南京科远智慧科技集团股份有限公司

**基于大模型的EmpoworX智能问答系统开发项目**

**工程实施说明**

**（版本：V1.1）**

|  |  |
| --- | --- |
| 编 制： | 靳鸿祥 |
| 审 核： | 顾 坤 |
| 批 准： | 周松明 |
| 受控状态： | 【受控】 |

**各版本建立及修订履历**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | | **建立/修订履历** | **编制人/日期** | **审核人/日期** | **批准人/日期** |
| V1.0 | | 初次建立 | 靳鸿祥/2025.05.12 | 顾坤/2025.05.12 | 周松明/2025.05.12 |
| V1.1 | | 结项发布 | 靳鸿祥/2025.07.31 | 顾坤/2025.07.31 | 周松明/2025.07.31 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1 程序安装 2](#_Toc12044)

[1.1 微服务Setup方式 2](#_Toc3439)

[1.1.1 步骤1：通过自动化运维平台下载智能问答项目并且部署 2](#_Toc3627)

[1.1.2 步骤2：微服务平台端增加Nacos配置信息 2](#_Toc17851)

[1.1.3 步骤3：微服务平台配置智能问答菜单 5](#_Toc2420)

[1.2 Python服务源码启动准备工作 6](#_Toc30717)

[1.2.1 步骤1：安装 Miniconda（推荐） 6](#_Toc21623)

[1.2.2 步骤2：创建虚拟环境 6](#_Toc1848)

[1.2.3 步骤3：启动python后端服务程序 6](#_Toc5399)

[1.3 Python服务安装包部署 7](#_Toc15624)

[1.3.1 下载安装包，修改配置文件 7](#_Toc17593)

[1.3.2 启动文件解析服务 7](#_Toc4751)

[1.3.3 启动web服务 8](#_Toc4094)

[1.4 模型部署过程 8](#_Toc25033)

[1.4.1 安装显卡驱动 8](#_Toc20819)

[1.4.1.1 安装NVIDIA驱动 8](#_Toc19046)

[1.4.1.2安装CUDA加速 10](#_Toc25806)

[1.4.3配置CUDA环境变量 11](#_Toc11936)

[1.4.4 验证CUDA是否安装成功 12](#_Toc18490)

[1.4.2 安装xinference 13](#_Toc4013)

[1.4.3 启动嵌入模型 14](#_Toc20543)

[1.4.4 下载安装大模型 14](#_Toc3764)

[2 配置前需要和客户沟通的主要问题 16](#_Toc17415)

[3 系统配置 16](#_Toc15980)

[4 系统卸载 17](#_Toc31440)

[4.1 Setup方式 17](#_Toc4826)

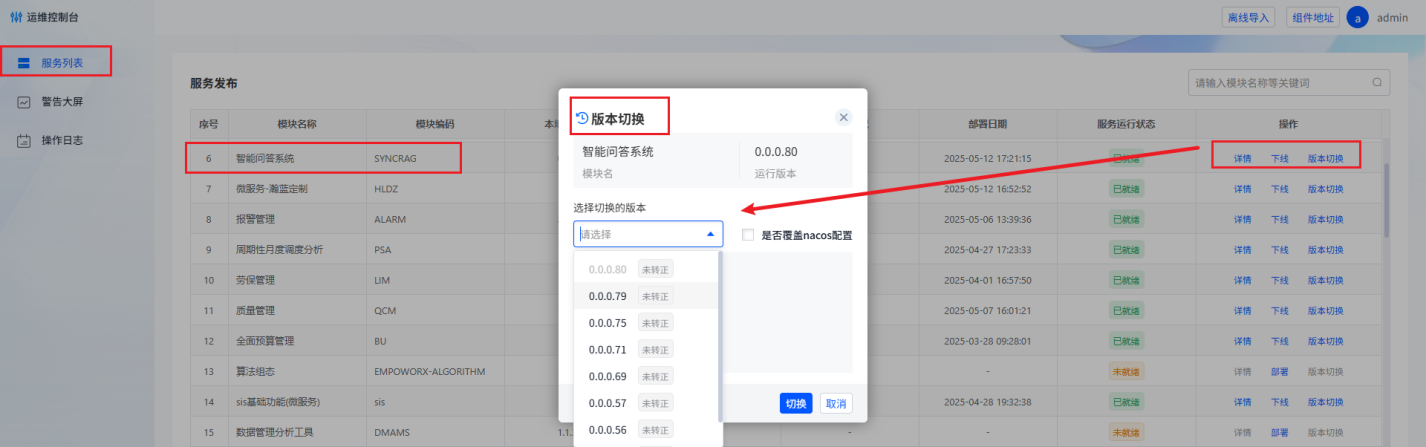
[4.2 文件复制方式 17](#_Toc13131)

[5 部署需注意事项 17](#_Toc26559)

# 程序安装

## 微服务Setup方式

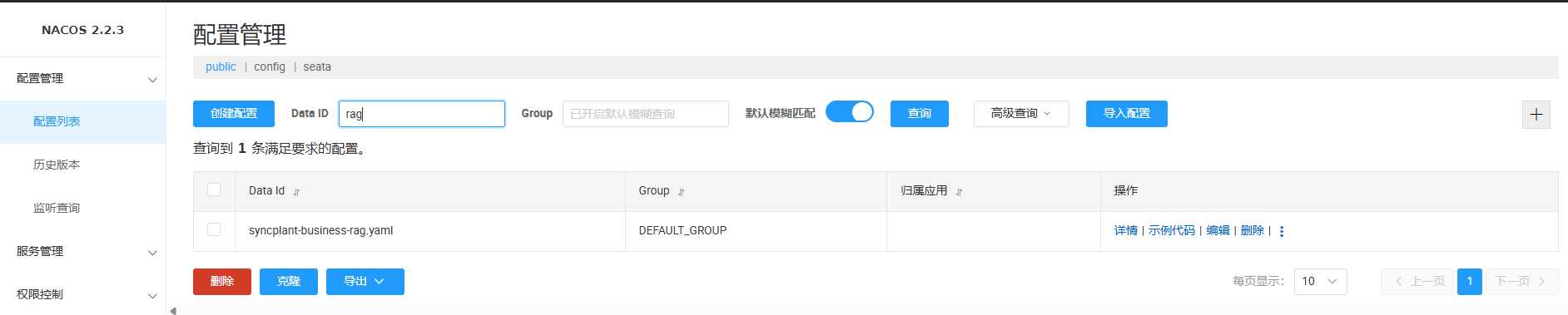
### 步骤1：通过自动化运维平台下载智能问答项目并且部署



注：如果对运维平台或nacos等工具不熟悉，不会使用，请咨询平台部署同事

### 步骤2：微服务平台端增加Nacos配置信息

nacos新增配置syncplant-business-rag.yaml，具体配置如下：



#服务器端口

server:

