Ending job scripts_txt2img (7.39 seconds)
INFO:     [redacted]:4252 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:18:59,581][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.04it/s]
[2023-09-24 04:19:06,918][INFO][modules.shared] - Ending job scripts_txt2img (7.34 seconds)
INFO:     [redacted]:4258 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:19:19,226][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  6.93it/s]
[2023-09-24 04:19:26,679][INFO][modules.shared] - Ending job scripts_txt2img (7.45 seconds)
INFO:     [redacted]:4267 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:21:08,028][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  6.99it/s]
[2023-09-24 04:21:15,416][INFO][modules.shared] - Ending job scripts_txt2img (7.39 seconds)
INFO:     [redacted]:4345 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:23:03,304][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.02it/s]
[2023-09-24 04:23:10,652][INFO][modules.shared] - Ending job scripts_txt2img (7.35 seconds)
INFO:     [redacted]:4450 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:25:37,844][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.05it/s]
[2023-09-24 04:25:45,181][INFO][modules.shared] - Ending job scripts_txt2img (7.34 seconds)
INFO:     [redacted]:4604 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:26:32,920][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.00it/s]
[2023-09-24 04:26:40,295][INFO][modules.shared] - Ending job scripts_txt2img (7.38 seconds)
INFO:     [redacted]:4626 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:27:40,003][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  7.03it/s]
[2023-09-24 04:27:47,359][INFO][modules.shared] - Ending job scripts_txt2img (7.36 seconds)
