

# Manus 本地部署（教程版）

## 准备：

1、开源地址：<https://github.com/mannaandpoem/OpenManus>

但很多人无法下载

2、OpenManus 直接下载 ZIP 包：

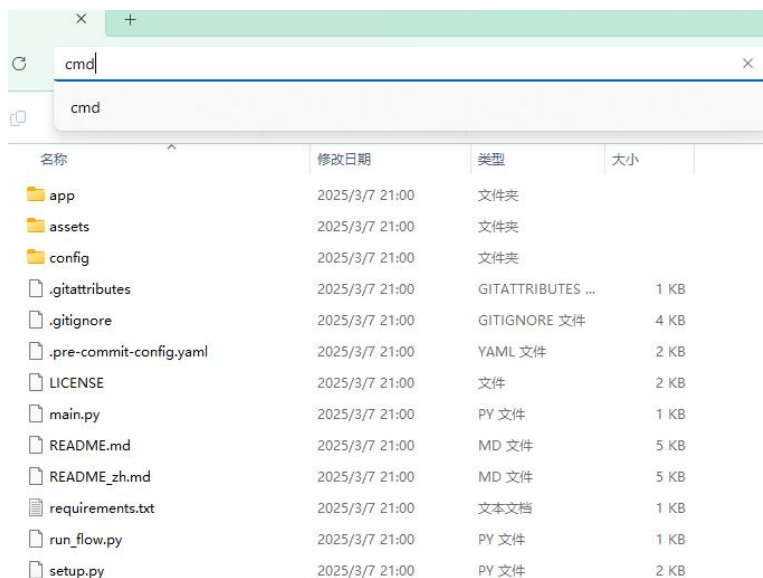
<https://www.faxianai.com/wp-content/uploads/2025/03/1741408550-OpenManus-main.zip>

3、相关资料《Manus 邀请码获取方法 Manus 使用指南》

<https://www.faxianai.com/data/10090.html>

## 开始部署

1、进入 OpenManus 项目根目录，在路径栏输入 cmd，回车，进入根目录路径下的控制台



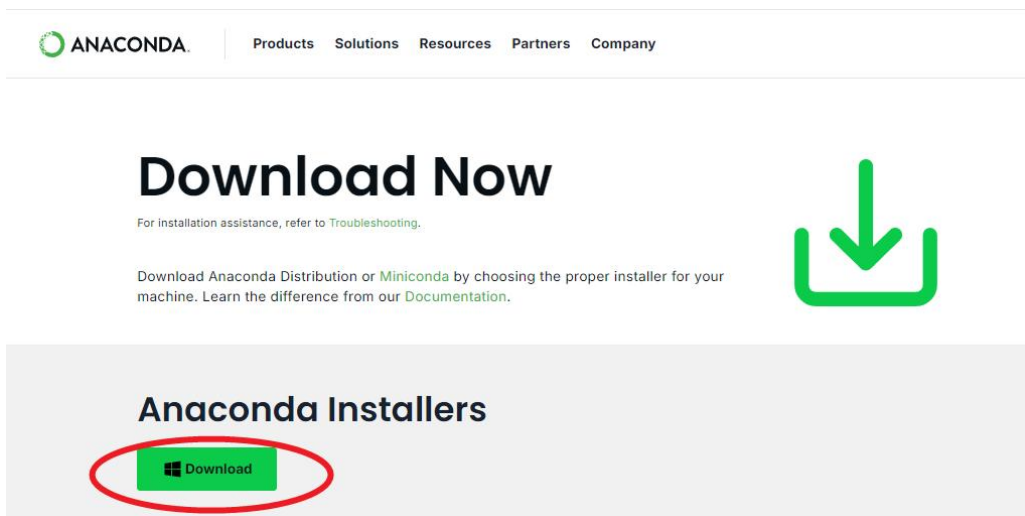
2、指令如下

```
conda create -n open_manus python=3.12
```