port: 8080

tomcat:

max-threads: 2000

spring:

application:

chinesename: 智能问答

name: syncplant-business-rag

datasource:

url: jdbc:mysql://mysql-primary.middle:3306/SYNCRAG?useSSL=false&useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull&transformedBitIsBoolean=true&tinyInt1isBit=false&allowMultiQueries=true&serverTimezone=GMT%2B8&allowPublicKeyRetrieval=true&connectTimeout=10000&socketTimeout=10000

username: root

password: syncplant

hikari:

minimum-idle: 10

maximum-pool-size: 50

connection-timeout: 60000

mybatis-plus:

mapper-locations: classpath:mapper/\*\*/\*Mapper.xml

sciyon:

log:

level: INFO

pythonService:

host: 10.3.24.40

port: 9889

url: http://10.3.24.40:9889

elastic:

host: 10.3.24.40

port: 9200

username: elastic

password: infini\_rag\_flow

indexName: ragflow\_sciyonff8bcdc11efbdcf88aedd524325

jedis:

db: 1

host: 10.44.2.168

password: Redissciyon2024\*-+

port: 31117

minio:

bucket: sciyon

host: 10.44.2.168:31507

password: Miniosciyon2024\*-+

user: minio

mysql:

host: 10.44.2.168

max\_connections: 2000

name: syncrag

password: Mysqlsciyon2024\*-+

port: 30136

stale\_timeout: 30

user: root

# 模型

model:

embedding\_batch\_size: 16

max\_llm\_len: 6000

rerank\_batch\_size: 16

nltk:

path: utils/nltk\_data

# 嵌入模型

embedding:

provider: 'Xinference'

base\_url: 'http://10.3.24.46:9997'

model\_name: 'bge-large-zh-v1.5'

key: '123'

# 重排序模型

rerank:

provider: 'Xinference'

base\_url: 'http://10.3.24.46:9997'

model\_name: 'bge-large-zh-v1.5'

key: '123'

# 搜索配置项

search:

es\_index: ragflow\_sciyonff8bcdc11efbdcf88aedd524325

hit\_test\_size: 20

page: 1

page\_size: 10

similarity\_threshold: 0.5

top: 1024

vector\_similarity\_weight: 0.5

**注意事项：**

1、其中dataSource数据库配置根据现场实际情况进行替换，url中ip和数据库名根据实际情况替换，username和password也要同步修改。由于python服务需要真实的mysql连接信息，需要查出现场真实的ip、端口、数据库名、用户名、密码填入到mysql下对应的host,port,name,user,password中。如果不是mysql数据库，对应的链接地址和驱动也要进行相应的更改。

2、智能问答项目需要启动python服务，需要根据现场实际运行的python服务来修改上述配置文件中pythonService中host、port和url。

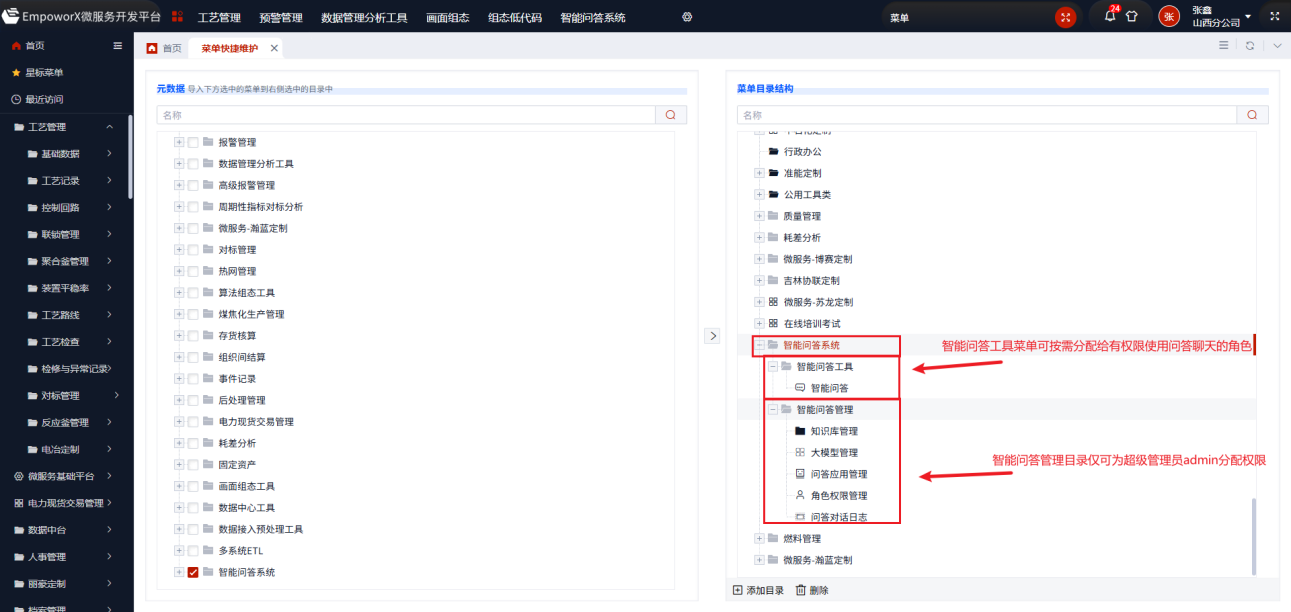
3、Redis需要查出现场真实的ip、端口、密码和db库，填入对应的host,port,password,db中。

