如何在本地搭建一个大语言模型

本文件为GeekHour的【十分钟部署一个本地大模型】视频的配套教程，
主要介绍如何在本地搭建一个大型语言模型服务。

视频地址：

哔哩哔哩

YouTube

抖音

教程中的所有资料、笔记和文档可以从以下位置得到：

【百度网盘】
https://pan.baidu.com/s/1Q4wVSwKx_qbWUMyDL5rwGA?pwd=7g54 提取码:7g54

【夸克网盘】https://pan.quark.cn/s/b78c37513723提取码：SU5w

Step 1: 安装Ollama

Ollama是一个开源的大型语言模型服务工具，
可以快速在本地安装和运行大模型。
通过一条命令就可以轻松启动和运行各种开源的大型语言模型。
提供了一个简洁易用的命令行界面，专为构建大型语言模型应用而设计。

1.1 安装Ollama

Ollama支持MacOS、Linux和Windows三大主流操作系统，
MacOS和Windows从官网下载自己系统对应的版本安装即可。

Linux系统下可以直接复制以下命令到终端执行：

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

但是一般会因为网络问题下载不了，可以直接下载一个离线的安装包，
或者从我们提供的资源里面找到对应的安装包来直接解压安装。

```
# 下载安装包
curl -L https://ollama.com/download/ollama-linux-amd64.tgz -o ollama-linux-amd64.tgz
# 解压安装
sudo tar -C /usr -xzf ollama-linux-amd64.tgz
```

1.2 启动Ollama

```
ollama serve
```

```
# 设置开机启动
sudo systemctl daemon-reload
sudo systemctl enable ollama
```

1.3 安装大模型

执行下面的命令就可以下载一个Llama3.2的大模型并且运行起来：

```
ollama run llama3.2
```

由于网络原因，可能会下载不了，可以直接下载我们提供的大模型文件，
下载网盘中的Modelfile和模型文件（Meta-Llama-3-8B-Instruct.Q8_0.gguf），
并把它们放在同一个目录里面，然后CMD进入到这个目录下，
执行下面的命令就可以从模型文件直接创建和运行一个Llama3模型：

```
ollama create llama3 -f ./Modelfile
```

如果想要使用其他的模型，也可以使用同样的方法来下载和运行。
需要先到huggingface或者其他的模型仓库里面找到对应的gguf模型文件，
然后编辑一个Modelfile文件写入下面的内容：

```
FROM ./model_file.gguf
```

然后使用下面的命令来创建一个模型：

```
ollama create model_name -f ./Modelfile
```

1.4 Ollama常用命令

输入ollama之后回车，就会提示常用的命令和用法：

```
ollama serve        # 启动Ollama服务
ollama create       # 从模型文件创建一个模型
# e.g: ollama create example -f ./vicuna-33b.gguf
ollama show         # 显示模型基本信息
ollama run          # 运行一个模型
# e.g: ollama run llama3.2
ollama stop         # 停止一个正在运行的模型
ollama pull         # 从仓库拉取一个模型
ollama push         # 将一个模型推送至仓库
ollama list         # 显示模型清单
ollama ps           # 显示正在运行的模型
ollama cp           # 复制一个模型
ollama rm           # 删除一个模型
ollama help         # 显示某个命令的帮助信息
# e.g: ollama help run
```

对于某个命令的详细使用方法，
可以使用ollama help 后面加上命令名称来查看。

Step 2: 安装WebUI界面

由于Ollama是一个命令行工具，如果想要更加直观的管理和使用模型，
可以安装一个WebUI界面，比如MaxKB或者OpenWebUI（二者选一个安装即可）。

MaxKB是一个基于大语言模型和RAG的知识库系统。
安装部署文档在这里
社区版下载地址在这里

2.1 安装MaxKB（使用Docker安装）

网络环境好的话可以直接拉取官方镜像来安装：

```
docker run -d --name=maxkb -p 8080:8080 -v ~/.maxkb:/var/lib/postgresql/data 1panel/maxkb
```

网络环境不好的话可以下载我们提供的镜像文件，
然后使用下面的命令来安装：

```
# 导入镜像
docker load < maxkb.tar
# 启动容器
docker run -d --name=maxkb -p 8080:8080 -v ~/.maxkb:/var/lib/postgresql/data 1panel/maxkb
```

2.2 安装MaxKB(离线方式)

2.2.1 解压安装包

```
tar -zxvf maxkb-v1.6.1-offline.tar.gz
```

2.2.2 安装前配置（可选）

MaxKB 安装目录、服务运行端口、数据库配置等信息可在安装包中的 install.conf 文件进行配置。

```
## 安装目录
MAXKB_BASE=/opt
## Service 端口
MAXKB_PORT=8080
## docker 网段设置
MAXKB_DOCKER_SUBNET=172.19.0.0/16
# 数据库配置
## 是否使用外部数据库
MAXKB_EXTERNAL_PGSQL=false
## 数据库地址
MAXKB_PGSQL_HOST=pgsql
## 数据库端口
MAXKB_PGSQL_PORT=5432
```

```
## 数据库库名
MAXKB_PGSQL_DB=maxkb
## 数据库用户名
MAXKB_PGSQL_USER=root
## 数据库密码
MAXKB_PGSQL_PASSWORD=Password123@postgres
```