### 3、如果出现如下报错

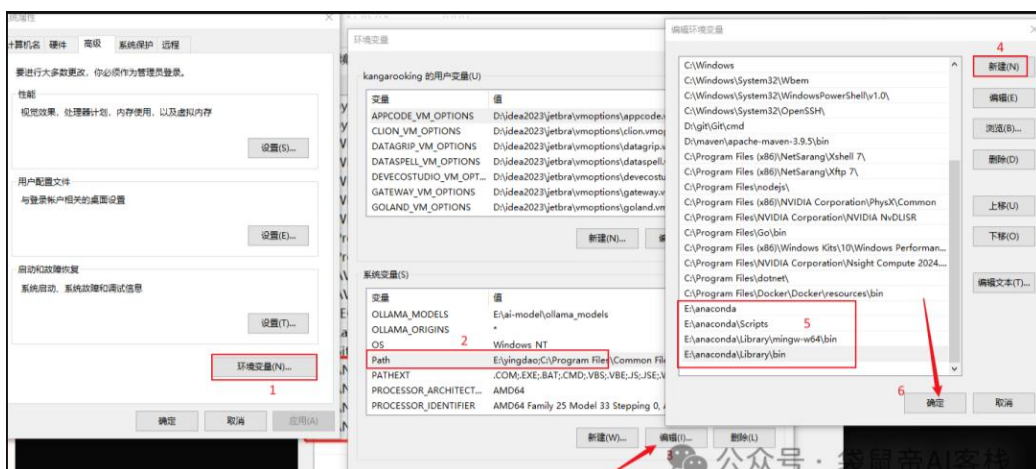
```
D:\myGit\OpenManus>conda create -n open_manus python=3.12
'conda' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

请先安装 anaconda: <https://www.anaconda.com/download/success>



### 配置 conda 的环境变量

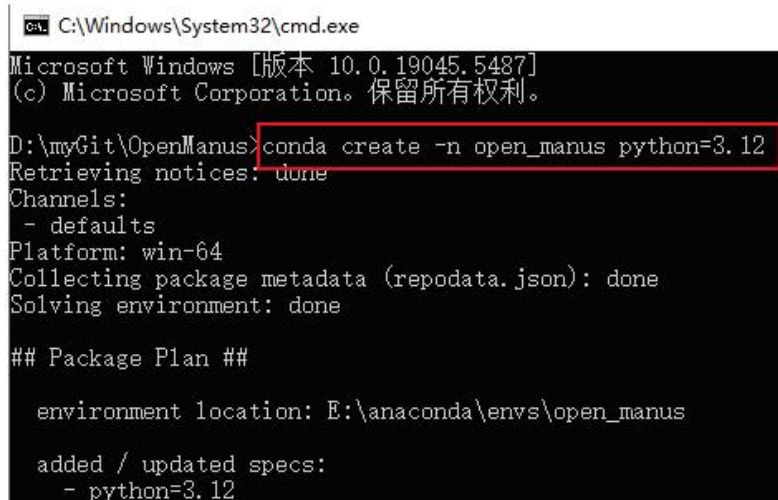
E:\anaconda`E:\anaconda\Scripts`E:\anaconda\Library\mingw-w64\bin`E:\anaconda\Library\bin



PS: 配置完一定要逐个窗口点击确定, 确保保存下来了, 以及 cmd 的窗口需要关闭, 重新进入。

4、按照刚才的方式, 重新开一个控制台窗口, 输入指令。

```
conda create -n open_manus python=3.12
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.5487]
(c) Microsoft Corporation。保留所有权利。

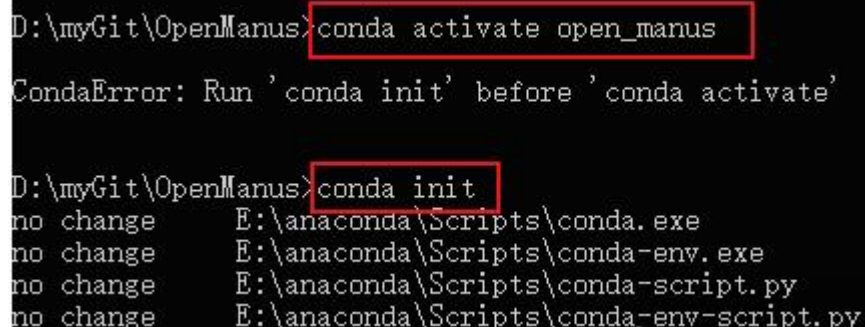
D:\myGit\OpenManus>conda create -n open_manus python=3.12
Retrieving notices: done
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: E:\anaconda\envs\open_manus

added / updated specs:
  - python=3.12
```

5、这步本来是要执行 `conda activate open_manus`, 但是报错提示需要先执行 `conda init` (初始化)



```
D:\myGit\OpenManus>conda activate open_manus
CondaError: Run 'conda init' before 'conda activate'

D:\myGit\OpenManus>conda init
no change      E:\anaconda\Scripts\conda.exe
no change      E:\anaconda\Scripts\conda-env.exe
no change      E:\anaconda\Scripts\conda-script.py
no change      E:\anaconda\Scripts\conda-env-script.py
```

6、执行完 `conda init`, 记得需要重新 cmd, 新开一个 OpenManus 根目录下的控制台窗口, 执行如下指令