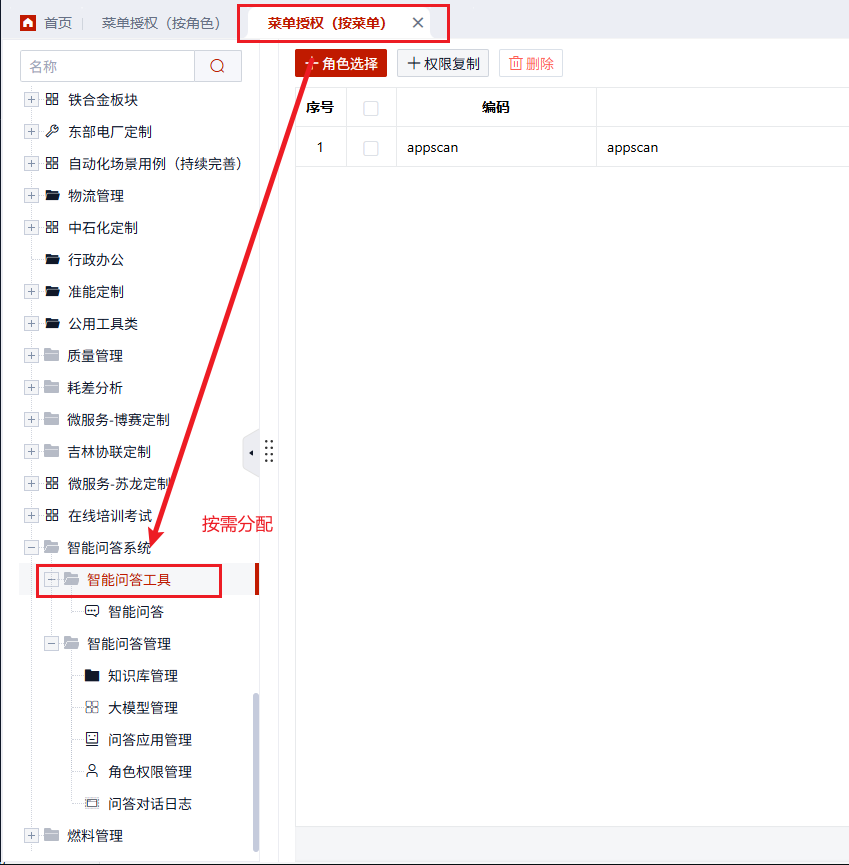
4、Minio服务器的参数需要查出现场的真实ip端口以及密码，填入对应的host,user和password中，指定桶名bucket:sciyon。

5、Elastic数据库的地址也要根据现场实际情况进行调整，并且指定索引的名称indexName。其余和python模型相关的参数（model,nltk,embedding,rerank,search）请联系开发人员进行实际配置。

### 步骤3：微服务平台配置智能问答菜单

菜单无需手动配置。已通过平台菜单元数据在部署时导入。但仍需配置菜单权限





## Python服务源码启动准备工作

### 步骤1：安装 Miniconda（推荐）

# 下载 Miniconda 安装脚本

wget <https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh>

# 运行安装脚本

bash Miniconda3-latest-Linux-x86\_64.sh

# 激活conda 命令

source ~/.bashrc

# 查看是否安装成功

conda --version

以 64-bit Linux 系统为例，执行以下命令下载最新的 Miniconda 安装脚本：

### 步骤2：创建虚拟环境

# 创建新的虚拟环境

conda create -n myenv python=3.10

# 激活虚拟环境

conda activate myenv

# 安装依赖

pip install -r requirements.txt

### 步骤3：启动python后端服务程序

# 进入项目中python服务目录

cd src/backend/syncplant-business-rag-service

# 启动web\_server.py程序

python web\_server.py

# 启动task\_executor.py程序

python task\_executor.py

启动python服务程序后，控制台会打印日志，检查日志信息，若没有报错，则成功启动python服务程序。

**注意事项:**

1. 若启动失败，需检查文件src/backend/syncplant-business-rag-service/conf/syncplant -business-rag-service.yaml中是否正确配置nacos信息，如下图；

2、若src/backend/syncplant-business-rag-service/conf/syncplant -business-rag-service.yaml文件没有问题，进一步返回1.1.2步骤检查nacos中的数据库配置信息是否有误。

# 指定python服务的端口号  
web:  
 host: 0.0.0.0  
 http\_port: 9890  
  
# 104开发环境  
nacos:  
 server\_addresses : "10.44.2.104:8848"  
 namespace : ""  
 group : "DEFAULT\_GROUP"  
 username : "nacos"  
 password : "nacos"  
 data\_id : "syncplant-business-rag"

## Python服务安装包部署

### 下载安装包，修改配置文件

目前支持linux系统的安装包，其中cuda版本为12.2

配置文件位于“Empoworx-RAG/\_internal/conf/syncplant -business-rag-service.yaml”，在配置文件中是否正确配置数据库信息。

# 指定python服务的端口号  
web:  
 host: 0.0.0.0  
 http\_port: 9890  
  
# 104开发环境  
nacos:  
 server\_addresses : "10.44.2.104:8848"  
 namespace : ""  
 group : "DEFAULT\_GROUP"  
 username : "nacos"  
 password : "nacos"  
 data\_id : "syncplant-business-rag"

**注意事项：**

1. web中的http\_port:9890端口号可以根据现场实际需求更改；
2. Nacos中的配置server\_addresses:10.44.2.104:8848也可以根据现场实际进行更改。

