RAG课程安装文档

本课程主要涉及以下运行环境

- python软件包管理软件：anaconda3/miniforg3
- python环境：3.9和相关package （python nlpk torch langchain llamaindex jupyter-lab等）
- RAGFlow deepdoc
- 向量数据库：chroma/milvus
- 图数据库： neo4j
- 模型下载
- GPU环境安装(可选)
- ollama安装（可选）

【重点】安装文档和章节对应关系说明

文档中提供了windows和linux两个环境的安装，请根据自己的环境安装对应的部分（linux只要安装linux部分，windows只要安装windows部分）其他章节的具体的依赖：

- 基本环境：【一和二】
- 第3-4章：【一、二和六】（七和八可选，如果大模型是使用api）
- 第5章：需要安装向量数据库，对应【四】
- 第6章：有使用RAGflow 需要安装 【三】
- 第10章：使用到图数据库，需要安装neo4j， 需要安装【五】

一、python软件包管理软件：Conda

Conda是一个开源的包管理系统和环境管理器，主要用于简化Python及其他语言的软件包管理和项目环境管理。它最初由Anaconda公司开发，广泛用于数据科学、机器学习和科学计算等领域。

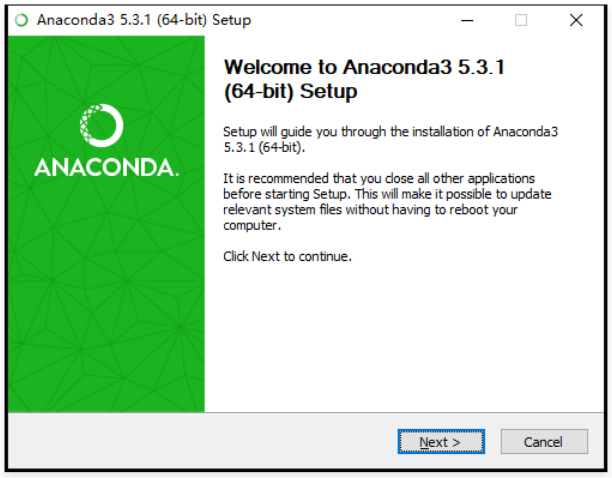
目前主流有anaconda3和miniforg3, anaconda3对于企业有版权限制，miniforg3为替代方案，对于个人用户无影响。

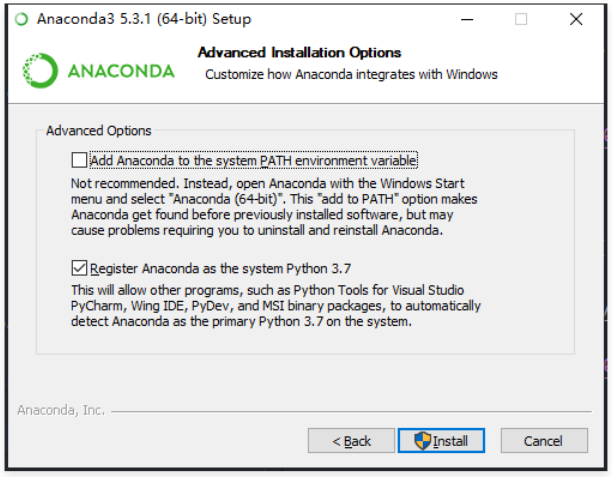
ananconda3 安装

- 软件包下载：https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/

Anaconda3-5.3.1-Linux-x86.sh	527.3 MiB	2018-11-20 04:00
Anaconda3-5.3.1-Linux-x86_64.sh	637.0 MiB	2018-11-20 04:00
Anaconda3-5.3.1-MacOSX-x86_64.pkg	634.0 MiB	2018-11-20 04:00
Anaconda3-5.3.1-MacOSX-x86_64.sh	543.7 MiB	2018-11-20 04:01
Anaconda3-5.3.1-Windows-x86.exe	509.5 MiB	2018-11-20 04:04
Anaconda3-5.3.1-Windows-x86_64.exe	632.5 MiB	2018-11-20 04:04

```
# linux安装
# Anaconda3-5.3.1-Linux-x86_64.sh

sh Anaconda3-5.3.1-Linux-x86_64.sh -b -u -p /root/anaconda3
export PATH="~/anaconda3/bin:$PATH"
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config --set show_channel_urls yes
conda update conda
conda upgrade --all

# windows安装
直接安装 Anaconda3-5.3.1-Windows-x86_64.exe
```



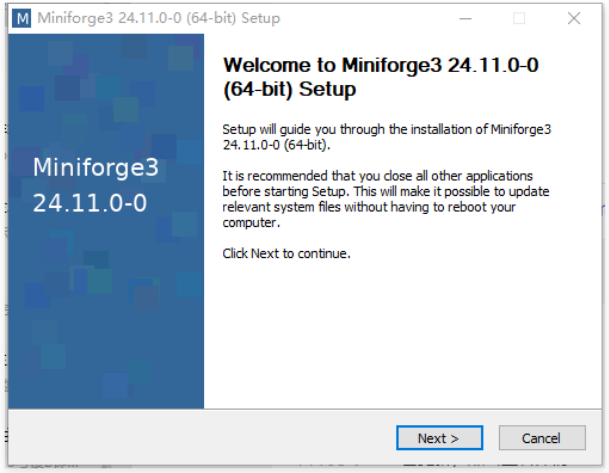


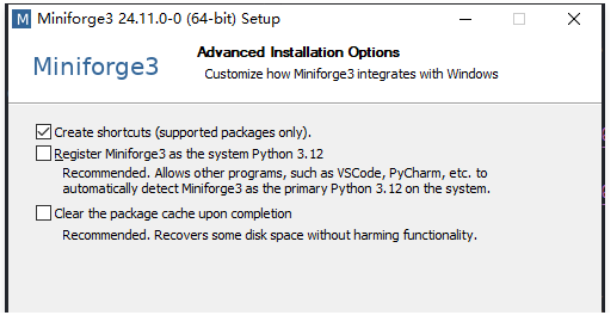
miniforge3安装（可选）

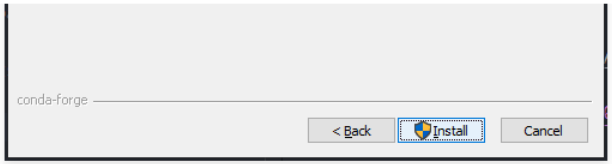
- 软件包下载：https://conda-forge.org/miniforge/

Miniforge3	Linux	aarch64	Miniforge3-24.11.0-0-Linux-aarch64.sh	76.5 MB	47cfd3caf···
		ppc64le	Miniforge3-24.11.0-0-Linux-ppc64le.sh	83.0 MB	877e39920···
		x86_64	Miniforge3-24.11.0-0-Linux-x86_64.sh	78.0 MB	5fa69e429···
	MacOSX	arm64	Miniforge3-24.11.0-0-MacOSX-arm64.sh	60.8 MB	3c7c115de···
		x86_64	Miniforge3-24.11.0-0-MacOSX-x86_64.sh	51.5 MB	1f0527ec1···
	Windows	x86_64	Miniforge3-24.11.0-0-Windows-x86_64.exe	70.6 MB	2ff523753···

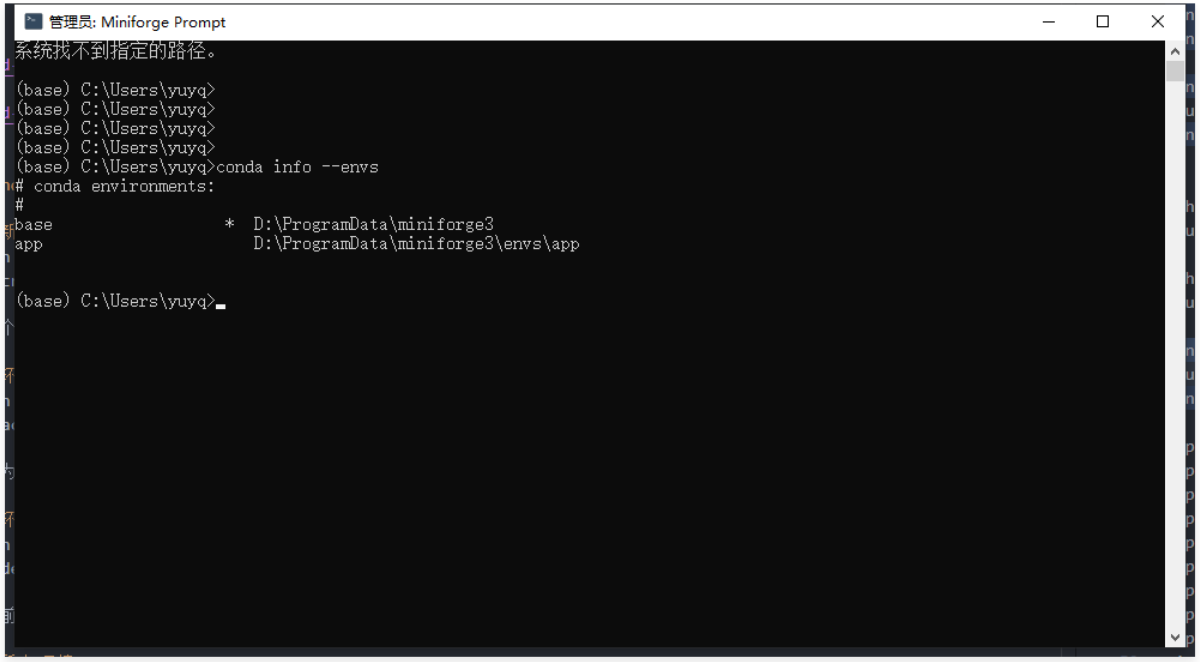
```
# liunx
sh Miniforge3-24.11.1-0-Linux-x86_64.sh -b -u -p /root/miniforge3
export PATH="~/miniforge3/bin:$PATH"

# windows
直接安装 Miniforge3-24.11.0-0-Windows-x86_64.exe
```







