

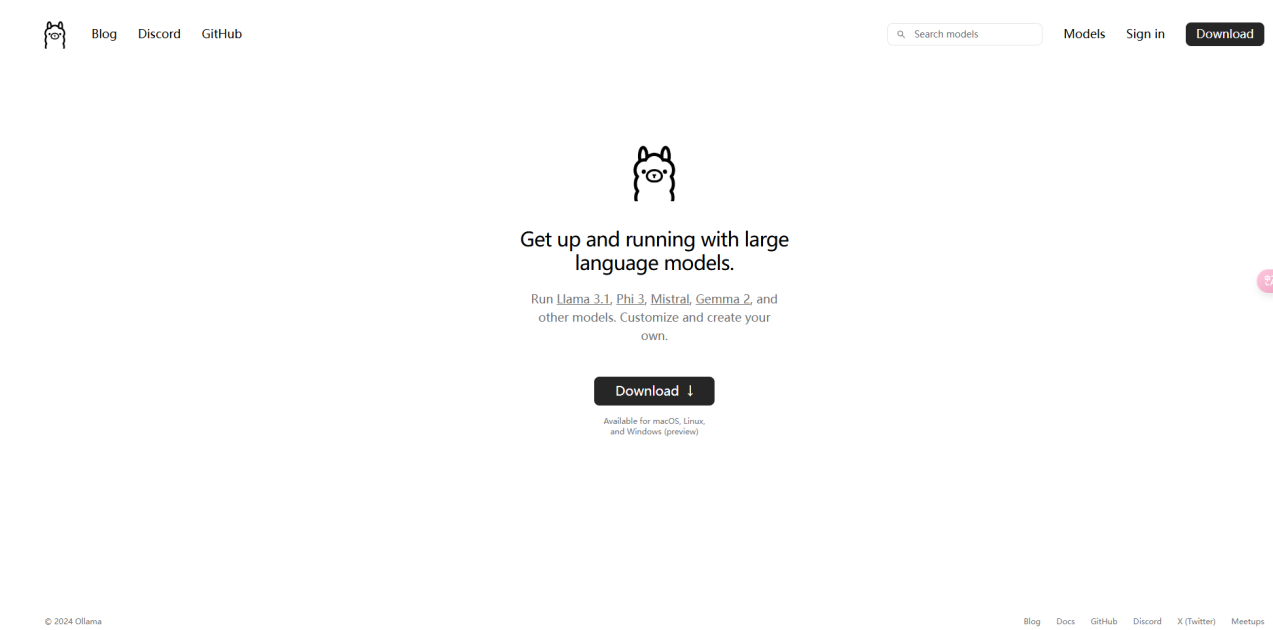
基于ollama私有化大模型

建议：在虚拟机上操作

ollama

ollama是一个将许多大模型管理起来，可以进行下载的，类似于docker镜像的平台。

网站: <https://ollama.com/>



安装

ollama平台下载

点击download，并选择Linux

Download Ollama



Linux

Windows

Install with one command:

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



[View script source](#) • [Manual install instructions](#)

下面会出现一个下载指令，在虚拟机上打开终端输入指令

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

如果显示没有 `curl`，输入指令 `sudo apt install curl`

然后在输入下载指令，等待下载完成。

but!!!!!!!!!! 你以为这就完了???? 这个时候，你可能发现，根本下不下来。通过咨询发现，原来是ollama3之后，国内下载就非常的龟速。这个时候，我们要对.sh文件进行操作。

首先，在自己想要存放的文件夹里面打开终端。

输入指令

```
1 # 下载安装脚本
2 curl -fsSL https://ollama.com/install.sh -o ollama_install.sh
3 # 给脚本添加执行权限
4 chmod +x ollama_install.sh
```

然后，打开 `ollama_install.sh` 文件

找到

- [https://ollama.com/download/ollama-linux-\\${ARCH}\\${VER_PARAM}](https://ollama.com/download/ollama-linux-${ARCH}${VER_PARAM})
- [https://ollama.com/download/ollama-linux-amd64-rocm.tgz\\${VER_PARAM}](https://ollama.com/download/ollama-linux-amd64-rocm.tgz${VER_PARAM})

这两个下载地址，然后分别改成

- <https://github.moeyy.xyz/https://github.com/ollama/ollama/releases/download/v0.3.2/ollama-linux-amd64>
- <https://github.moeyy.xyz/https://github.com/ollama/ollama/releases/download/v0.3.2/ollama-linux-amd64-rocm.tgz>