### 启动文件解析服务

在linux系统下面，执行命令行

# 进入安装包

cd Empoworx-RAG

# 执行命令

./Empoworx-RAG-task

注意事项：将命令挂载在后台终端，通过screen命令

### 启动web服务

在linux系统下面，执行命令行

# 进入安装包

cd Empoworx-RAG

# 执行命令

./Empoworx-RAG-web

注意事项：将命令挂载在后台终端，通过screen命令或者systemd命令

## 模型部署过程

### 安装显卡驱动

#### 1.4.1.1 安装NVIDIA驱动

注意事项：以下安装环境需要在linux系统下进行操作

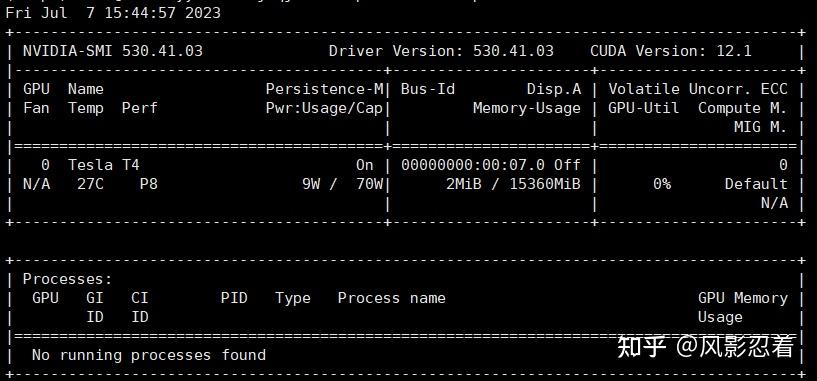
# 确认电脑上安装了 NVIDIA 显卡

lspci | grep -i nvidia



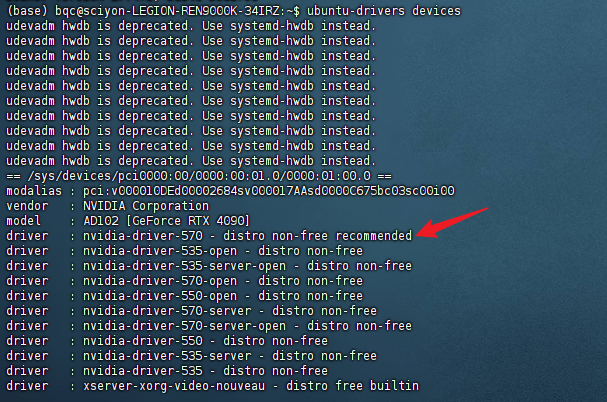
# 确认有显卡以后输入下面命令，以检查之前是否安装了驱动

nvidia-smi



# 如果发现没有安装过驱动，则可再输入下面命令，以查看显卡型号以及推荐的驱动版本。

ubuntu-drivers devices

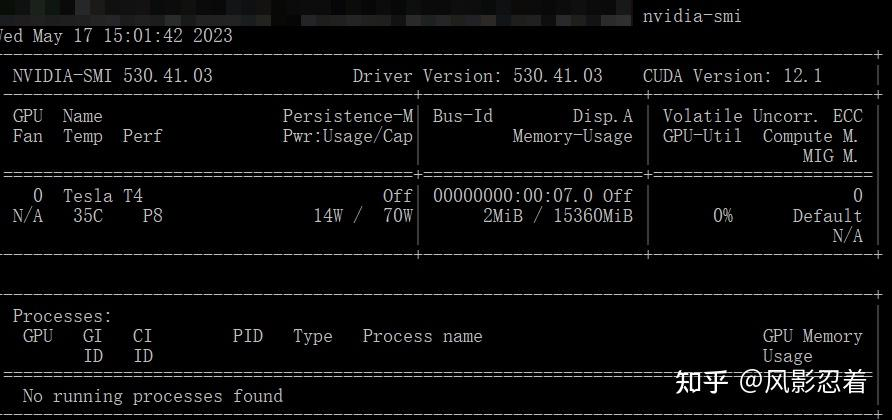


# 默认按照推荐的安装驱动，则输入即可安装该驱动：

sudo ubuntu-drivers autoinstall

# 安装过程中按照提示操作，所有的提示都选择默认，安装完成后重启系统，显卡就可以正常工作了，再次输入

nvidia-smi

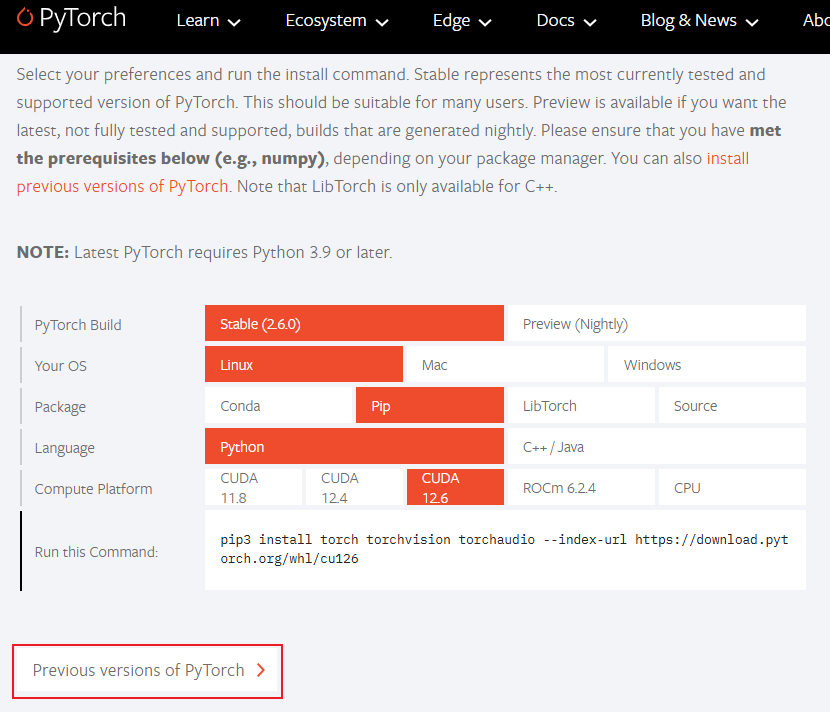


#### 1.4.1.2安装CUDA加速

CUDA（Compute Unified Device Architecture）是 NVIDIA 推出的一个并行计算平台和编程模型，它允许你使用 NVIDIA 的 GPU 来加速计算任务，主要通过 C/C++ 或 Python API 实现。而python服务安装包中的PyTorch 基于 CUDA 来支持 GPU 加速，因此必须要安装满足pytorch要求的CUDA开发工具包。