windows可以通miniforge prompt来使用conda



conda命令简单实用说明

1. **创建新环境**:

```
conda create --name myenv
```

创建一个名为 `myenv` 的新环境。

2. **激活环境**:

```
conda activate myenv
```

激活名为 `myenv` 的环境。

3. **停用环境**:

```
conda deactivate
```

停用当前活跃的Conda环境。

4. **列出所有环境**:

```
conda env list
```

或

```
conda info --envs
```

显示所有已创建的Conda环境。

5. **删除环境**:

```
conda remove --name myenv --all
```

删除名为`myenv`的环境。

6. **克隆环境**:

```
conda create --name newenv --clone myenv
```

克隆名为`myenv`的环境到`newenv`。

7. **安装包**:

```
conda install package-name
```

安装指定的包。

8. **更新包**:

```
conda update package-name
```

更新指定的包到最新版本。

9. **删除包**:

```
conda remove package-name
```

删除指定的包。

10. **列出已安装包**:

```
conda list
```

列出当前环境中已安装的所有包。

11. **搜索包**：

```
conda search package-name
```

搜索可用的包。

12. **导出环境**：

```
conda env export > environment.yml
```

将当前环境导出为 `environment.yml` 文件。

13. **从文件创建环境**：

```
conda env create -f environment.yml
```

从 `environment.yml` 文件创建环境。

二、python环境安装

通过conda来创建python环境，通过pip来安装课程的依赖的软件包

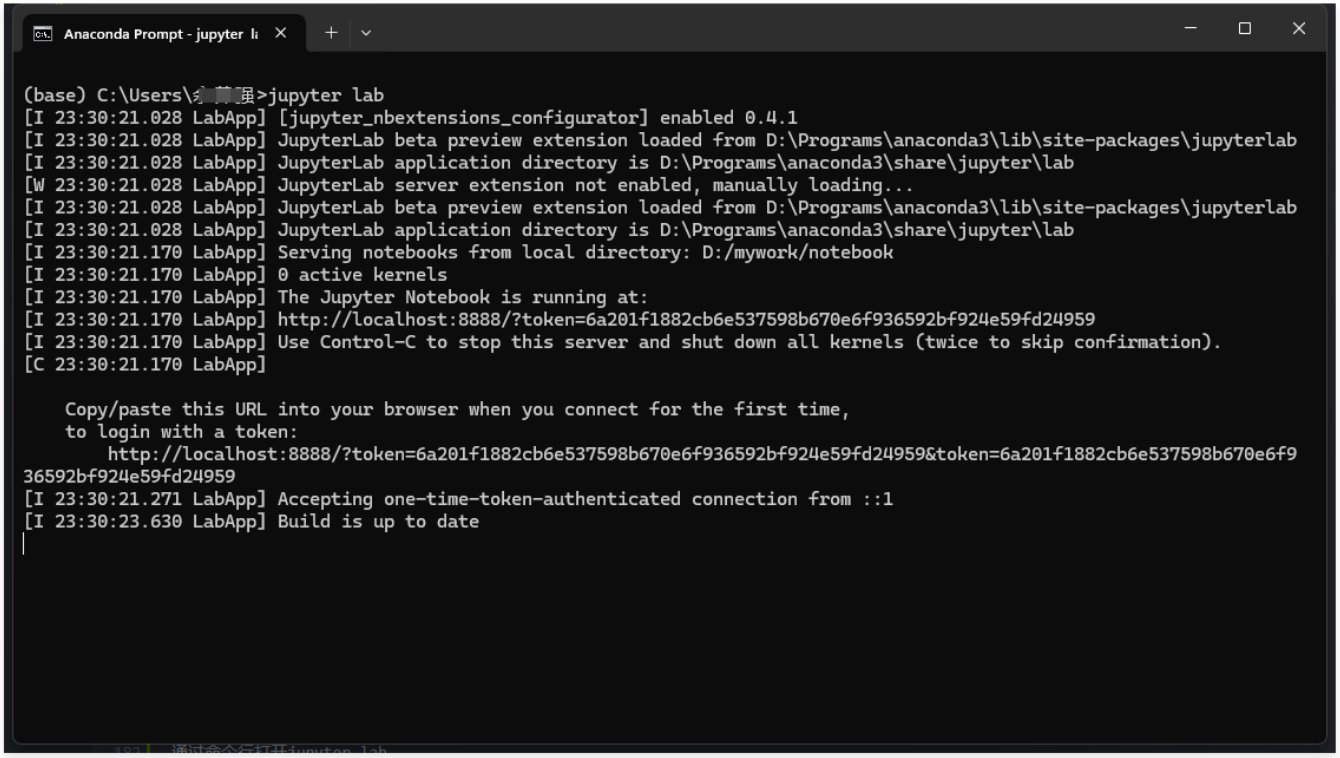
```
# 创建一个叫llm的环境 python 3.9的环境
conda create -n llm python=3.9 -y

# 激活
conda activate llm

pip install --no-cache-dir -i https://pypi.tuna.tsinghua.edu.cn/simple -r rag_requirements.txt
```

```
(base) C:\Users\yuyq>conda create -n llm python=3.9 -y
Channels:
 - defaults
 - conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
    current version: 24.7.1
    latest version: 24.11.3

Please update conda by running

    $ conda update -n base -c conda-forge conda



## Package Plan ##
```

```
  environment location: D:\ProgramData\miniforge3\envs\llm

  added / updated specs:
    - python=3.9


The following packages will be downloaded:

    package                            |            build
    -----------------------------------|-----------------
    ca-certificates-2024.11.26         |       haa95532_0         132 KB
    openssl-3.0.15                     |       h827c3e9_0         7.8 MB
    pip-24.2                           |   py39haa95532_0         2.4 MB
    python-3.9.21                      |       h8205438_1        19.6 MB
    setuptools-75.1.0                  |   py39haa95532_0         1.6 MB
    sqlite-3.45.3                      |       h2bbff1b_0         973 KB
    tzdata-2024b                       |       h04d1e81_0         115 KB
    vc-14.40                           |       haa95532_2          10 KB
    vs2015_runtime-14.42.34433         |       h9531ae6_2         1.2 MB
    wheel-0.44.0                       |   py39haa95532_0         137 KB
    ------------------------------------------------------------
                                           Total:        33.9 MB

The following NEW packages will be INSTALLED:

  ca-certificates    pkgs/main/win-64::ca-certificates-2024.11.26-haa95532_0
  openssl            pkgs/main/win-64::openssl-3.0.15-h827c3e9_0
  pip                pkgs/main/win-64::pip-24.2-py39haa95532_0
  python             pkgs/main/win-64::python-3.9.21-h8205438_1
  setuptools         pkgs/main/win-64::setuptools-75.1.0-py39haa95532_0
  sqlite             pkgs/main/win-64::sqlite-3.45.3-h2bbff1b_0
  tzdata             pkgs/main/noarch::tzdata-2024b-h04d1e81_0
  vc                 pkgs/main/win-64::vc-14.40-haa95532_2
  vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.42.34433-h9531ae6_2
  wheel              pkgs/main/win-64::wheel-0.44.0-py39haa95532_0



Downloading and Extracting Packages:
python-3.9.21        | 19.6 MB   | ##################################################################################1 |  99%
openssl-3.0.15       | 7.8 MB    | ################################################################################## | 100%
pip-24.2             | 2.4 MB    | ################################################################################## | 100%
setuptools-75.1.0    | 1.6 MB    | ################################################################################## | 100%
vs2015_runtime-14.42 | 1.2 MB    | ################################################################################## | 100%
sqlite-3.45.3        | 973 KB    | ################################################################################## | 100%
wheel-0.44.0         | 137 KB    | ################################################################################## | 100%
ca-certificates-2024 | 132 KB    | ################################################################################## | 100%
tzdata-2024b         | 115 KB    | ################################################################################## | 100%

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate llm
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

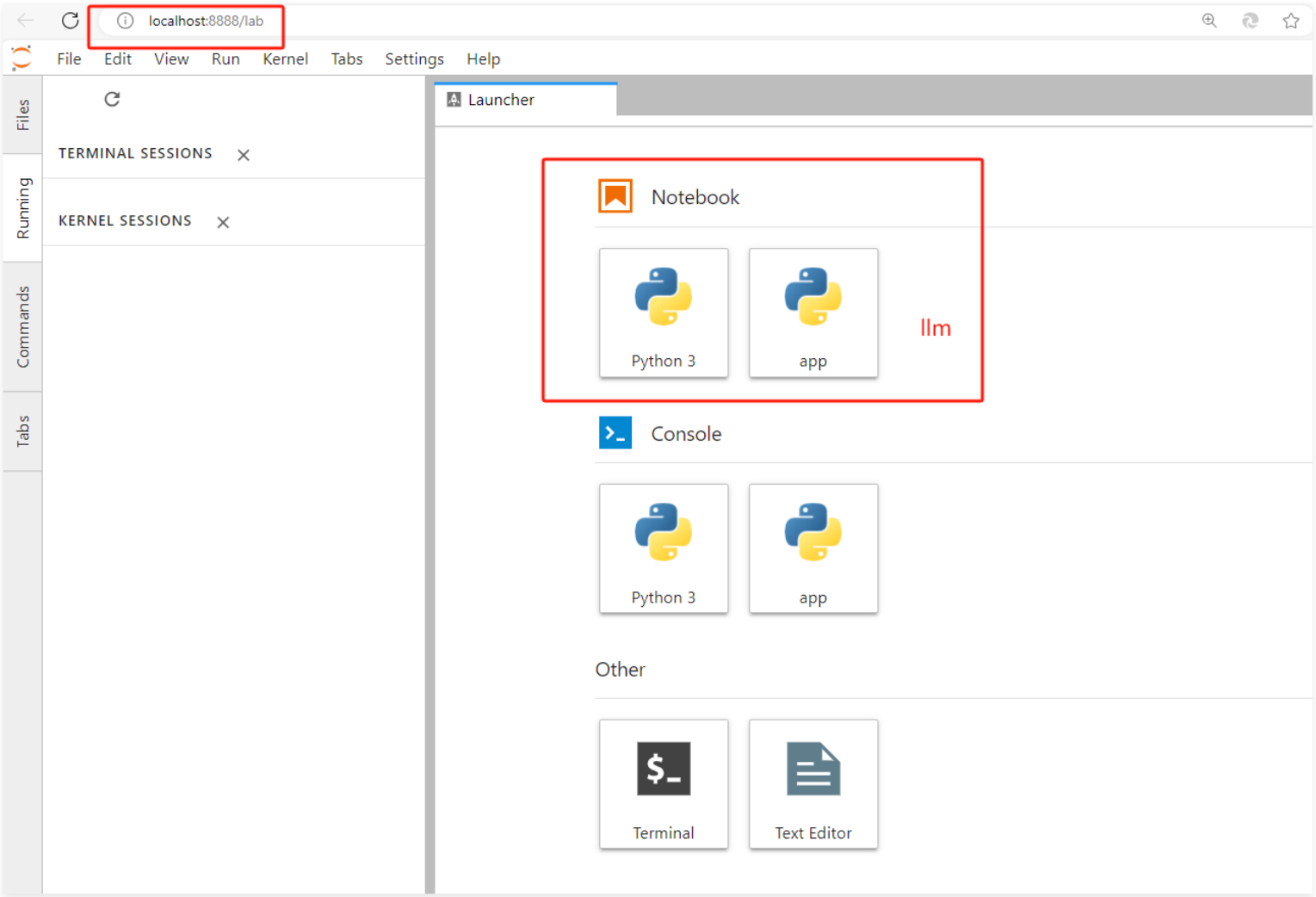
```