注意!!! 这里我们使用的是GitHub下载，记得打开梯子进行下载，如果不知道如何在虚拟机上挂梯子，可以尝试使用 [桥接模式](#)。

最后，再次打开终端，输入运行 `.sh` 的指令

```
1 sudo ./ollama_install.sh
```

还有就是，在下载过程，可能会出现如下情况，不要慌，这是因为梯子不稳定，重新上述 `.sh` 运行指令就行。

```
ljh@ljh-virtual-machine:~/AIagent$ sudo ./ollama_install.sh
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 CLI
##### 44.4%
curl: (92) HTTP/2 stream 0 was not closed cleanly: INTERNAL_ERROR (err 2)
```

下载完成

```
ljh@ljh-virtual-machine:~/AIagent$ sudo ./ollama_install.sh
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 CLI
##### 100.0%
>>> Making ollama accessible in the PATH in /usr/local/bin
>>> Creating ollama user...
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
Created symlink /etc/systemd/system/default.target.wants/ollama.service → /etc/systemd/system/ollama.service.
>>> The Ollama API is now available at 127.0.0.1:11434.
>>> Install complete. Run "ollama" from the command line.
WARNING: No NVIDIA/AMD GPU detected. Ollama will run in CPU-only mode.
```

大模型下载

(这里我们用千问举例子)

[Blog](#)[Discord](#)[GitHub](#)[Models](#)[Sign in](#)[Download](#)

Get up and running with large language models.

Run [Llama 3.1](#), [Phi 3](#), [Mistral](#), [Gemma 2](#), and other models. Customize and create your own.

[Download ↓](#)

Available for macOS, Linux,
and Windows (preview)

[中文](#)

熟悉的界面不同的选择，则此我一定会拿回我的一切！！！！（咳咳，中二了）

点击上方的Models

在搜索里面输入 `qwen2`



Models

[Featured](#)

qwen2

Qwen2 is a new series of large language models from Alibaba group

[0.5B](#) [1.5B](#) [7B](#) [72B](#)

↓ 2.2M Pulls 🔖 97 Tags ⌚ Updated 2 months ago

[中文](#)

点击进入后，先不要着急下载

qwen2

Qwen2 is a new series of large language models from Alibaba group

0.5B 1.5B 7B 72B

2.2M Pulls Updated 2 months ago

7b

97 Tags

ollama run qwen2

Updated 2 months ago		e0d4e1163c58 · 4.4GB
model	arch qwen2 · parameters 7.62B · quantization Q4_0	4.4GB
params	{ "stop": ["< im_start >", "< im_end >"] }	59B
template	{{ if .System }}< im_start >system {{ .System }}< im_end > {{ end...	182B
license	Apache License Version 2.0, January 2004 http://www.apache.org/li...	11kB

看到这个东西，点击它

7b

97 Tags

ollama run qwen2

latest4.4GB

0.5b352MB

1.5b935MB

7blatest4.4GB

72b41GB

View more

e0d4e1163c58 · 4.4GB	
parameters	7.62B · quantization Q4_0
stop	["< im_start >", "< im_end >"]
template	< im_start >system {{ .System }}< im_end > {{ end...
license	Version 2.0, January 2004 http://www.apache.org/li...

你会发现有很多不同的选择，如果是个人尝试的话，我建议下个最小的0.5b的就行，不然，你的虚拟机带不起来，30b要求30个核（好像maybe忘记了）

点击0.5b

0.5b	97 Tags	ollama run qwen2:0.5b	
Updated 2 months ago		6f48b936a09f · 352MB	
model	arch qwen2 · parameters 494M · quantization Q4_0		352MB
params	{ "stop": ["< im_start >", "< im_end >"] }		59B
template	{{ if .System }}< im_start >system {{ .System }}< im_end > {{ end...}}		182B
license	Apache License Version 2.0, January 2004 http://www.apache.org/li...		11kB

在图片的左上角，复制下来，输入到终端执行

```
1 ollama run qwen2:0.5b
```

等待下载

```

ljh@ljh-virtual-machine: ~/Desktop
ljh@ljh-virtual-machine:~/Desktop$ sudo ./ollama_install.sh
[sudo] password for ljh:
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 CLI
##### 100.0%
>>> Making ollama accessible in the PATH in /usr/local/bin
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
>>> The Ollama API is now available at 127.0.0.1:11434.
>>> Install complete. Run "ollama" from the command line.
WARNING: No NVIDIA/AMD GPU detected. Ollama will run in CPU-only mode.
ljh@ljh-virtual-machine:~/Desktop$ ollama run qwen2:0.5b
pulling manifest
pulling 8de95da68dc4... 6% | 22 MB/352 MB 2.9 MB/s 1m55s