```
conda activate open_manus
```

```
C:\Windows\System32\cmd.exe - "E:\anaconda\condabin\conda.b...
Microsoft Windows [版本 10.0.19045.5487]
(c) Microsoft Corporation。保留所有权利。

D:\myGit\OpenManus>conda activate open_manus

(open_manus) D:\myGit\OpenManus>
```

## 7、接下来是安装项目的依赖项

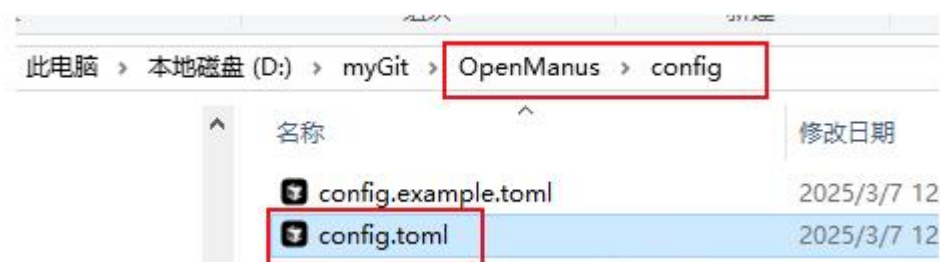
```
pip install -r requirements.txt
```

```
(open_manus) D:\myGit\OpenManus>pip install -r requirements.txt
Collecting pydantic~=2.10.4 (from -r requirements.txt (line 1))
  Using cached pydantic-2.10.6-py3-none-any.whl.metadata (30 kB)
Collecting openai~=1.58.1 (from -r requirements.txt (line 2))
  Using cached openai-1.58.1-py3-none-any.whl.metadata (27 kB)
Collecting tenacity~=9.0.0 (from -r requirements.txt (line 3))
  Using cached tenacity-9.0.0-py3-none-any.whl.metadata (1.2 kB)
Collecting pyyaml~=6.0.2 (from -r requirements.txt (line 4))
  Downloading PyYAML-6.0.2-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting loguru~=0.7.3 (from -r requirements.txt (line 5))
  Using cached loguru-0.7.3-py3-none-any.whl.metadata (22 kB)
Collecting numpy (from -r requirements.txt (line 6))
```

## 8、依赖安装完成后

需要配置一下调用大模型的 API：API 地址和 apikey

在 OpenManus/config 目录下，把原有的 config.example.toml 复制一个副本，并把副本文件名改成 config.toml



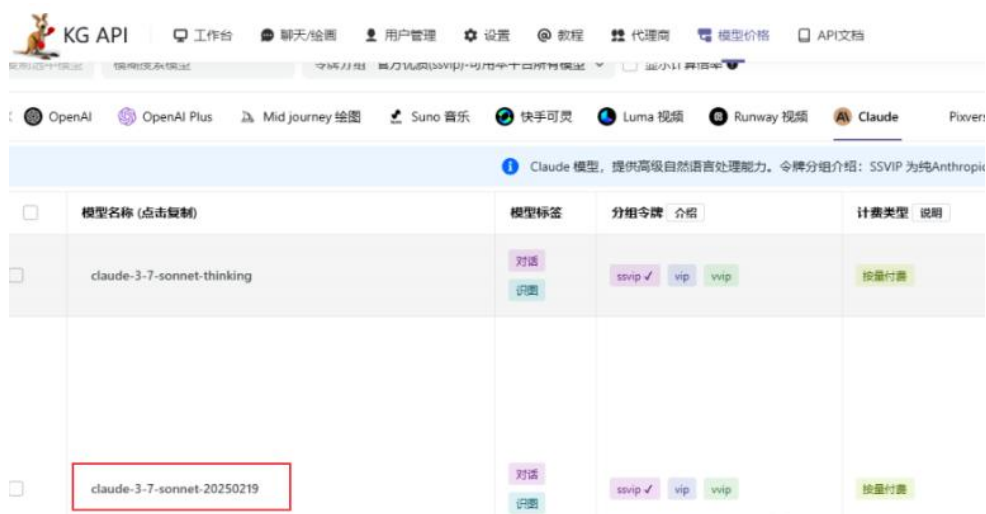
## 9、编辑 config.toml