Installing collected packages: wcwidth, wcwidth, striprtf, StrEnum, sortedcontainers, sentencepiece, ruamel.base, pywin32, pytz, PyPika, pydub, pyclipper, pure-eval, ptyprocess, picklesbare, mpmath, monotonic, mmh3, lit, jieba, hanziconv, flatbuffers, filetype, executing, docx2txt, dirtyjson, crcmod, Chroma, ban
al, backcall, appdirs, antlr4-python3-runtime, addict, zipp, xxhash, XlsxWriter, wrapt, websockets, websocket-client, urllib3, ujson, tzdata, typing_extensions, traitlets, tornado, tomlkit, tomli, threadpoolctl, tenacity, tabulate, sympy, sparkai, soupsieve, snifflo, six, simplejson, shellingham, semantic-version, s
afetensors, ruff, ruamel.yaml.clib, roman-numbers, regex, RapidFuzz, pyzmq, PyYAML, python-multipart, python-magic, python-iso639, python-dotenv, pysbd, pyreadline3, pyproject_hooks, pypdfium2, pyparsing, PyMuPDFb, Pygments, pycryptodomex, pycryptodome, pycparser, pyasn1, pyarrow-hotfix, psutil, protobuf, prompt-too
lkit, process, primp, ppft, pox, portalocker, platformdirs, pillow, pexpect, parso, packaging, overrides, orjson, orderly-set, opentelemetry-util-http, olefile, oauthlib, numpy, networkx, nest-asyncio, mypy-extensions, multidict, mdurl, MarkupSafe, lxml, kiwisolver, jsonpointer, jsonpath-python, joblib, jmespath, jit
er, itsdangerous, idna, httptools, hll, greenlet, fsspec, frozenlist, fonttools, filelock, ffmpy, exceptiongroup, et-xmlfile, emoji, dnspython, distro, dill, decorator, debugpy, cycler, cssselect, colorama, cmake, charset-normalizer, chardet, certifi, cachetools, bcrypt, backoff, attrs, async-timeout, annotated-type
s, aiofiles, yarl, unstructured.pytesseract, typing-inspect, types-requests, tqdm, SQLAlchemy, shapely, sclpy, ruamel.yaml, rsa, rouge, requests, readability-lxml, rank-bm25, python-pptx, python-docx, python-dateutil, pytesseract, pypdf, PyMuPDF, PyExecJS, pydantic_core, pyasn1_modules, pyarrow, pulsar-client, pillo
w_heif, pi_heif, pdf2image, panel, opentelemetry-proto, openpyxl, opencv-python-headless, opencv-python, onnx, omegaconf, multiprocess, matplotlib-inline, marshmallow, markdown-it-py, Mako, lxml_html_clean, langdetect, jupyter_core, jsonpatch, Jinja2, jedi, interchange, importlib_resources, importlib-metadata, human
friendly, httpcore, grpcio, googleapis-common-protos, gekko, faiss-cpu, email_validator, Deprecated, deepdiff, contourpy, comm, cn2an, click, chroma-hnswlib, cffi, beautifulsoup4, asttokens, asgiref, anyio, aiosignal, xgboost, watchfiles, uvicorn, torch, tiktoken, tika, starlette, stack-data, scikit-learn, rich, req
uests-toolbelt, requests-oauthlib, python-oxmsg, pydantic, py2neo, posthog, pikepdf, pathos, pandas, opentelemetry-exporter-otlp-proto-common, opentelemetry-api, nltk, modelscope, matplotlib, langchainhub, jupyter_client, iopath, huggingface-hub, httpx, html_text, grpcio-tools, google-auth, environs, dataclasses-jso
n-speakeasy, dataclasses-json, cryptography, coloredlogs, build, alembic, aiohttp, unstructured-client, typer, translators, torchvision, tokenizers, pymilvus, pycocotools, pdfminer.six, opentelemetry-semantic-conventions, opentelemetry-instrumentation, openai, onnxruntime, llamaindex-py-client, langsmith, kubernetes
, ipython, gradio_client, frontend, dataset, aliyun-python-sdk-core, accelerate, unstructured, transformers, timm, pdfplumber, opentelemetry-sdk, opentelemetry-instrumentation-asgi, llama-index-legacy, llama-index-core, langchain-core, ipykernel, fastapi-cli, datasets, aliyun-python-sdk-kms, sentence-transformers, p
eft, oss2, opentelemetry-instrumentation-fastapi, opentelemetry-exporter-otlp-proto-grpc, llama-parse, llama-index-readers-file, llama-index-llms-openai, llama-index-indices-managed-llama-cloud, llama-index-embeddings-openai, layoutparser, langgraph, langchain-unstructured, langchain-text-splitters, langchain-openai
, fastapi, effdet, optimum, llama-index-readers-llama-parse, llama-index-multi-modal-llms-openai, llama-index-cli, llama-index-agent-openai, langchain, gradio, chromadb, auto_gptq, unstructured-inference, llama-index-vector-stores-chroma, llama-index-program-openai, langchain-community, langchain-chroma, ragas, llam
a-index-question-gen-openai, langchain-experimental, llama-index, spark-ai-python
Successfully installed Chroma-0.2.0 Deprecated-1.2.14 Jinja2-3.1.4 Mako-1.3.8 MarkupSafe-2.1.5 PyExecJS-1.5.1 PyMuPDF-1.24.5 PyMuPDFb-1.24.3 PyPika-0.48.9 PyYAML-6.0.1 Pygments-2.15.1 RapidFuzz-3.10.0 SQLAlchemy-1.4.54 StrEnum-0.4.15 XlsxWriter-3.2.0 accelerate-0.30.1 addict-2.4.0 aiofiles-23.2.1 aiohttp-3.9.5 aiosi
gnal-1.3.1 alembic-1.14.0 aliyun-python-sdk-core-2.15.2 aliyun-python-sdk-kms-2.16.5 annotated-types-0.7.0 antlr4-python3-runtime-4.9.3 anyio-4.4.0 appdirs-1.4.4 asgiref-3.8.1 asttokens-2.0.5 async-timeout-4.0.3 attrs-23.2.0 auto_gptq-0.7.1 backcall-0.2.0 backoff-2.2.1 banal-1.0.6 bcrypt-4.2.0 beautifulsoup4-4.12.3
```

通过命令行打开jupyter lab