INFO:     [redacted]:4722 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:28:31,724][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  6.98it/s]
[2023-09-24 04:28:39,120][INFO][modules.shared] - Ending job scripts_txt2img (7.40 seconds)
INFO:     [redacted]:4737 - "POST /sdapi/v1/txt2img HTTP/1.1" 200 OK
[2023-09-24 04:29:21,187][INFO][modules.shared] - Starting job scripts_txt2img
100%|██████████| 50/50 [00:07<00:00,  6.99it/s]
[2023-09-24 04:29:28,576][INFO][modules.shared] - Ending job scripts_txt2img (7.39 seconds)
```

快速使用 ChatGLM 对话模型应用

最近更新时间：2025-06-26 14:57:42

背景介绍

腾讯云高性能应用服务 HAI 是为开发者量身打造的澎湃算力平台。无需复杂配置，便可享受即开即用的 GPU 云服务体验。在 HAI 中，根据应用智能匹配并推选出最适合的 GPU 算力资源，以确保您在数据科学、LLM、AI 作画等高性能应用中获得最佳性价比。

HAI 服务优势

- 智能选型：根据应用匹配推选 GPU 算力资源，实现最高性价比。同时，打通必备云服务组件，大幅简化云服务配置流程。
- 一键部署：分钟级自动构建 LLM、AI 作画等应用环境。提供多种预装模型环境，包含如 StableDiffusion、ChatGLM2 等热门模型。
- 可视化界面：友好的图形界面，AI 调试更为简单。

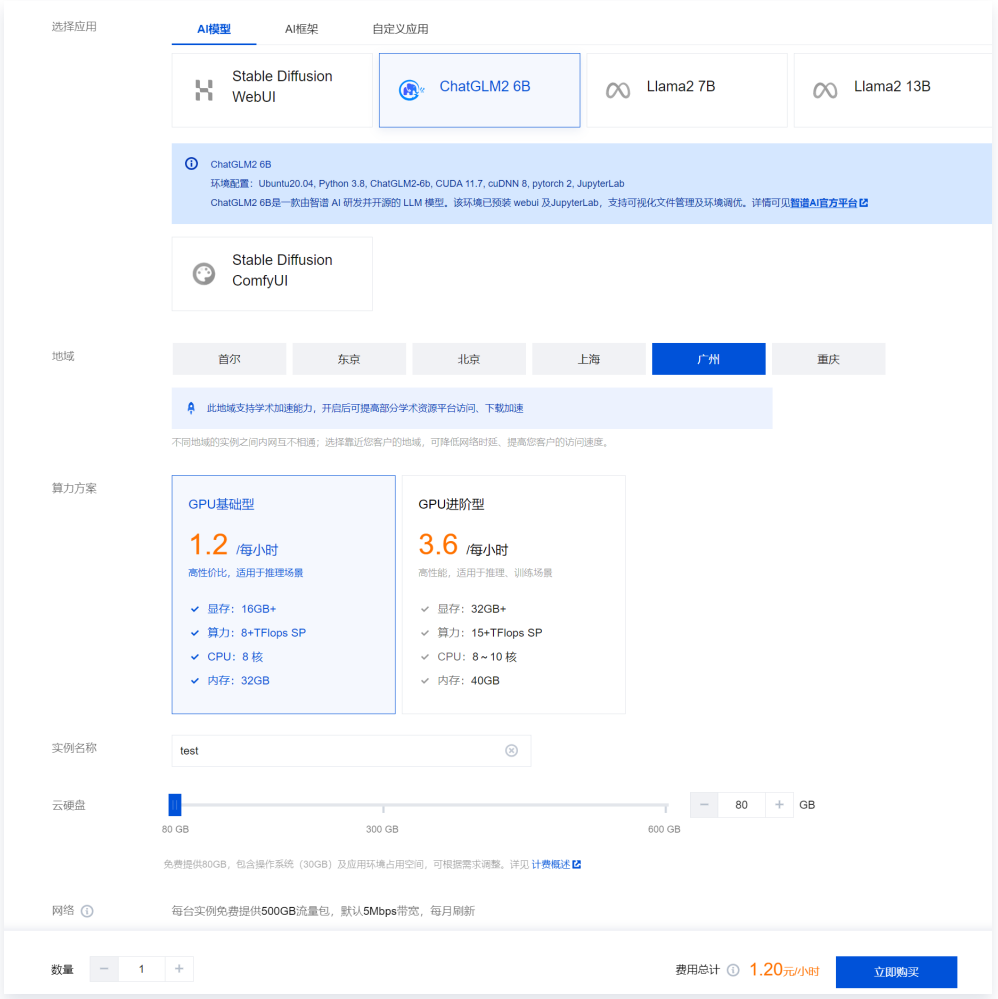
场景介绍