```
D: > myGit > OpenManus > config > ⚙ config.toml
1  # Global LLM configuration
2  [llm]
3  model = "claude-3-5-sonnet"
4  base_url = "https://api.openai.com/v1"
5  api_key = "sk-..."
6  max_tokens = 4096
7  temperature = 0.0
8
9  # Optional configuration for specific LLM models
10 [llm.vision]
11 model = "claude-3-5-sonnet"
12 base_url = "https://api.openai.com/v1"
13 api key = "sk-..."
```

10、都说 manus 使用的是 Claude，所以，咱们使用 OpenManus 的时候也配置 Claude 才能更接近原生体验~

这个中转平台支持 Claude 最新模型 claude-3-7-sonnet-20250219 的 API

调用地址：<https://kg-api.cloud/>



The screenshot shows the KG API website interface. At the top, there's a navigation bar with links like '工作台', '聊天/绘画', '用户管理', '设置', '@ 教程', '代理商', '模型价格', and 'API文档'. Below the navigation bar, there's a section for 'Claude' models. A table lists the available models with columns for '模型名称 (点击复制)', '模型标签', '介绍令牌 介绍', and '计费类型 说明'.

模型名称 (点击复制)	模型标签	介绍令牌 介绍	计费类型 说明
claude-3-7-sonnet-thinking	对话 识图	ssvip ✓ vip vvip	按量付费
claude-3-7-sonnet-20250219	对话 识图	ssvip ✓ vip vvip	按量付费

11、到工作台->API 令牌->添加 API 令牌 (即可创建一个 apikey)

API 地址: <https://kg-api.cloud/v1> 我们修改后的配置如下:

```
# Global LLM configuration
[llm]
model = "gpt-4o"
base_url = "https://kg-api.cloud/v1"
api_key = "sk-eSdUogRl...5b7c74C42d"
max_tokens = 4096
temperature = 0.5

# Optional configuration for specific LLM models
[llm.vision]
model = "gpt-4o"
base_url = "https://kg-api.cloud/v1"
api_key = "sk-eSdUogRWcer...53b9b5b7c74C42d"
```

12、经过实测, 由于 Claude 没有 function call (工具调用) 功能和视觉能力, 所以, 目前只能用有 function call 功能和视觉能力大模型来跑。我这里选择了 gpt-4o, 希望官方尽快优化, 早日用上搭载 Claude 的完全体~

修改配置之后别忘了保存

最后我们在之前开启的控制台窗口, 输入如下指令 (启动 OpenManus)

```
python main.py
```

```
(open_manus) D:\myGit\OpenManus>python main.py
INFO [browser_use] BrowserUse logging setup complete with level info
INFO [root] Anonymized telemetry enabled. See https://docs.browser-use.com/development/telemetry for more information.
```

接下来就可以输入您的要求, 让 OpenManus 帮你干活啦

# OpenManus 本地配置 (高阶版)

---

简单了解了 OpenManus 主要设计之后，接下来我们本地部署起来。

首先，安装 Conda 工具，用于管理 Python 虚拟环境。可以参考文章：[大模型应用研发基础环境配置](#)

1、然后，创建 Python 虚拟环境（环境名：OpenManus），并激活环境：

```
# 创建虚拟环境 conda create -n OpenManus python=3.12
```

```
# 激活虚拟环境 conda activate OpenManus
```

2、最后，安装 OpenManus 依赖：

```
pip install -r requirements.txt
```

3、接下来，我们配置大模型 API，我们将使用 **QwQ-32B** 作为 OpenManus 底层大模型。先复制一份配置文件：[config/config.toml](#)

```
cp config/config.example.toml config/config.toml
```

4、然后，修改配置文件内容，把其中基础大模型配置成 **QwQ-32B** 基座模型，把视觉模型配置成 **Qwen-Max** 最新模型：

```
# 全局模型配置
```

```
[llm]
```

```
model = "qwen-max-latest"
```



```
base_url = "https://dashscope.aliyuncs.com/compatible-mode/v1"
```

```
api_key = "sk-..." # 替换为真实 API 密钥
```

```
max_tokens = 4096
```

```
temperature = 0.0
```

```
# 视觉模型配置
```

```
[llm.vision]
```

```
model = "qwen-max-latest"
```

```
base_url = "https://dashscope.aliyuncs.com/compatible-mode/v1"
```

```
api_key = "sk-..." # 替换为真实 API 密钥
```

5、注意：llm.model 不能使用 QwQ 模型（感觉有点翻车了），原因在于 QwQ 仅支持流式输出：

```
2025-03-07 21:00:32.259 | ERROR | app.llm:ask_tool:250 - API error: Error
```

```
code: 400 - {'error': {'code': 'invalid_parameter_error', 'param': None, 'message':
```

```
'This model only support stream mode, please enable the stream parameter to
```