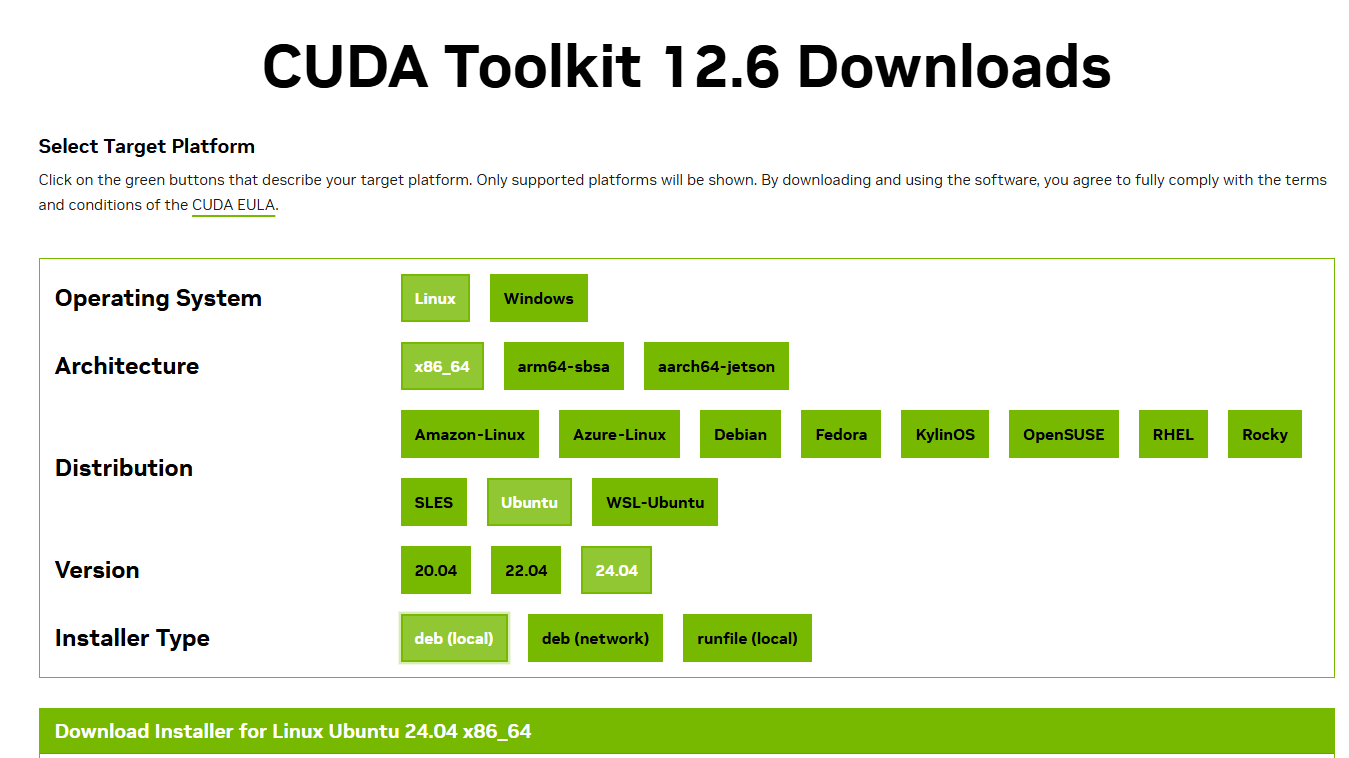
**注意事项:**

首先查看当前官网pytorch支持的CUDA版本（<https://pytorch.org/>），如下图pytorch官网显示当前支持最新的CUDA版本为12.6，则在下载的CUDA版本不能超过12.6，否则pytorch无法使用CUDA来加速计算任务。这是由于NVIDIA官网所支持的CUDA版本通常是最新的，而torch还没有及时更新兼容CUDA最新版本。



**安装 CUDA步骤：**

NVIDIA CUDA 工具包可点击<https://developer.nvidia.com/cuda-downloads>进行获取。所安装的CUDA不能超过torch规定的最新cuda版本，这里以安装12.6版本的CUDA为例（其他版本的CUDA如12.2也是按照下面的流程），如下图所示：





根据上图中的命令行脚本，依次复制命令行粘贴到控制台回车即可：

最后，驱动安装完成后，驱动安装完成后，请重启系统以加载驱动：

sudo reboot

#### 1.4.3配置CUDA环境变量

使用以下命令行打开 .bashrc 文件，准备添加环境变量。

sudo vim ~/.bashrc

若打开的.bashrc文件中大致有以下内容：

bash

# 设置PATH环境变量

export PATH=$PATH:/usr/local/bin

# 创建别名

alias ll='ls -l'

# 自定义提示符

PS1='\u@\h:\w\$ '

# 运行其他Shell脚本

if [ -f ~/custom\_scripts.sh ]; then

. ~/custom\_scripts.sh

Fi

则在文件中追加以下代码，如果没有，则将上述文件中脚本内容和下述脚本内容一起复制到.bashrc文件中。

export PATH=/usr/local/cuda-12.6/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda-12.6/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

脚本解释：这是一个设置环境变量的命令，将CUDA 12.6的二进制文件目录添加到系统的PATH环境变量中。

然后按下 Esc 键，从编辑模式返回命令模式。

然后输入英文冒号进入末行模式。

然后输入 wq 并回车，表示保存并退出。

最后，通过下述命令行激活.bashrc文件中的内容。

source ~/.bashrc

#### 1.4.4 验证CUDA是否安装成功

终端里面依次输入下面代码：

python

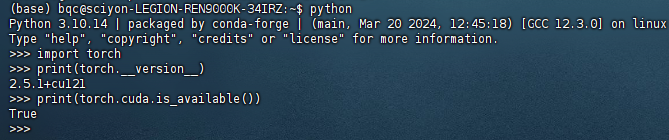
然后

import torch

然后

print(torch.version.cuda)