```

成功

```
ljh@ljh-virtual-machine: ~/Desktop
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 CLI
##### 100.0%
>>> Making ollama accessible in the PATH in /usr/local/bin
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
>>> The Ollama API is now available at 127.0.0.1:11434.
>>> Install complete. Run "ollama" from the command line.
WARNING: No NVIDIA/AMD GPU detected. Ollama will run in CPU-only mode.
ljh@ljh-virtual-machine:~/Desktop$ ollama run qwen2:0.5b
pulling manifest
pulling 8de95da68dc4... 100% 352 MB
pulling 62fbfd9ed093... 100% 182 B
pulling c156170b718e... 100% 11 KB
pulling f02dd72bb242... 100% 59 B
pulling 2184ab82477b... 100% 488 B
verifying sha256 digest
writing manifest
removing any unused layers
success
>>> Send a message (/? for help)
```

这里其实就已经可以使用了。

下载OpenWebUI

[OpenWebUI](#)是一个可扩展、功能丰富且用户友好的自托管WebUI，它支持完全离线操作，并兼容Ollama和OpenAI的API。这为用户提供了一个可视化的界面，使得与大型语言模型的交互更加直观和便捷。

安装OpenWebUI

在已经下载的虚拟机上输入指令

```
1 docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open
2 -webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/
3 open-webui:main
```

没有 `docker` 的输入下载 `docker` 的指令

安装docker

你需要先安装 Docker 才能运行 Docker 容器。你可以使用以下命令来安装 Docker:

```
1 sudo apt update
2 sudo apt install docker.io
```

安装完成后，启动 Docker 服务：

```
1 sudo systemctl start docker
2 sudo systemctl enable docker
```

安装完成后，你可以使用以下命令检查 Docker 是否正常安装：

```
1 docker --version
```

如果返回 Docker 的版本信息，说明安装成功。

运行刚刚的指令

```
1 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
```

然后等待下载完成

```
ljh@ljh-virtual-machine:~/Desktop$ sudo 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
Unable to find image 'ghcr.io/open-webui/open-webui:main' locally
main: Pulling from open-webui/open-webui
e4fff0779e6d: Pull complete
d97016d0706d: Pull complete
53db1713e5d9: Pull complete
a8cd795d9ccb: Pull complete
de3ba92de392: Pull complete
6f4d87c224b0: Pull complete
4f4fb700ef54: Pull complete
dd92a6022ddb: Pull complete
bbbfe48a772: Pull complete
a825beebdb5b: Pull complete
e0694836bfeb: Pull complete
89130c556665: Downloading 3.78MB/1.03GB
65c0b34ebdc: Downloading 15.69MB/51.57MB
7fa483e72b55: Download complete
9f941e1e5a9f: Download complete
2a97a550d722: Download complete
```


等了两天终于成功了，主要是网络下的太慢了，第一天超时了第二天下了一天才下完

```
st.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main
Unable to find image 'ghcr.io/open-webui/open-webui:main' locally
main: Pulling from open-webui/open-webui
e4fff0779e6d: Pull complete
d97016d0706d: Pull complete
53db1713e5d9: Pull complete
a8cd795d9ccb: Pull complete
de3ba92de392: Pull complete
6f4d87c224b0: Pull complete
4f4fb700ef54: Pull complete
dd92a6022ddb: Pull complete
bbbfe48a772: Pull complete
a825beebdb5b: Pull complete
e0694836bfeb: Pull complete
89130c556665: Pull complete
65c0b34ebdc: Pull complete
7fa483e72b55: Pull complete
9f941e1e5a9f: Pull complete
2a97a550d722: Pull complete
Digest: sha256:5bf373c9885c3f5b8411a26f5d233aef11cb36056cc22077b7a6fcd7f62901ce
Status: Downloaded newer image for ghcr.io/open-webui/open-webui:main
f37d400fc9669ad8f568a0e9b3c49e3da9f50717ea51a624b9e1343a7dbf2cdf
```