```
conda activate llm

# 将llm环境添加到kernel中
conda install ipykernel
python -m ipykernel install --user --name llm

# 打开jupyter lab
jupyter lab --allow-root
```

```
# 创建一个notebook，kernel选择llm
```



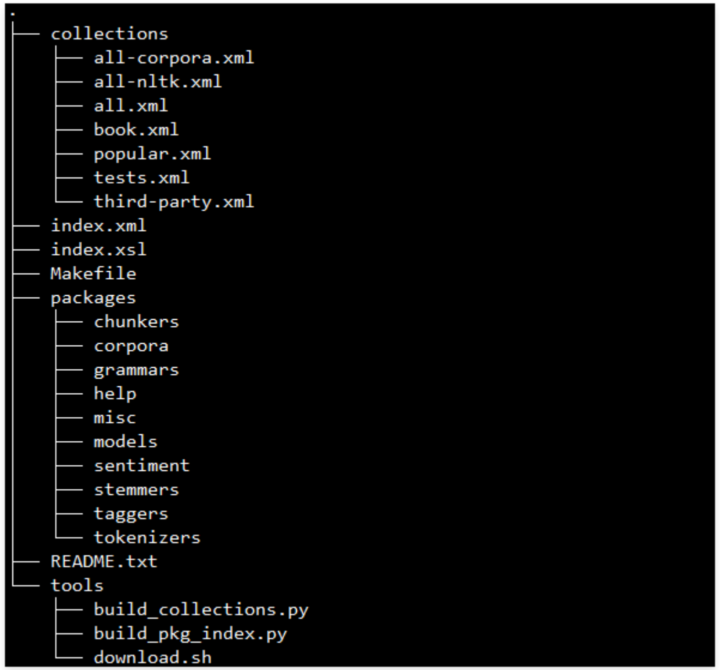
三、 RAGFlow

RagFlow 是一个用于构建和部署基于检索增强生成（Retrieval-Augmented Generation，RAG）的应用程序的开源框架。本课程中主要使用RAGflow的文档解析模块deepdoc

```
# 下载代码
git clone https://github.com/infiniflow/ragflow.git

# 依赖安装已经包含在第二部分的 rag_requirements.txt中
```

NTLK数据下载：ragflow需要使用nltk库，需要额外下载一些词表

```
import nltk
nltk.download()
```



```
.
├── collections
│   ├── all-corpora.xml
│   ├── all-nltk.xml
│   ├── all.xml
│   ├── book.xml
│   ├── popular.xml
│   ├── tests.xml
│   └── third-party.xml
├── index.xml
├── index.xsl
├── Makefile
├── packages
│   ├── chunkers
│   ├── corpora
│   ├── grammars
│   ├── help
│   ├── misc
│   ├── models
│   ├── sentiment
│   ├── stemmers
│   ├── taggers
│   └── tokenizers
├── README.txt
└── tools
    ├── build_collections.py
    ├── build_pkg_index.py
    └── download.sh
```

- https://github.com/nltk/nltk_data

四、向量数据库

chroma

- 安装和部署

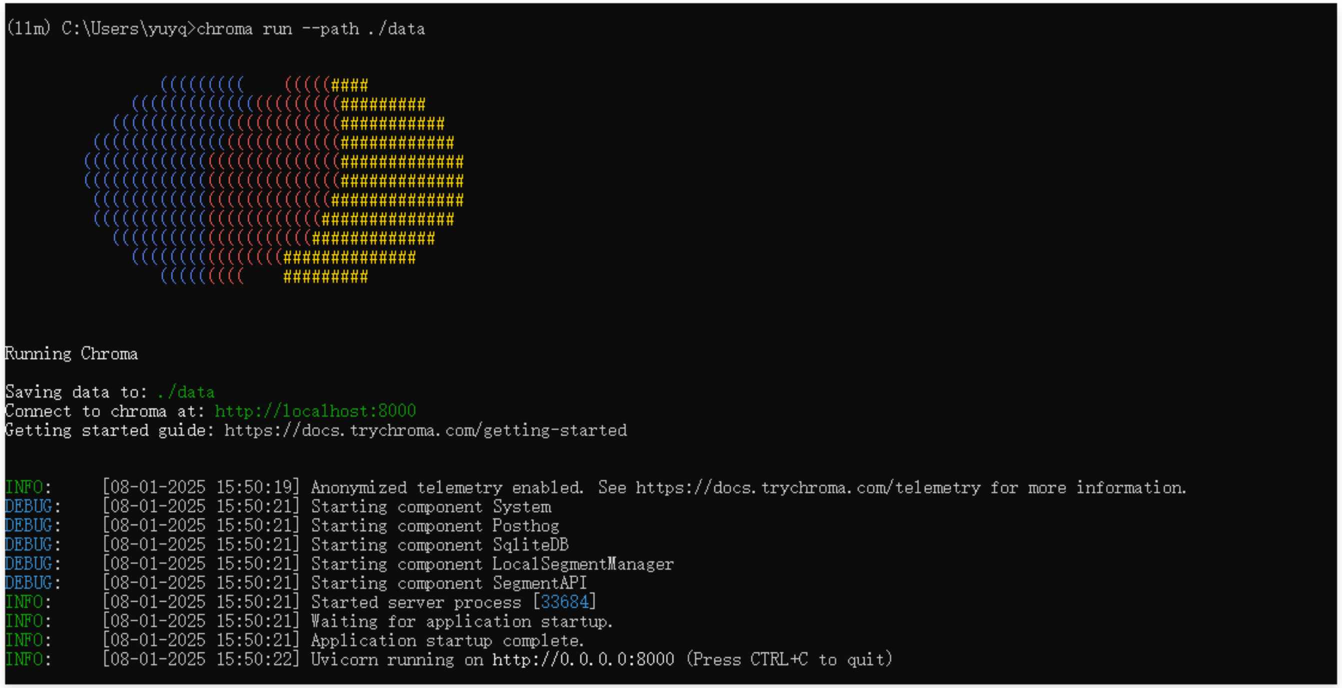
```
# 已经包含在第二部分的 rag_requirements.txt中
pip install chromadb
```

```
# 服务端部署
chroma run --path ./data

 Usage: chroma run [OPTIONS]
 Run a chroma server
╭─ Options ─────────────────────────────────────────────────────────────────────────────────────────────────────
```

```
│ --path          TEXT     The path to the file or directory. [default: ./chroma_data]
│ --host          TEXT     The host to listen to. Default: localhost [default: localhost]
│ --log-path      TEXT     The path to the log file. [default: chroma.log]
│ --port          INTEGER  The port to run the server on. [default: 8000]
│ --help                   Show this message and exit.
╰──────────────────────────────────────────────────────────────────────────────────────────────
```

```
客户端使用
import chromadb
chroma_client = chromadb.HttpClient(host='localhost', port=8000)
```

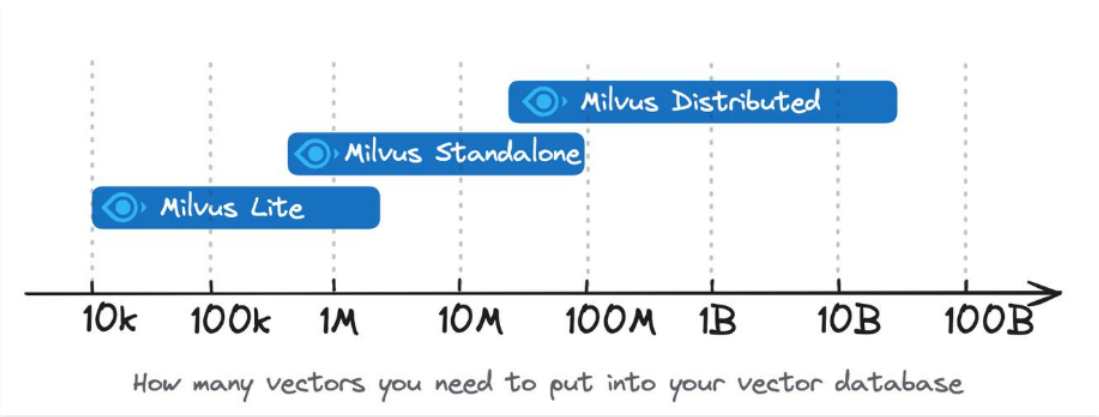
```
# 直接使用
import chromadb
client = chromadb.Client()
client = chromadb.PersistentClient(path="./data")
```