显示如下图，则表示安装CUDA成功！



到此，已经完成CUDA的安装。

### 安装xinference

# 利用1.2.1节安装的Miniconda创建虚拟环境

conda create --name xinference python=3.11

# 激活虚拟环境

conda activate xinference

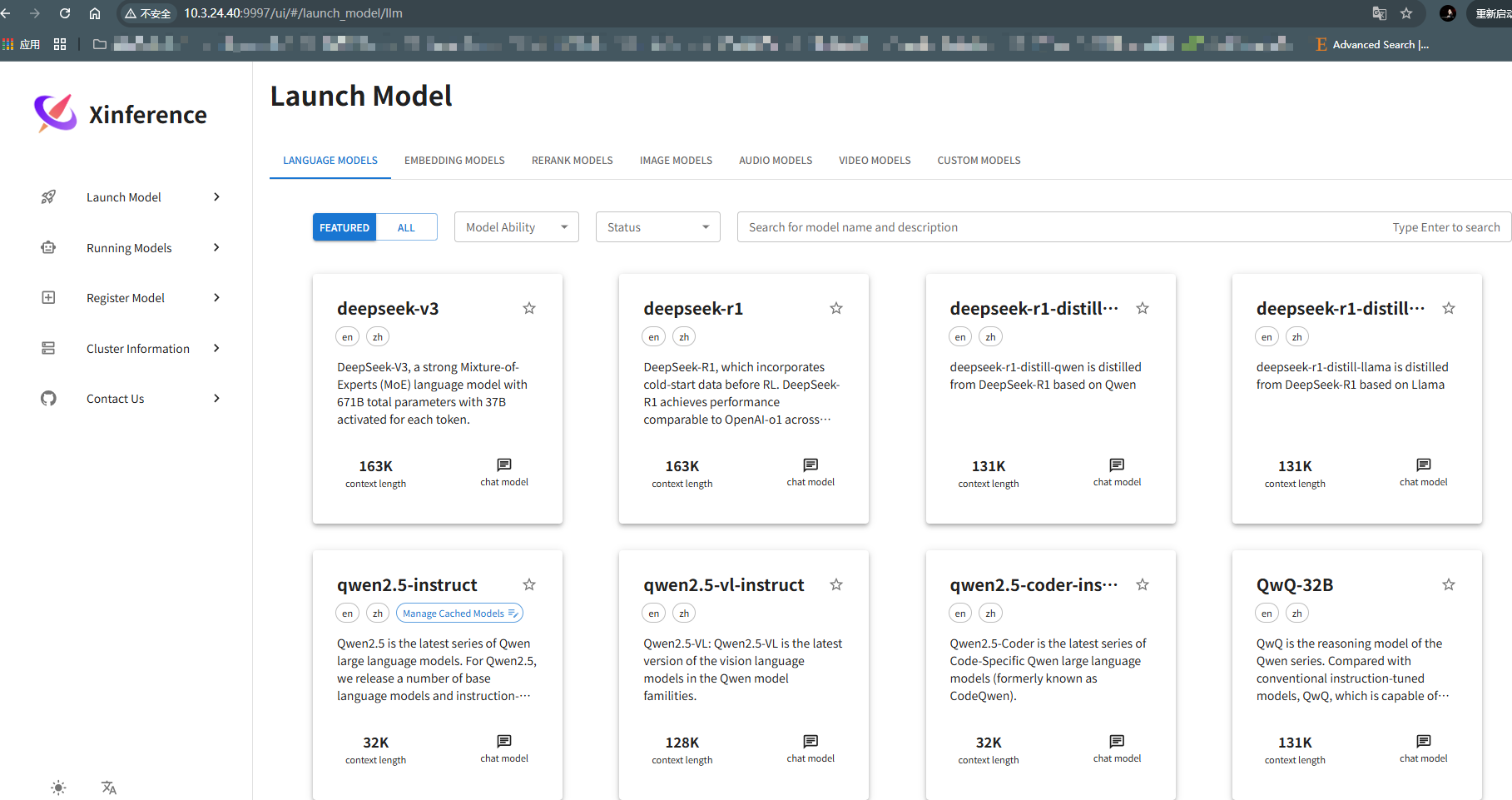
# 安装xinference的推理包，选择[all]能够推理所有支持的模型，用以下命令安装所有需要的依赖

pip install "xinference[all]"