本次我们使用 腾讯云高性能应用服务 HAI 体验快速搭建并使用 AI 模型 **ChatGLM2-6B**，实现思路如下：

- 体验高性能应用服务 HAI 一键部署 **ChatGLM2-6B**。
- 启动 **Gradio WebUI** 进行对话生成。

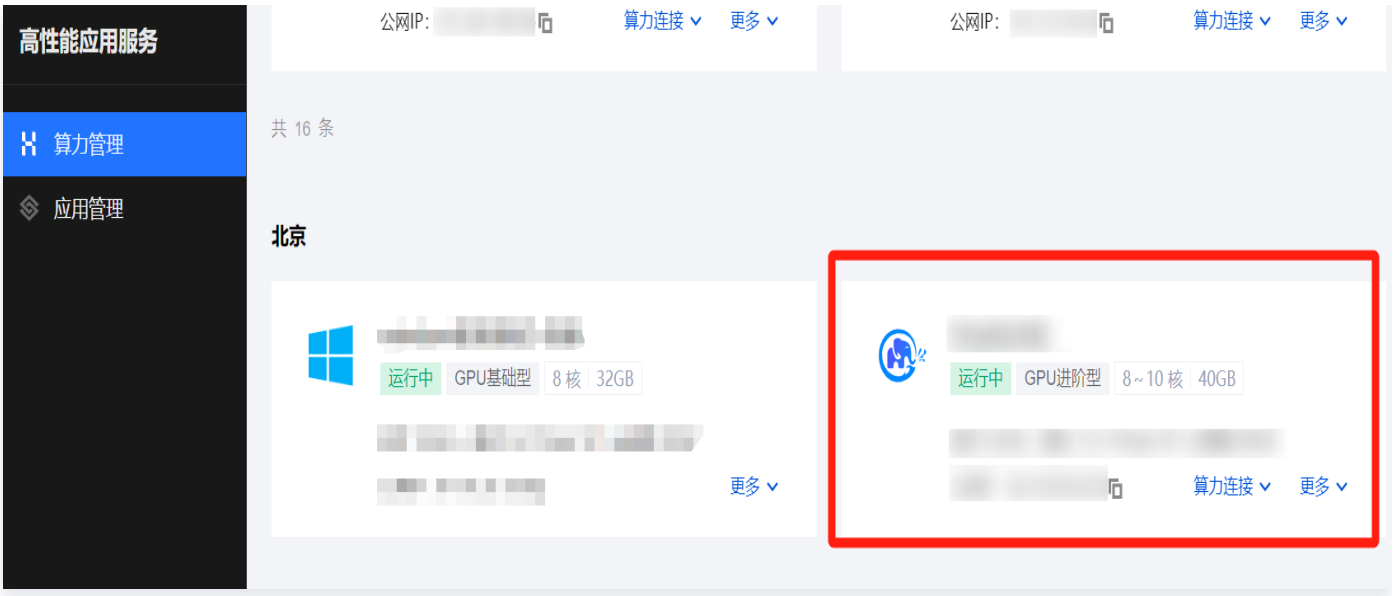
步骤一：快速部署

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。

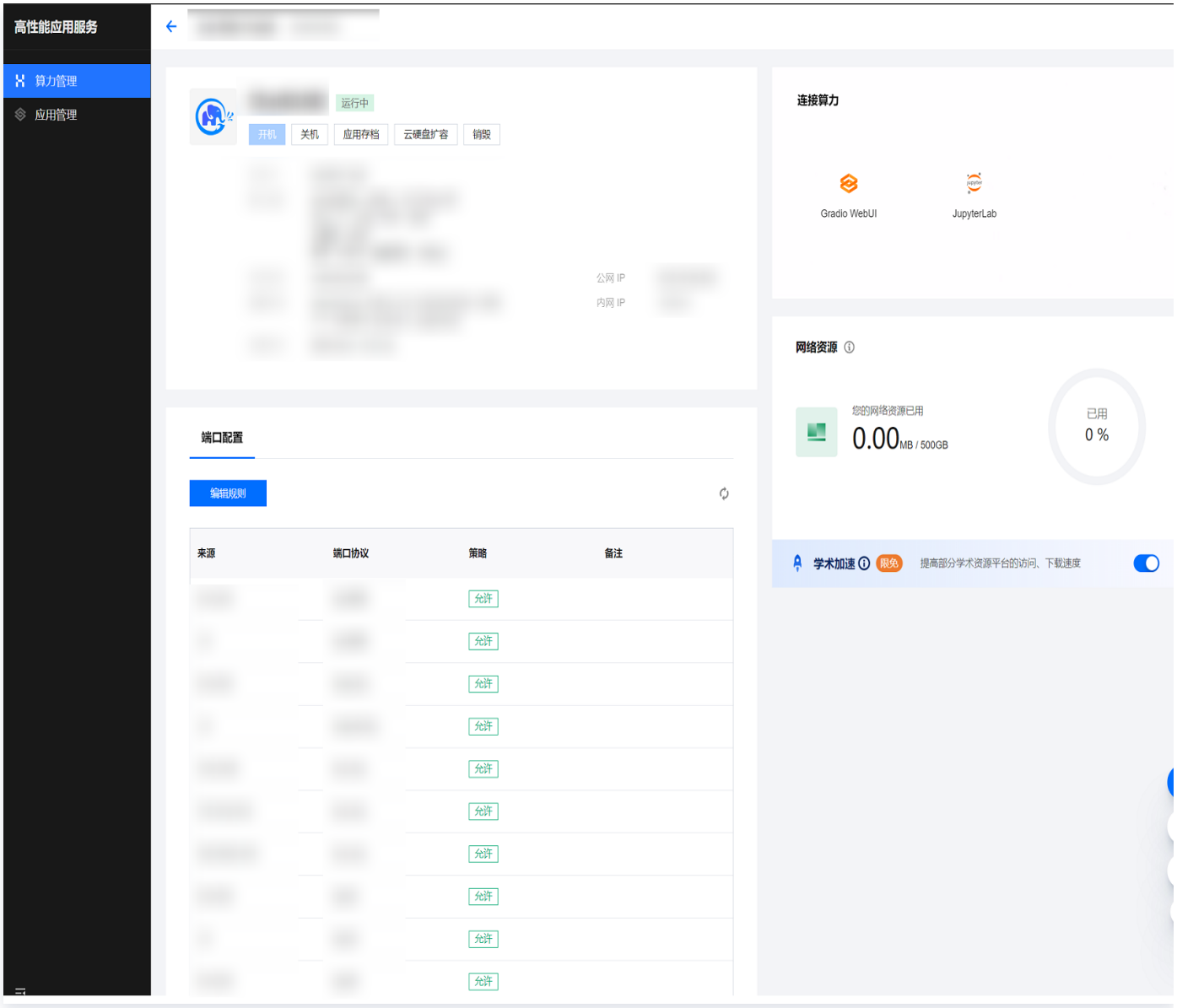


- **选择应用：**目前提供 AI 框架、AI 模型两类应用，请根据实际需求进行选择。
- **地域：**建议选择靠近目标客户的地域，降低网络延迟、提高您的客户的访问速度。
- **算力方案：**本次算力方案选择进阶型，对话生成效率更高。
- **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
- **硬盘：**默认提供 80GB 免费空间，可根据实际使用需求进行调整。
- **网络：**每台实例每月免费提供 500GB 流量包，默认 10Mbps 带宽，每月刷新。
- **购买数量：**默认1台。

3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页。



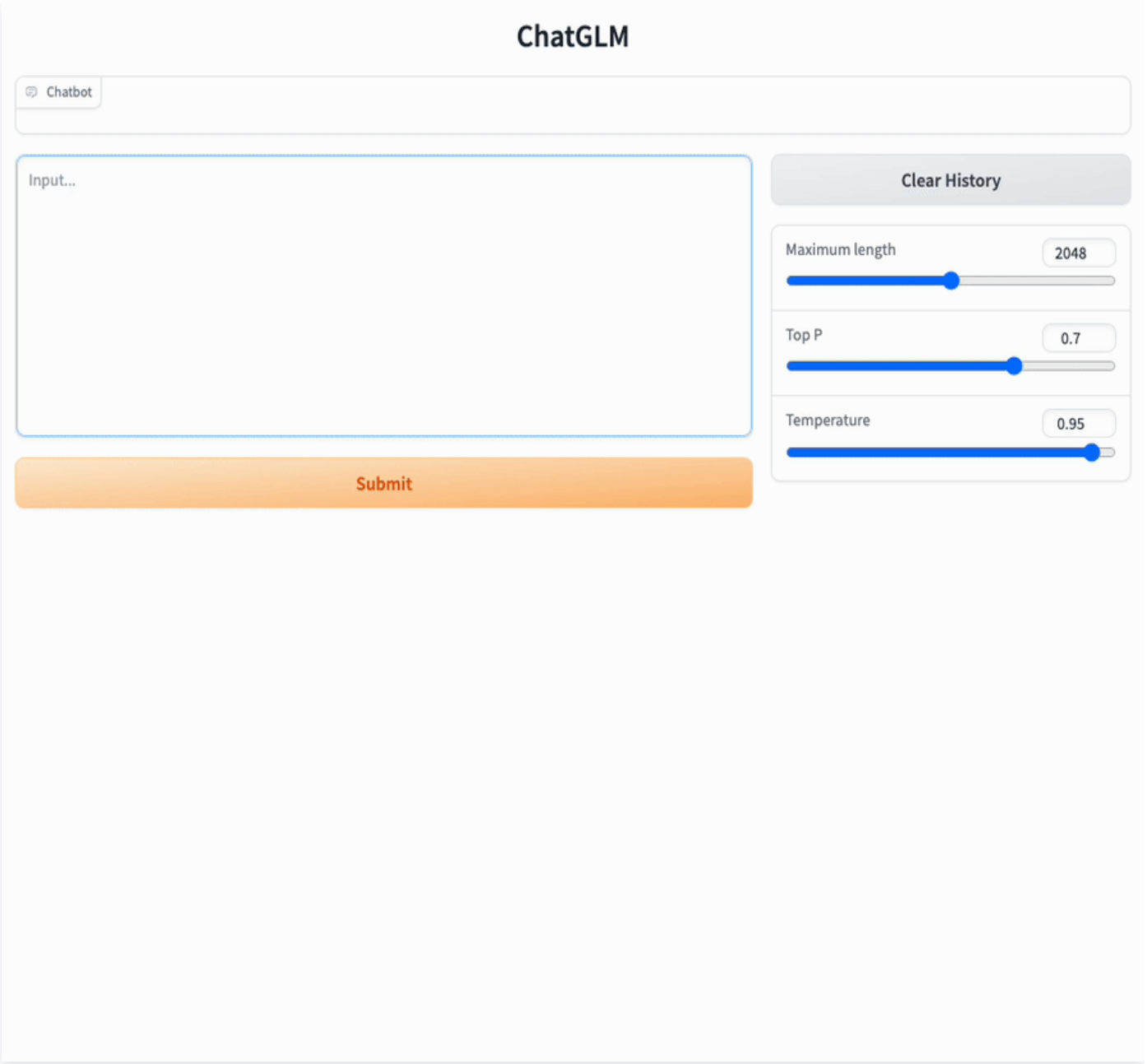
6. 您可以在此页面查看 ChatGLM2-6B 详细的配置信息。



步骤二：启动图形界面

1. 在实例列表中选择**算力连接** > **Gradio WebUI** 并进入该实例的详情页。

2. 使用高性能应用服务 HAI 部署的 **ChatGLM2-6B** 体验简单的对话。



参数说明

- **Maximum length 参数**

  通常用于限制输入序列的最大长度。

  - 因为 ChatGLM2-6B 是2048长度推理的，一般这个保持默认就行，太大可能会导致性能下降。
- **Temperature 参数**