2.2.3 安装

```
# 进入安装包解压缩后目录
cd maxkb-v1.2.0-offline

# 执行安装命令
bash install.sh
```

2.2.4 登录访问

安装成功后，通过浏览器访问地址 http://localhost:8080，使用默认的管理员用户和密码登录MaxKB。

```
用户名：admin
默认密码：MaxKB@123..
```

2.3 安装OpenWebUI

```
# 如果Ollama已经安装在你本地电脑上，可以直接运行下面的命令来启动OpenWebUI
docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-
webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-
webui:main

# 如果本地电脑上有Nvidia的GPU，可以使用下面的命令来启动OpenWebUI
docker run -d -p 3000:8080 --gpus all --add-host=host.docker.internal:host-gateway -v
open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-
webui:cuda
```

同样的，如果由于网络原因无法顺利拉取镜像，可以直接下载我们提供的镜像文件，
然后使用下面的命令来安装：

```
# 导入镜像
docker load < open-webui.tar
# 然后再使用上面的命令来启动容器
```

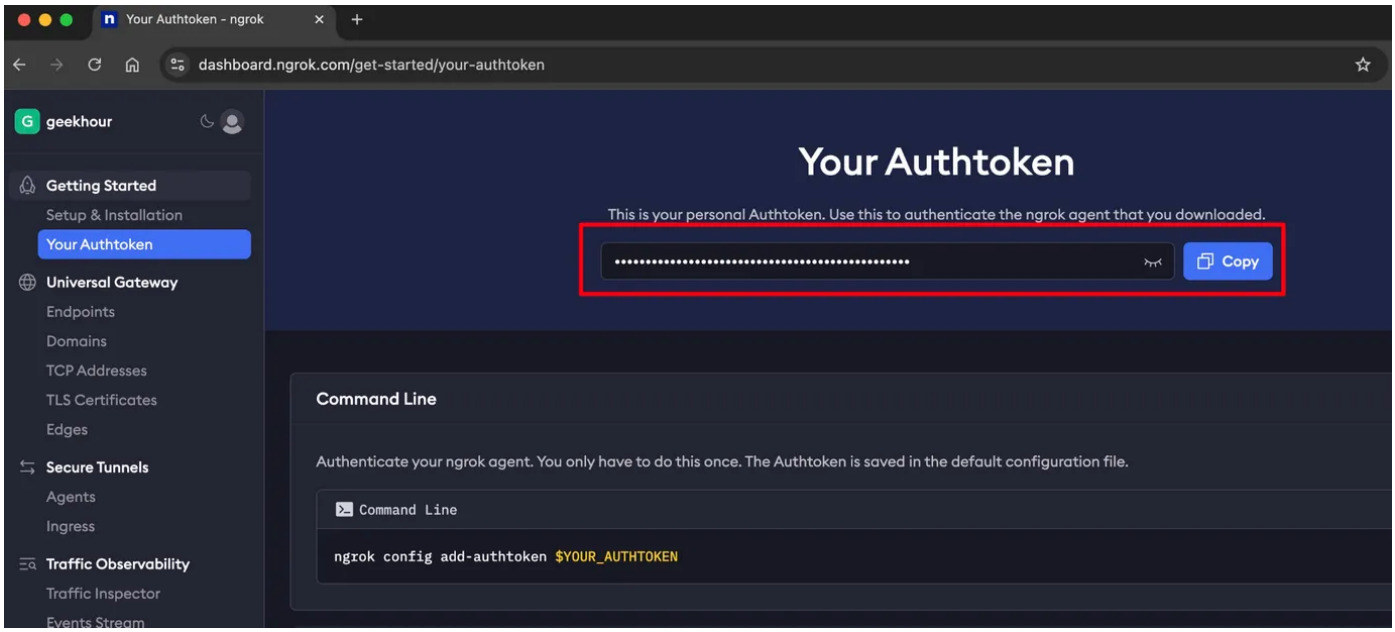
安装完成之后，可以通过浏览器访问 http://localhost:3000 来使用OpenWebUI。

Step 3: 配置ngrok支持公网访问（可选）

这一步并不是必须的，如果只想要在自己电脑本地访问的话，那么到上面就可以了。
但是如果想要把这个应用发布到公网上，让你的好朋友或者你公司的客户也一起来使用的话，那么需要配置一下ngrok。

ngrok是一个内网穿透工具，可以将本地的应用映射到公网上，下载地址在这里。

安装完成之后需要注册一个账号，注册的过程中可能会需要使用到Authenticator软件来执行双重身份验证，
然后在控制台里面找到自己的authtoken，



复制一下Authtoken，然后回到命令行终端，
执行下面的命令来设置一下：

```
ngrok config add-authtoken <your_auth_token>
```

设置完成之后就可以使用下面的命令来映射公网地址到本地：

```
ngrok http 8080
```

ngrok启动之后就可以通过下面的公网地址访问到本地搭建的大模型了。