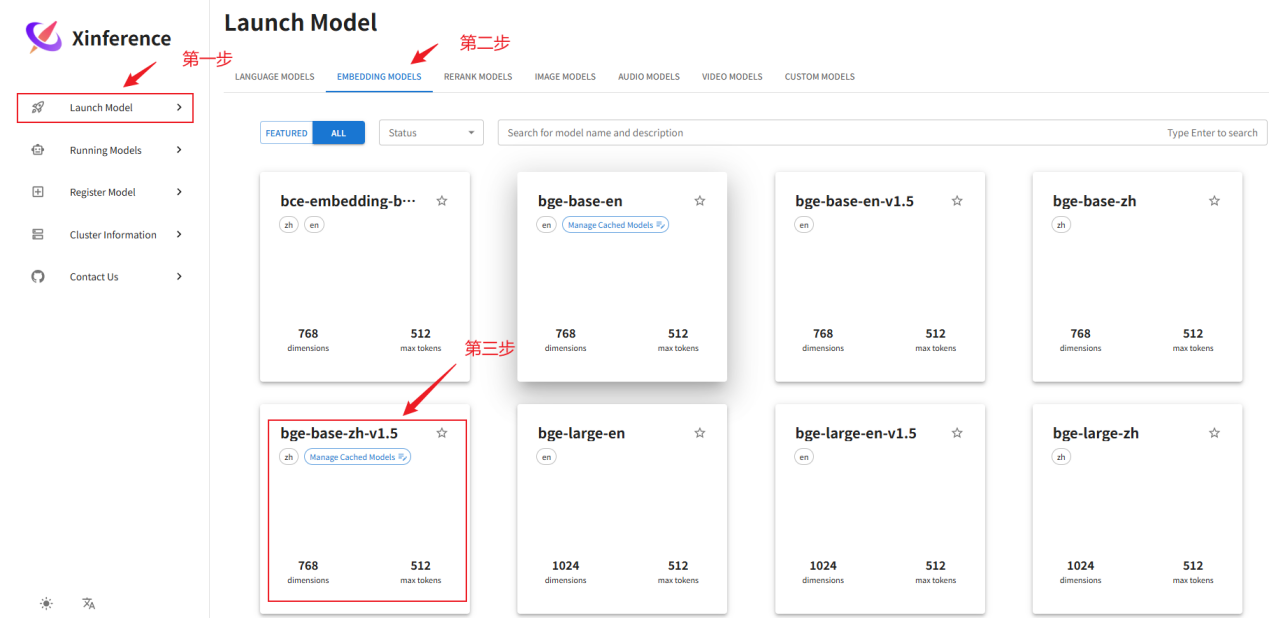
# 启动本地inference服务，使用如下命令行：

XINFERENCE\_MODEL\_SRC=modelscope xinference-local --host 0.0.0.0 --port 9997

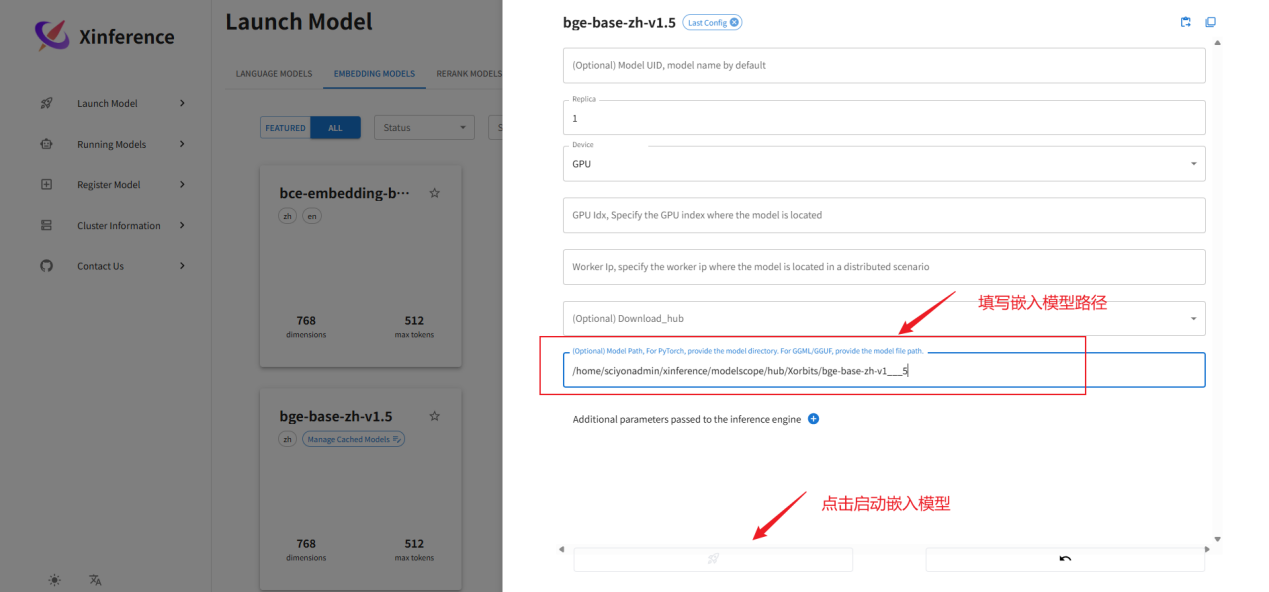
启动xinference服务后，可以通过访问 <http://127.0.0.1:9997/ui>使用图形界面管理工具。



### 启动嵌入模型



弹出弹窗



Xinference图形界面启动嵌入模型

### 下载安装大模型

方式1：

* 通过命令行的方式启动大模型

第一次运行大模型时，首先要运行是要从HuggingFace 下载大模型，一般来说需要根据模型大小下载10到30分钟不等。当下载完成后，Xinference本地会有缓存的处理，以后再运行相同的模型不需要重新下载。

# 下载和启动模型（核心步骤）,使用下面的命令行：

xinference launch --model-engine vllm --model-uid qwen2.5-instruct --model-name qwen2.5-instruct --quantization none --size-in-billions 7 --model-format pytorch