然后，你可以输入指令进行查看当前运行的容器（open-webui）

```
1 docker ps
```

如果显示 `permission ...` 应该是没有权限，加上 `sudo` 就行

```
1 sudo docker ps
```

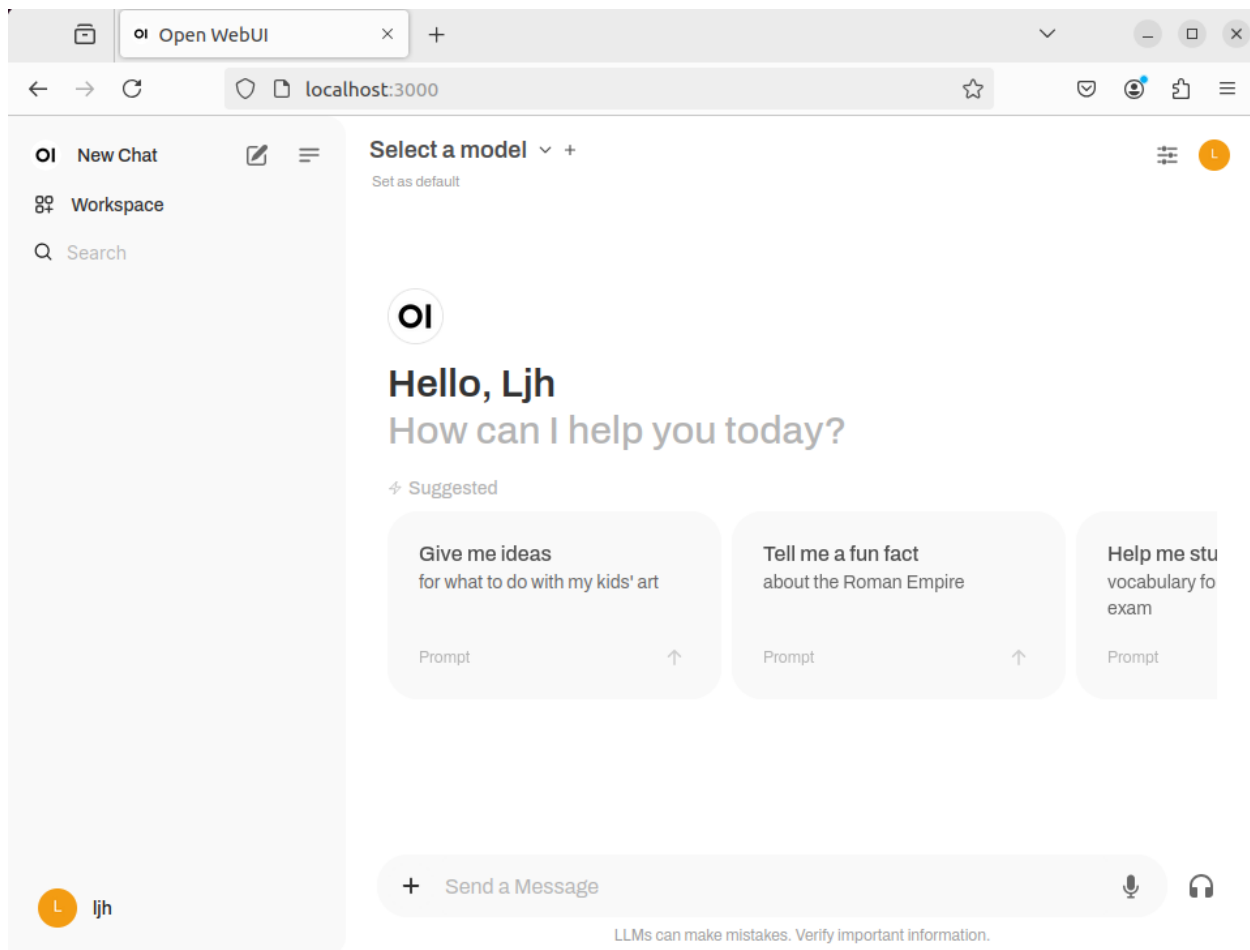
```
ljh@ljh-virtual-machine:~/Desktop$ sudo docker ps
[sudo] password for ljh:
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
f37d400fc966   ghcr.io/open-webui/open-webui:main "bash start.sh"         15 hours ago   Up 11 minutes (healthy)   0.0.0.0:3000->8080/tcp, :::3000->8080/tcp   open-webui
```

可以看到正在运行中

如此，你就可以打开本地的浏览器，输入网站进行访问了，直接输入就行

```
1 http://localhost:3000
```

登录的时候需要注册，随便填就行，然后进入界面



ok! 第二个下载完成!!!