```
(llm) C:\Users\yuyq>chroma run --path ./data

                (((((((((    (((((####
             ((((((((((((((((((((((#########
           ((((((((((((((((((((((((###########
         ((((((((((((((((((((((((((############
        (((((((((((((((((((((((((((#############
        (((((((((((((((((((((((((((#############
         (((((((((((((((((((((((((##############
         ((((((((((((((((((((((((##############
           (((((((((((((((((((((#############
             ((((((((((((((((##############
                (((((((((    #########


Running Chroma

Saving data to: ./data
Connect to chroma at: http://localhost:8000
Getting started guide: https://docs.trychroma.com/getting-started


INFO:     [08-01-2025 15:50:19] Anonymized telemetry enabled. See https://docs.trychroma.com/telemetry for more information.
DEBUG:    [08-01-2025 15:50:21] Starting component System
DEBUG:    [08-01-2025 15:50:21] Starting component Posthog
DEBUG:    [08-01-2025 15:50:21] Starting component SqliteDB
DEBUG:    [08-01-2025 15:50:21] Starting component LocalSegmentManager
DEBUG:    [08-01-2025 15:50:21] Starting component SegmentAPI
INFO:     [08-01-2025 15:50:21] Started server process [33684]
INFO:     [08-01-2025 15:50:21] Waiting for application startup.
INFO:     [08-01-2025 15:50:21] Application startup complete.
INFO:     [08-01-2025 15:50:22] Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

milvus

milvus分为不同的版本：Milvus Lite, Milvus Standalone, and Milvus Distributed。 不同的数据规模和性能要求对应不同的版本



Feature	Milvus Lite	Milvus Standalone	Milvus Distributed
SDK / Client Lirary	Python gRPC	Python Go Java Node.js C# RESTful	Python Java Go Node.js C# RESTful
Data types	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON	Dense Vector Sparse Vector Binary Vector Boolean Integer Floating Point VarChar Array JSON
Search capabilities	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search	Vector Search (ANN Search) Metadata Filtering Range Search Scalar Query Get Entities by Primary Key Hybrid Search
CRUD operations	✔	✔	✔
Advanced data management	N/A	Access Control Partition Partition Key	Access Control Partition Partition Key Physical Resource Grouping
Consistency Levels	Strong	Strong Bounded Staleness Session	Strong Bounded Staleness Session

		Eventual	Eventual

- 参考文档

```
https://milvus.io/docs/install-overview.md
```

- milvus Lite安装

```
# 已经包含在第二部分的 rag_requirements.txt中
pip install pymilvus
```

```
# 本地使用

from pymilvus import MilvusClient
client = MilvusClient("./milvus_demo.db")

```

- milvus Standalone部署

  milvus Standalone只支持docker部署

  - https://milvus.io/docs/install_standalone-docker-compose.md

```
# 方式一
curl -sfL https://raw.githubusercontent.com/milvus-io/milvus/master/scripts/standalone_embed.sh -o standalone_embed.sh

bash standalone_embed.sh start
bash standalone_embed.sh stop
bash standalone_embed.sh delete
```

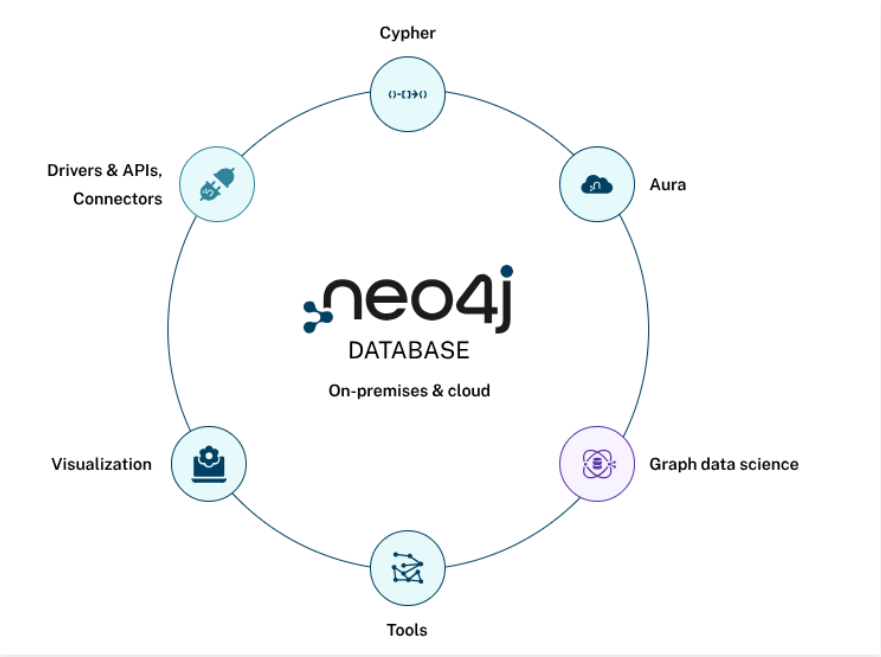
```
# 方式二
# 通过docker-compose
mkdir milvus_compose
cd milvus_compose
wget https://github.com/milvus-io/milvus/releases/download/v2.2.8/milvus-standalone-docker-compose.yml -O docker-compose.yml

sudo systemctl daemon-reload
sudo systemctl restart docker

# 启动服务
```

```
docker-compose up -d

# 安装 python接口库
pip install pymilvus
```

```
CONTAINER ID        IMAGE                                   COMMAND                  CREATED             STATUS                          PORTS                                                   NAMES
b7c262e22db2        milvusdb/milvus:v2.2.8                  "/tini -- milvus run… "   13 seconds ago      Up 12 seconds                   0.0.0.0:9091->9091/tcp, 0.0.0.0:19530->19530/tcp        milvus-standa
lone
0440bc4597bd        minio/minio:RELEASE.2023-03-20T20-16-18Z   "/usr/bin/docker-ent… "   14 seconds ago      Up 13 seconds (health: starting)   9000/tcp                                                milvus-minio
01494b27a4ad        quay.io/coreos/etcd:v3.5.5              "etcd -advertise-cli… "   14 seconds ago      Up 13 seconds                   2379-2380/tcp                                           milvus-etcd
```