命令行参数解释：

命令行解释：

-model-engine 模型推理引擎

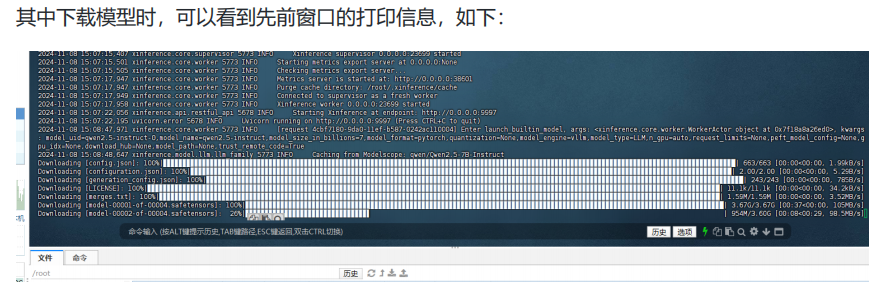
--model-uid 模型的id名称

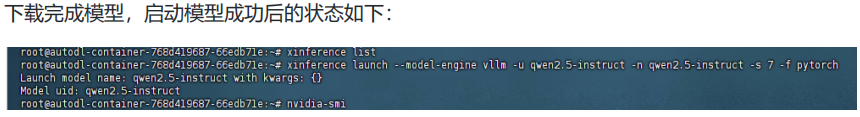
--model-name 模型的名称

--quantization 模型是否量化

--size-in-billions 模型的参数

--modelo-format 模型的格式





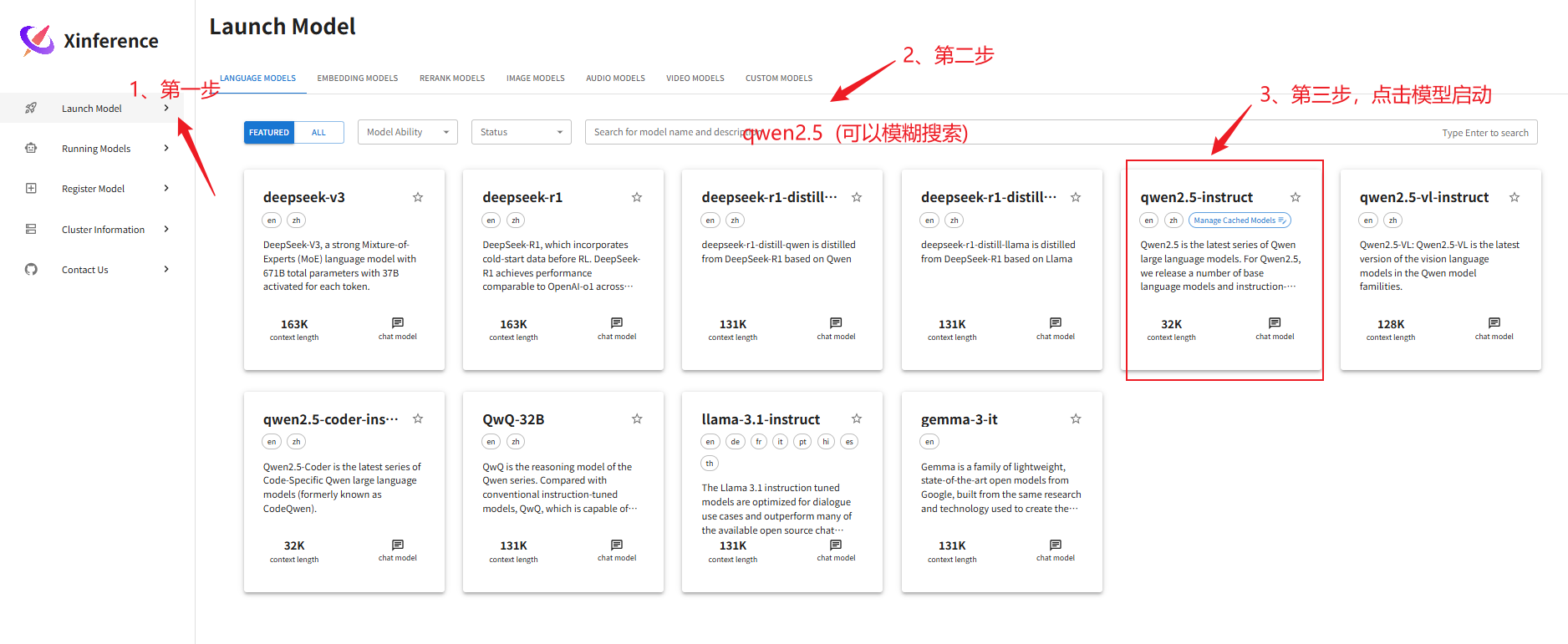
# 停止正在运行的模型，其中“qwen2.5-instruct”为模型名称

xinference terminate --model-uid “qwen2.5-instruct”

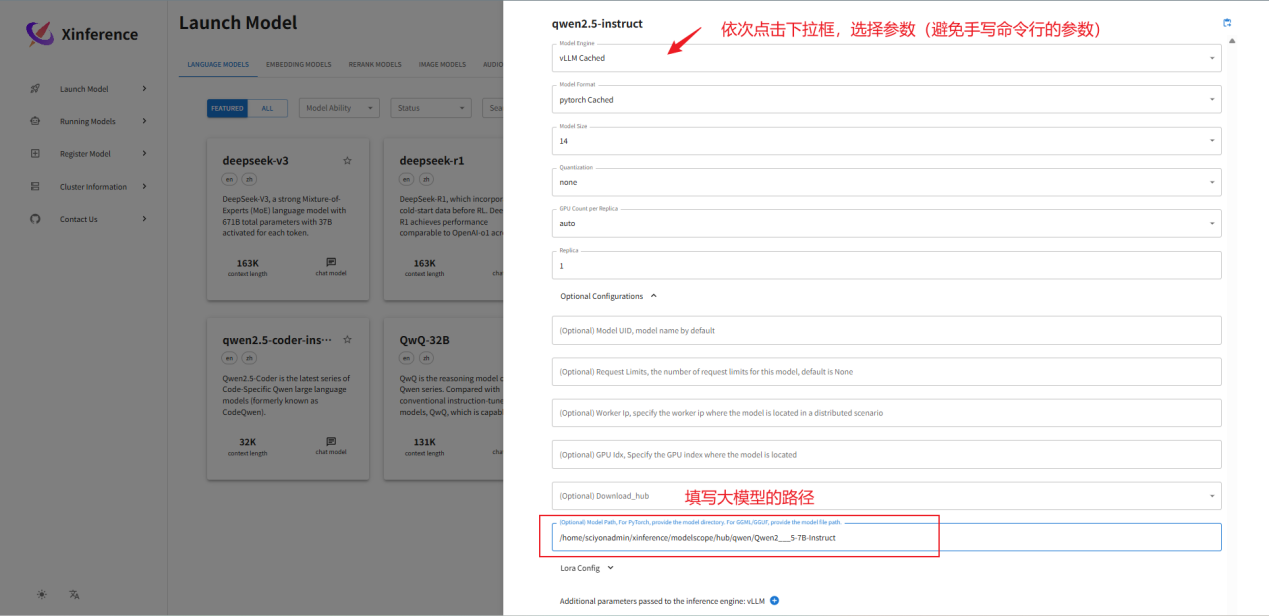
方式2（**推荐！！！**）：

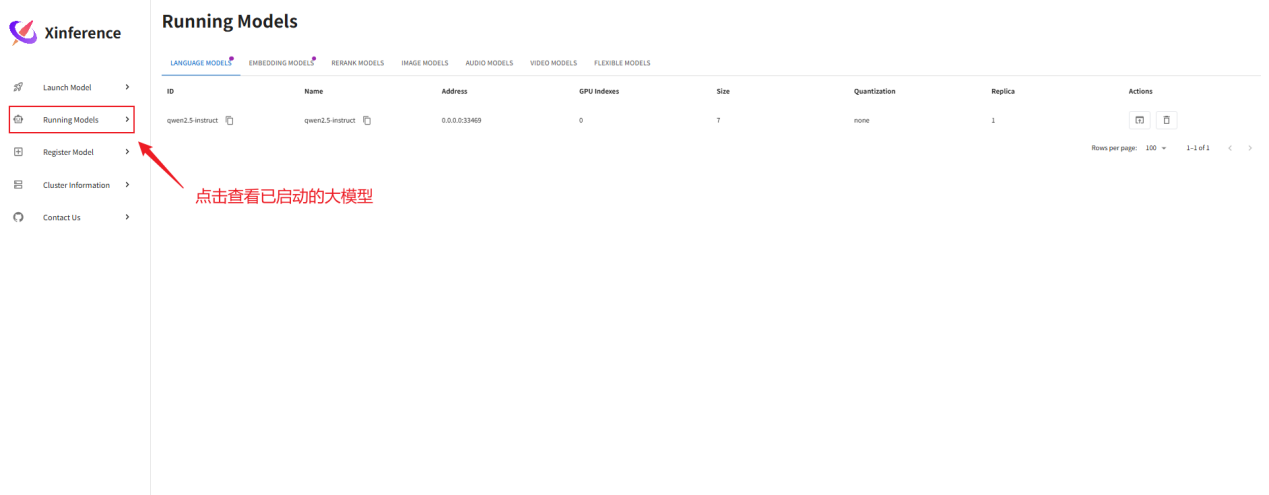
* 图形化启动大模型

启动xinference服务后，可以通过访问 <http://127.0.0.1:9997/ui>使用图形界面管理工具。



弹出弹窗，配置大模型的启动参数。





Xinference 图形界面启动大模型

**注意事项：**

1、启动大模型之前必须先启动嵌入模型!!!（参考1.4.3节启动嵌入模型步骤）

2、填写大模型的路径（非必填），不填写路径时，xinference后台会自动下载大模型（大概10分钟到30分钟，根据模型大小而定），填写大模型路径后，则不需要重新下载大模型，直接加载已有的本地模型。

# 配置前需要和客户沟通的主要问题

无

# 系统配置

无

# 系统卸载

## Setup方式

无

## 文件复制方式

无

# 部署需注意事项

无