结合ollama和OpenWebUI

配置环境变量

1. 打开并编辑服务文件

```
1 sudo vim /etc/systemd/system/ollama.service
```

2. 先点击任意键，进入insert模式，添加环境变量

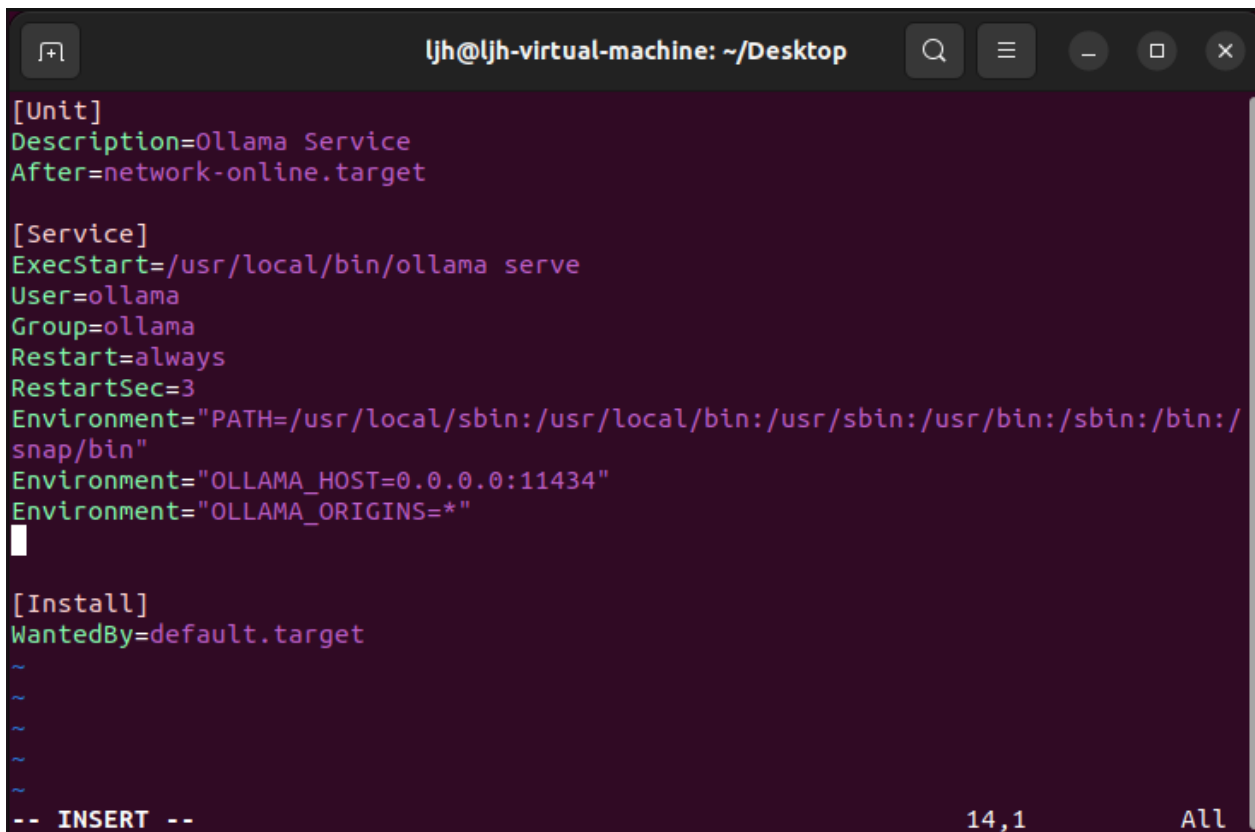
```
1 [Service]
2 ExecStart=/usr/local/bin/ollama serve
3 User=ollama
4 Group=ollama
```

```
5 Restart=always
6 RestartSec=3
7 Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
8 Environment="OLLAMA_HOST=0.0.0.0:11434"
9 Environment="OLLAMA_ORIGINS=*"