五、图数据库

本课程使用的图数据库是neo4j 社区版本



neo4j为java开发的，服务端安装分为2个部分

- jdk安装
- neo4j软件安装

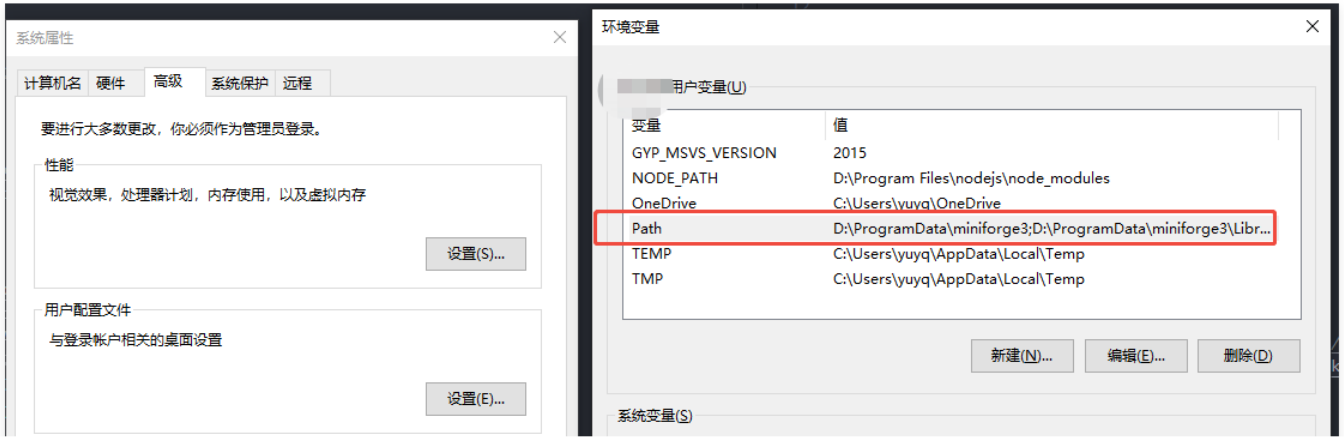
jdk安装

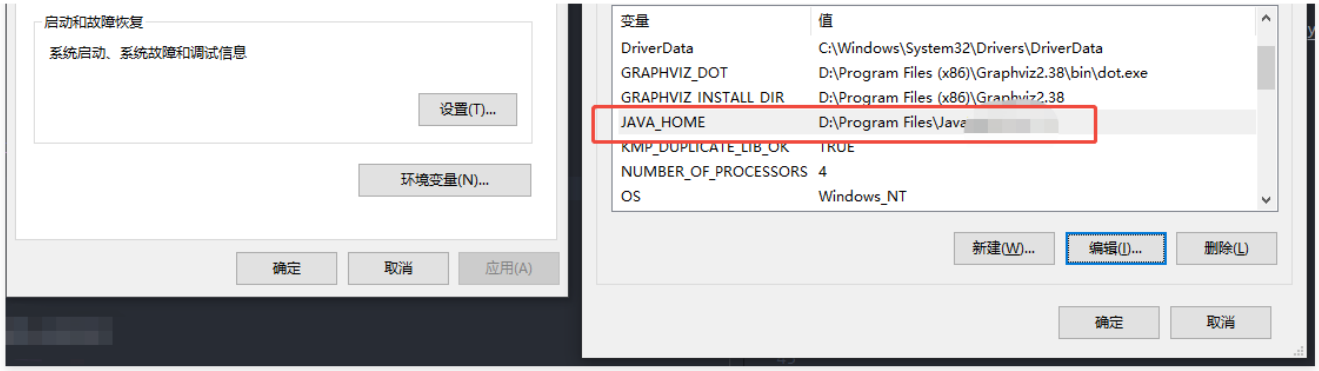
```
# jdk 版本  java 17.0.12 2024-07-16 LTS
```

```
# linux
wget https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.rpm
rpm -ivh jdk-17_linux-x64_bin.rpm


# windows
1.下载 https://www.oracle.com/java/technologies/downloads/?er=221886#java17-windows
2.点击安装 jdk-17.0.13_windows-x64_bin.exe
3.设置环境变量 JAVA_HOME  PATH=
# D:\Program Files\Java 为你的安装路径
JAVA_HOME D:\Program Files\Java\jdk17.0.13
PATH D:\Program Files\Java\jdk17.0.13\bin;D:\Program Files\Java\jdk17.0.13\jre\bin;

4.打开cmd 验证下
java -version
```

Linux **macOS** **Windows**

Product/file description	File size	Download
x64 Compressed Archive	172.79 MB	jdk-17.0.13_windows-x64_bin.zip
x64 Installer	153.98 MB	jdk-17.0.13_windows-x64_bin.exe
x64 MSI Installer	152.73 MB	jdk-17.0.13_windows-x64_bin.msi