  用于控制模型输出的结果的随机性。

  - 设置为0，对每个 prompt 都生成固定的输出。
  - 较低的值，输出更集中，更有确定性。
  - 较高的值，输出更随机，更有创意。
- **Top-p 参数**

用于控制模型生成文本的概率分布。

- 较小的 Top-p 值会导致模型更加倾向于选择高频词汇。
- 而较大的 Top-p 值则会使模型更加注重选择低频词汇。
- 合适的 Top-p 值能够平衡生成文本的准确性和多样性。

批量导出算力连接方式

最近更新时间：2025-06-26 14:57:42

面向场景

针对**高校课程、教培、企业内部使用**等需要批量创建多台算力，批量导出算力连接方式的场景。

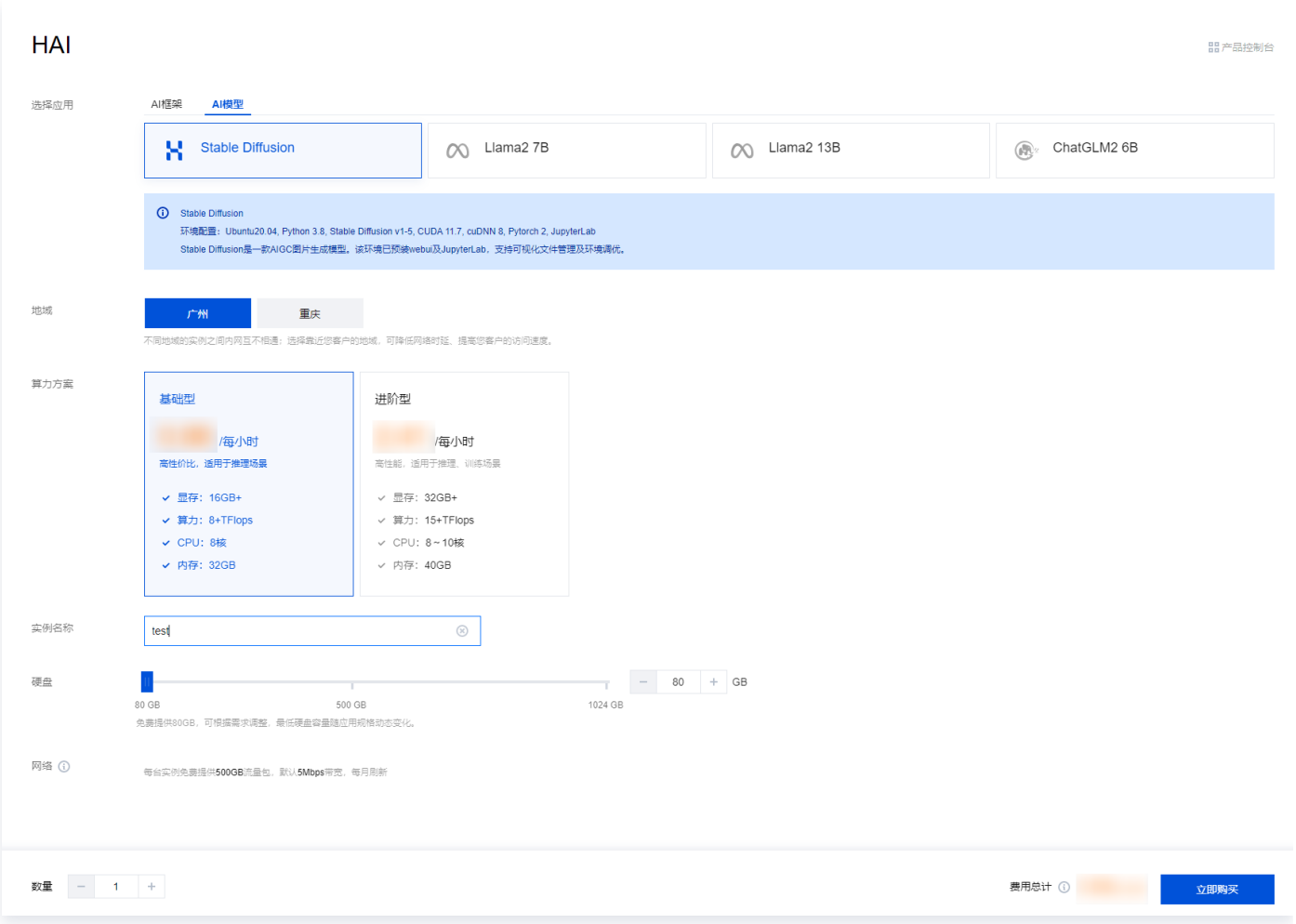
实现效果

支持批量创建多台算力，并将算力连接方式批量导出。导出后您可将算力分发给内部用户使用。

操作步骤

步骤1：创建高性能应用服务 HAI

1. 登录 高性能应用服务 HAI 控制台 。
2. 单击**新建**，进入 高性能应用服务 HAI 购买页面 。



- **选择应用：**目前提供 AI 框架、AI 模型两类应用，请根据实际需求进行选择。
- **地域：**建议选择靠近目标客户的地域，降低网络延迟、提高您的客户的访问速度。

- **算力方案：**本次算力方案选择进阶型，生成图片效率更高。
- **实例名称：**自定义实例名称，若不填则默认使用实例 ID 替代。
- **硬盘：**默认提供 80GB 免费空间，可根据实际使用需求进行调整。
- **网络：**每台实例每月免费提供 500GB 流量包，默认 10Mbps 带宽，每月刷新。
- **购买数量：**默认1台。您可在此处选择多台，目前支持一次性创建10台。

3. 单击**立即购买**。
4. 核对配置信息后，单击**提交订单**，并根据页面提示完成支付。
5. 等待创建完成。单击实例**任意位置**并进入该实例的详情页。

步骤2：批量导出算力连接方式