```
access the model. ', 'type': 'invalid_request_error'}, 'id':
```

```
'chatcmpl-dbc286db-797d-9e3f-8cf1-b3adf15abc1a', 'request_id':
```

```
'dbc286db-797d-9e3f-8cf1-b3adf15abc1a'}
```

### 3. OpenManus 功能体验

---



配置完毕，就可以开始使用了： `python main.py`

```
>python main.pyINFO    [browser_use] BrowserUse logging setup complete
with level infoINFO    [root] Anonymized telemetry enabled. See
https://docs.browser-use.com/development/telemetry for more
information.Enter your prompt (or 'exit' to quit):
```

为了和 **Manus** 结果进行比较，我们输入提示词： `Identify 20 CRM companies in the market and gather their slogans along with their branding stories.`

从日志可以看出，总共分为 **30 个推理步骤**：

```
2025-03-07 20:58:43.891 | WARNING | __main__:main:15 - Processing your
request...2025-03-07 20:58:43.891 | INFO | app.agent.base:run:137 - Executing
step 1/30
```

执行完成之后，看到进行了总结：

### Summary:

We have now gathered the slogans and branding stories for the following CRM companies:

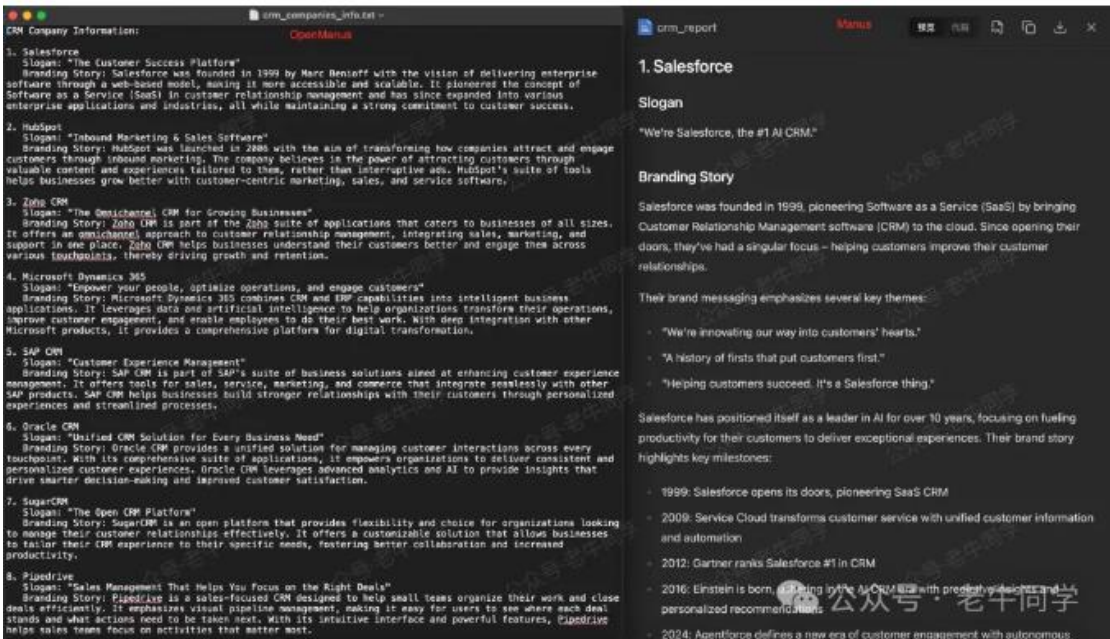
1. Salesforce2. HubSpot3. Zoho CRM4. Microsoft Dynamics 3655. SAP

- CRM6. Oracle
- CRM7. SugarCRM8. Pipedrive9. Freshsales10. Insightly11. Nimble12. Agile
- CRM13. Capsule CRM14. Copper (formerly ProsperWorks)15. Infusionsoft (by
- Keap)16. Zendesk Sell (formerly Base CRM)17. Really Simple Systems
- CRM18. Apptivo CRM19. Bitrix2420. Vtiger CRM

This information is saved in the file `crm\_companies\_info.txt`.

最终结果保存在本地文件：[crm\\_companies\\_info.txt](#)

打开本地结果文件，并且和 **Manus** 结果进行简单对比：



结尾：

Manus 本地部署教程，还在丰富和完善中，**更简单的 Manus 本地部署教程**会在后续发布，请留意发现 AI 的教程手册更新。

发现 AI 入口: <https://www.faxianai.com/>

满血版 Deepseek R1 入口: <https://ai.dangbei.com/chat>

最新版 Manus 教程, 请自行前往下载。