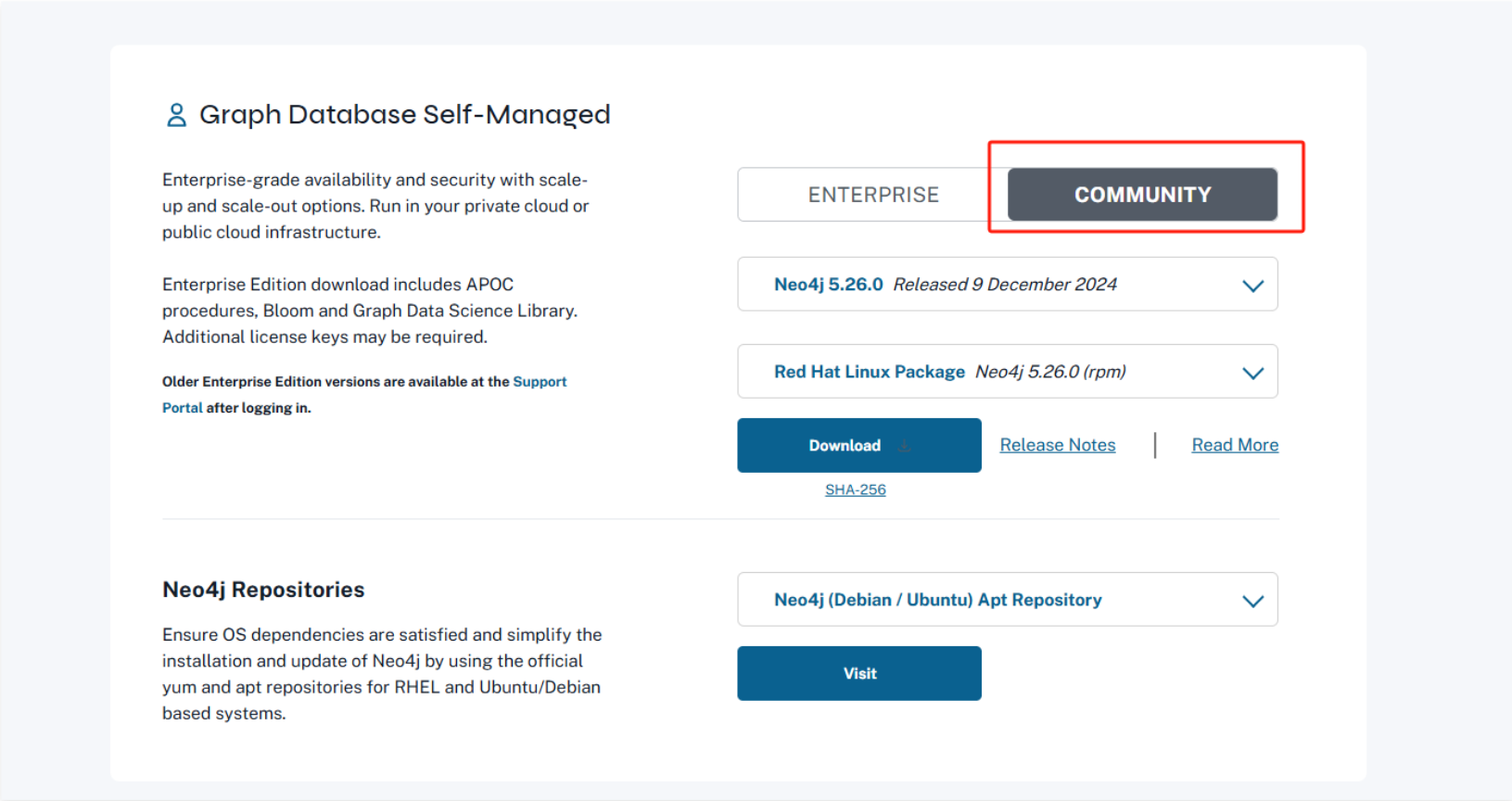


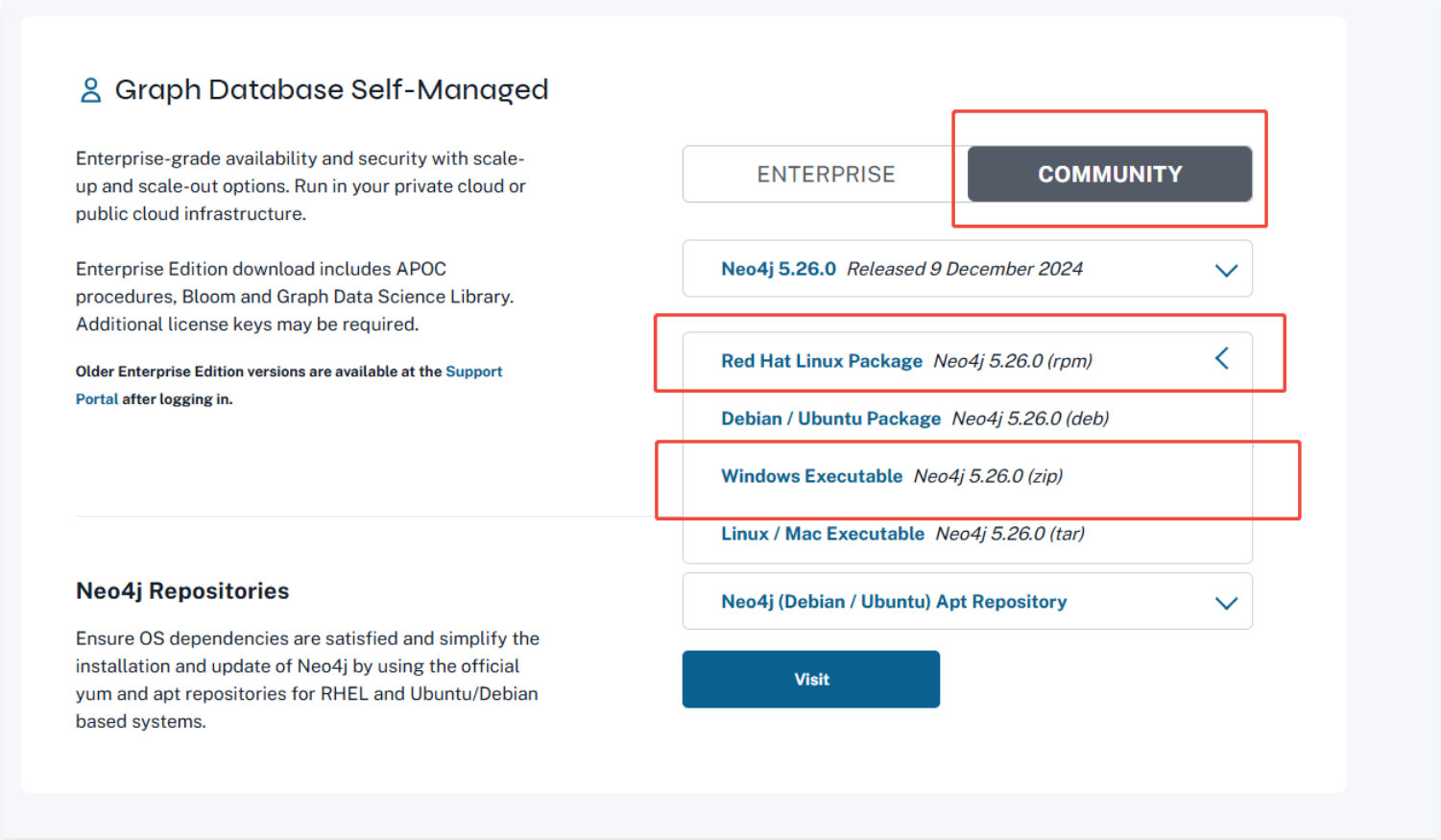


neo4j安装

下载路径

- https://neo4j.com/deployment-center/

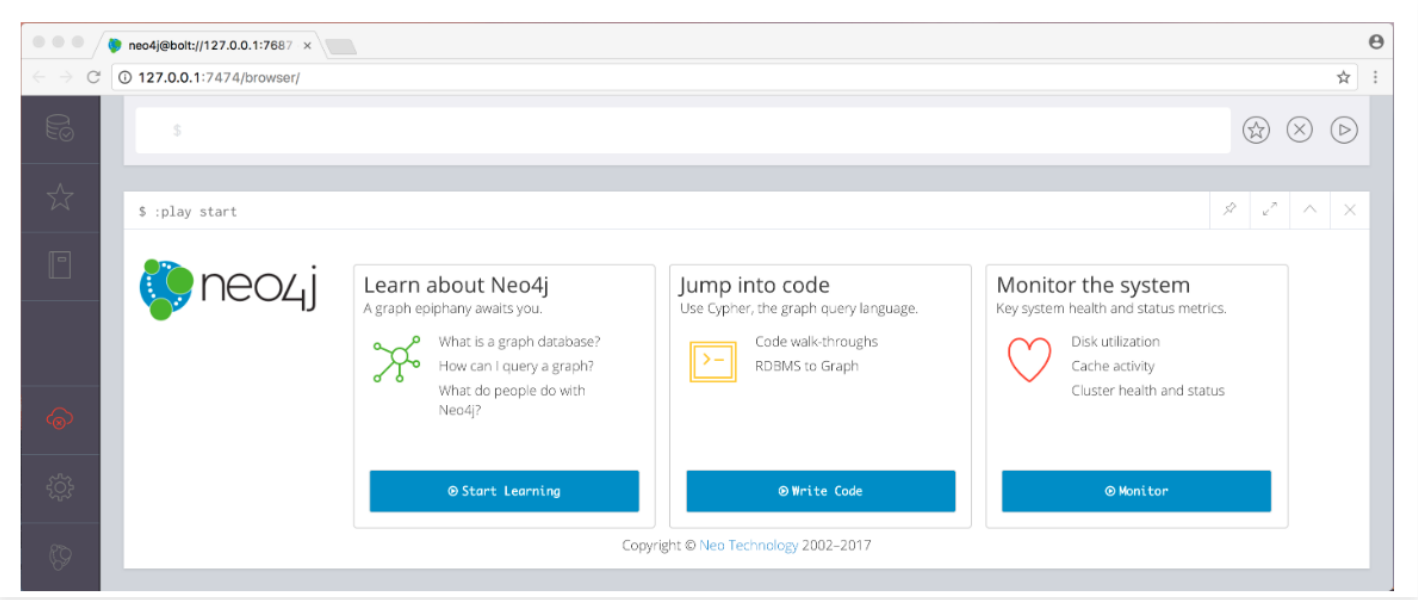





```
# linux安装
# 方式一: yum 安装
rpm --import https://debian.neo4j.com/neotechnology.gpg.key
cat << EOF >  /etc/yum.repos.d/neo4j.repo
[neo4j]
name=Neo4j RPM Repository
baseurl=https://yum.neo4j.com/stable/5
enabled=1
gpgcheck=1
EOF
yum install neo4j-5.26.0

# 方式二: rpm安装
curl -O https://dist.neo4j.org/rpm/neo4j-5.26.0-1.noarch.rpm
curl -O https://dist.neo4j.org/rpm/neo4j-enterprise-5.26.0-1.noarch.rpm


# neo4j启动和停止
```

```
neo4j start
neo4j stop
```

```
# windows 安装
1. 下载https://dist.neo4j.org/neo4j-community-5.26.0-windows.zip
2. 解压neo4j-community-5.26.0-windows.zip 到某个路径 比如D:\neo4j\
3. 设置环境变量 PATH D:\neo4j\bin
4. 打开cmd启动服务： neo4j install-service.

# 访问 http://localhost:7474
# username 'neo4j' and password 'neo4j'
```



参考

- https://neo4j.com/docs/operations-manual/current/installation/windows/
- https://neo4j.com/docs/operations-manual/current/installation/linux/rpm/#linux-rpm-install-standard

六、模型下载

本课程涉及到模型都主要来自开源模型（使用名字搜索即可）

- modelscope: https://modelscope.cn/models
- huggingface: https://huggingface.co/models

ps: huggingface国内访问不了，可以使用镜像站访问： https://hf-mirror.com/

七、GPU环境安装(可选)

这里的GPU特指nvidia GPU，安装GPU环境主要包括：

- GPU显卡驱动
- GPU开发环境：cuda和cdnn （CUDA是NVIDIA推出的用于自家GPU的并行计算框架；cuDNN是一个SDK，是一个专门用于神经网络的加速包）

GPU显卡驱动

根据你购买的nvidia显卡的型号，选择合适的驱动

- https://www.nvidia.cn/drivers/lookup/

```
sh NVIDIA-Linux-x86_64-550.142.run

# 测试是否安装成功

nvidia-smi
```

```
Thu Jan  9 16:46:26 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ...  Off  | 00000000:06:00.0 Off |                  N/A |
| 40%   35C    P0    51W / 250W |      0MiB / 11264MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA GeForce ...  Off  | 00000000:81:00.0 Off |                  N/A |
| 36%   35C    P0    24W / 250W |      0MiB / 11264MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

GPU开发环境

以linux centos 7例子为例子

```
# cuda  https://developer.nvidia.com/cuda-toolkit-archive
# https://developer.nvidia.com/cuda-11-8-0-download-archive?target_os=Linux&target_arch=x86_64&Distribution=CentOS&target_version=7&target_type=rpm_local

wget https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/cuda-repo-rhel7-11-8-local-11.8.0_520.61.05-1.x86_64.rpm
sudo rpm -i cuda-repo-rhel7-11-8-local-11.8.0_520.61.05-1.x86_64.rpm
sudo yum clean all
sudo yum -y install nvidia-driver-latest-dkms
sudo yum -y install cuda


# cudnn https://developer.nvidia.com/rdp/cudnn-archive
# 需要注册
wget https://developer.nvidia.com/downloads/compute/cudnn/secure/8.9.5/local_installers/11.x/cudnn-linux-x86_64-8.9.5.30_cuda11-archive.tar.xz/
```

```
# 解压会得到cuda目录，复制到已经安装好的cuda目录
cp cuda/include/cudnn.h /usr/local/cuda/include
cp cuda/lib64/libcudnn* /usr/local/cuda/lib64

# 必要情况下可以设置环境变量
export CUDA_HOME=/usr/local/cuda-11.8
```

```
root[redacted]:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

Download cuDNN v8.9.5 (October 27th, 2023), for CUDA 11.x

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

- Local Installer for Windows (Zip)
- Local Installer for Linux x86_64 (Tar)
- Local Installer for Linux PPC (Tar)
- Local Installer for Linux SBSA (Tar)
- Local Installer for Debian 11 (Deb)
- Local Installer for Ubuntu18.04 x86_64 (Deb)
- Local Installer for Ubuntu20.04 x86_64 (Deb)
- Local Installer for Ubuntu22.04 x86_64 (Deb)
- Local Installer for Ubuntu20.04 aarch64sbsa (Deb)
- Local Installer for Ubuntu22.04 aarch64sbsa (Deb)
- Local Installer for Ubuntu20.04 cross-sbsa (Deb)
- Local Installer for Ubuntu22.04 cross-sbsa (Deb)

windows

NVIDIA Studio 驱动程序 566.36 | Windows 10 64-bit

驱动主页 > NVIDIA GeForce RTX 4090 D | Windows 10 64-bit > **NVIDIA Studio 驱动程序**

驱动版本:	566.36 | **WHQL**
发布日期:	Tue Dec 10, 2024

自动更新您的驱动

操作系统:	Windows 10 64-bit, Windows 11
语言:	Chinese (Simplified)
文件大小:	732.88 MB

*此下载包括 NVIDIA 显卡驱动以及另外安装 GeForce Experience 应用的选项。有关软件使用的详细信息，请分别参阅 NVIDIA GeForce 软件许可证和 GeForce Experience 软件许可证。

下载

自动检测驱动并保持更新。截取视频、屏幕截图并与好友分享。NVIDIA app 是 GeForce 显卡的重要搭档。

NVIDIA App

CUDA Toolkit 11.8 Downloads

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the CUDA EULA.