目前暂不支持通过控制台批量导出算力连接方式。您可 提交工单 联系工作人员，我们会给您提供导出脚本及对应的使用指引。

步骤 3：分发算力连接方式

在完成 步骤 2 后，您将获得一个包含您账号下所有实例连接方式的 Excel 文件。

其他支持的深度学习框架

最近更新时间：2024-08-16 14:23:01

面向场景

HAI 的基础环境支持对深度学习框架的灵活的自定义配置。如果您的业务或研究依赖其他深度学习框架，您可购买 Ubuntu 20.04 应用或 Windows Server 应用，参照如下列表，自行安装其他支持的深度学习框架。

支持列表

HAI 提供可以适配如下框架的完备硬件环境。具体安装指南请参考各个框架官网的最新安装指南。

- Jax
- Keras
- Apache MXNet
- Caffe
- Caffe2
- XGBoost
- Theano
- NIMS
- TFLite
- AutoKeras
- NET
- TFLite：可参见 Tensorflow 官方文档。
- auto-sklearn：可参见 Pypi 的官方文档。

第三方教程

最近更新时间：2025-03-21 09:41:22

本文提供了高性能应用服务 HAI 不同场景下的第三方教程，您可参考教程进行相关实践操作。

说明：

- 由于高性能应用服务 HAI 产品持续更新与迭代，教程中的内容由于时效原因可能与产品最新的能力特性、操作步骤、价格不一致。
- 第三方教程来自 腾讯云开发者社区 ，仅供学习和参考。

推荐教程

- 使用高性能应用服务 HAI 搭建看图识成语益智游戏
- 使用高性能应用服务 HAI 创作小学语文教学方案
- 使用高性能应用服务 HAI 创作小红书笔记
- 使用高性能应用服务 HAI 创作情感小说
- 使用高性能应用服务 HAI 部署Magic-Animate让照片动起来
- 将腾讯云 HAI 上的 DeepSeek 集成到 IDE，打造 AI 代码助手

收录方式

腾讯云鼓励开发者通过 腾讯云开发者社区 （以下简称**社区**）分享最佳实践，我们将不定期收录社区中的优质实践内容作为第三方教程，欢迎参与。