Operating System	Linux, Windows
Architecture	x86_64
Version	10, 11, Server 2016, Server 2019, Server 2022
Installer Type	exe (local), exe (network)

Download Installer for Windows 10 x86_64

The base installer is available for download below.

> Base Installer — Download (3.0 GB)

Installation Instructions:

1. Double click cuda_11.8.0_522.06_windows.exe
2. Follow on-screen prompts

The checksums for the installer and patches can be found in Installer Checksums.
For further information, see the Installation Guide for Microsoft Windows and the CUDA Quick Start Guide.

八、ollama安装（可选）

Ollama是一个集成了多种大型语言模型的工具，它支持模型的部署、运行以及API的整合和调用

- https://ollama.com/download/linux
- 安装Ollama：

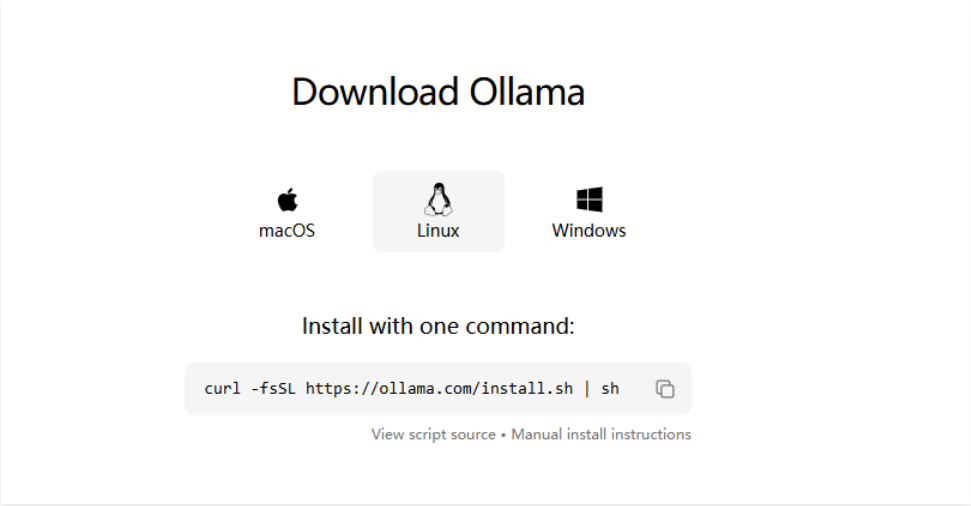
```
curl -fsSL https://ollama.com/install.sh | sh
```

- 验证安装：

```
# 输入来验证安装是否成功。
ollama --version
```

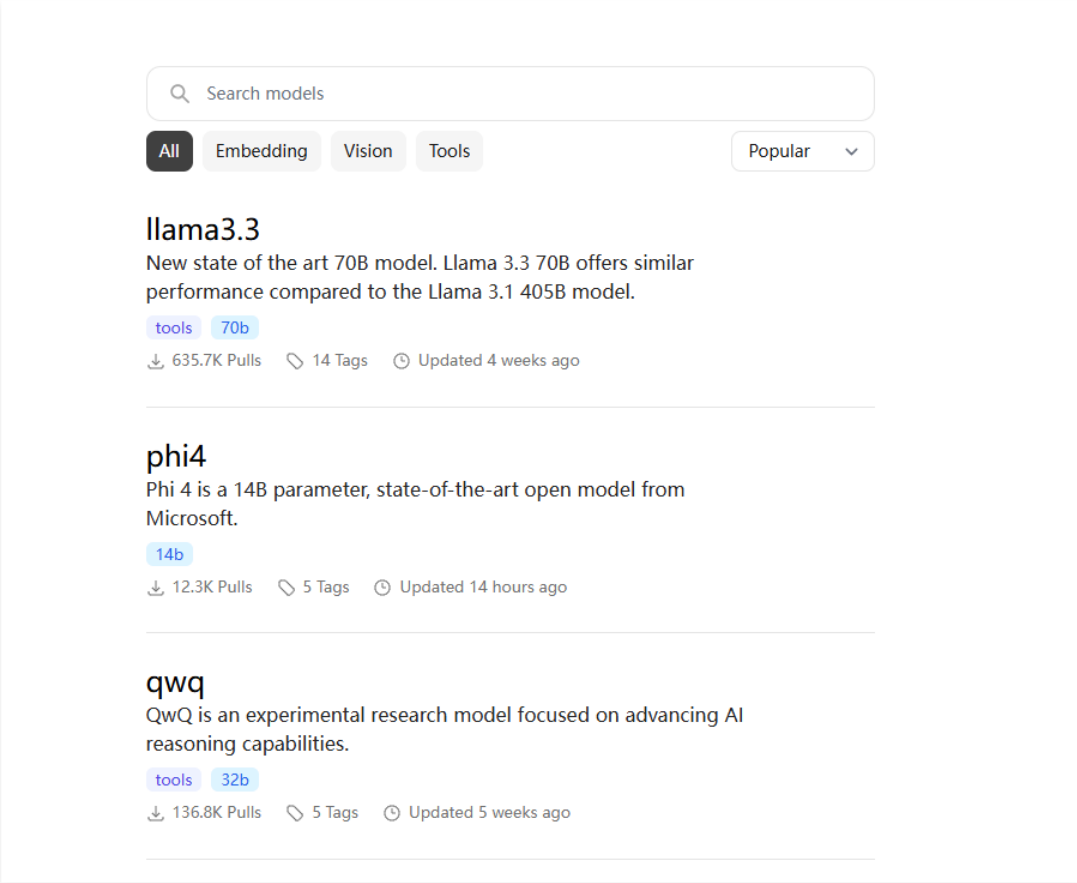
- 使用

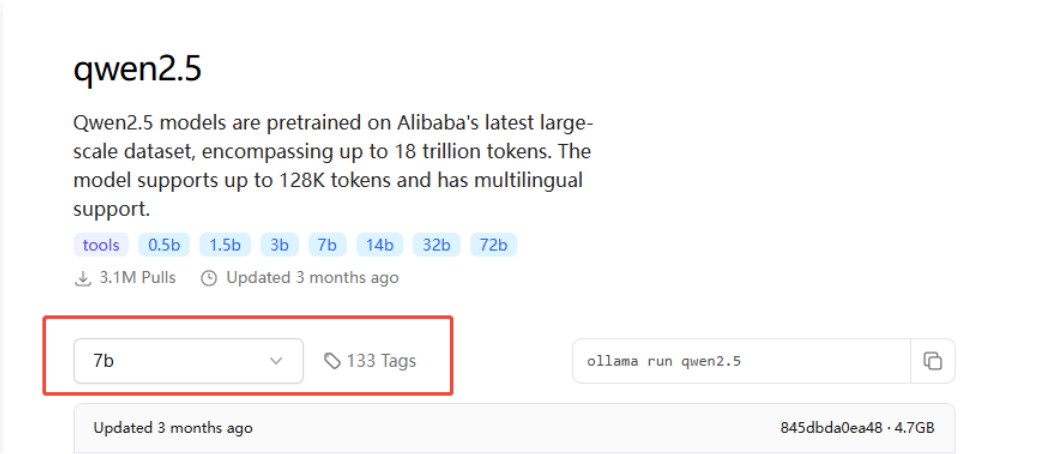
```
# 启动服务
ollama serve

# 运行模型

ollama run qwen2:70b
```



ollama 支持最流行的模型

- https://ollama.com/search





model	arch qwen2 · parameters 7.62B · quantization Q4_K_M	4.7GB
system	You are Qwen, created by Alibaba Cloud. You are a helpful assist…	68B
template	{{- if .Messages }} {{- if or .System .Tools }}<|im_start|>syste…	1.5kB
license	Apache License Version 2.0, January 200	11kB

九、说明

由于软件安装系统和版本迭代会随着时间发生变化，在安装过程中如遇到问题可以具体问题具体分析。

下一节：【文档】课程机器配置要求说明

下一节