```

把下面两条变量添加进去就行

- Environment="OLLAMA_HOST=0.0.0.0:11434"
- Environment="OLLAMA_ORIGINS=*"



```
[Unit]
Description=Ollama Service
After=network-online.target

[Service]
ExecStart=/usr/local/bin/ollama serve
User=ollama
Group=ollama
Restart=always
RestartSec=3
Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Environment="OLLAMA_HOST=0.0.0.0:11434"
Environment="OLLAMA_ORIGINS=*"

[Install]
WantedBy=default.target
~
~
~
~
-- INSERT --
14,1 All

```

3. 保存并退出 vim

在 vim 中，执行以下步骤保存并退出：

- 按 `Esc` 键确保您处于命令模式。
- 输入 `:wq` 然后按 `Enter` 保存文件并退出。

4. 重新加载 systemd 并重启服务

```
1 sudo systemctl daemon-reload
2 sudo systemctl restart ollama.service

```

重启docker容器

```
1 sudo docker restart open-webui
```

想要查看当前你的docker中的所有容器的话，输入指令

```
1 sudo docker ps -a
```

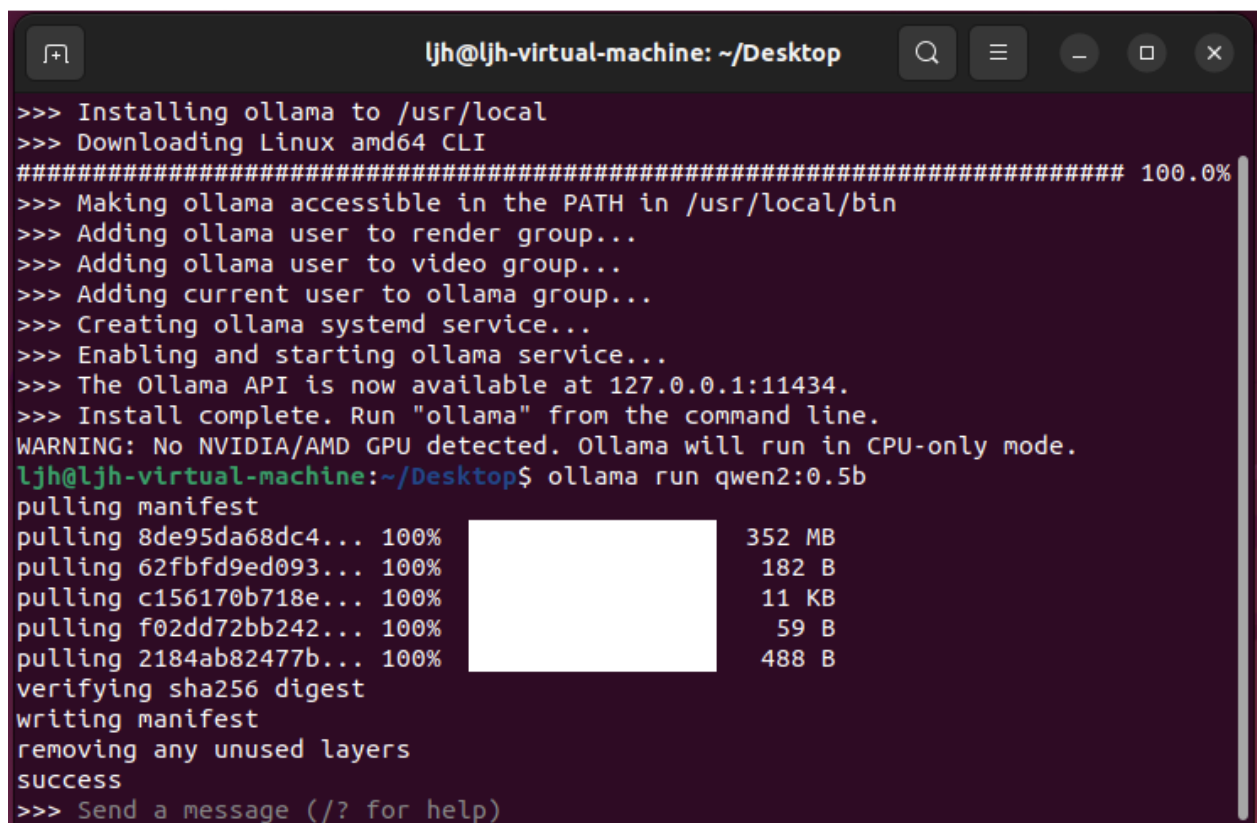
启动ollama模型

其实这里的ollama模型也就是我们刚刚之前下载的qwen2模型，就是后端

下载的OpenWebUI，就是前端

想要前端可以发挥功能，就应该要启动后端

如果如下界面你关闭了



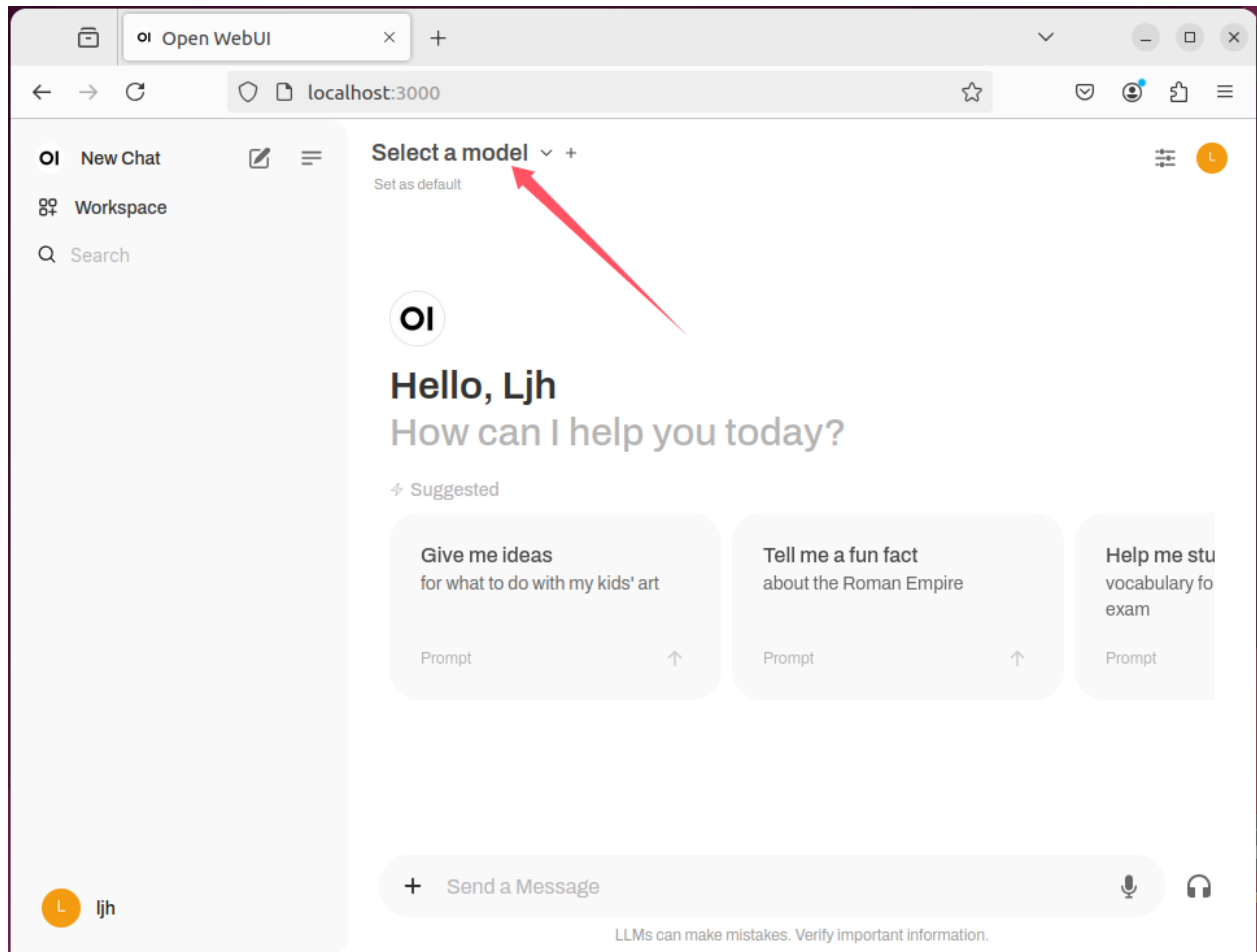
```
ljh@ljh-virtual-machine: ~/Desktop
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 CLI
##### 100.0%
>>> Making ollama accessible in the PATH in /usr/local/bin
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
>>> The Ollama API is now available at 127.0.0.1:11434.
>>> Install complete. Run "ollama" from the command line.
WARNING: No NVIDIA/AMD GPU detected. Ollama will run in CPU-only mode.
ljh@ljh-virtual-machine:~/Desktop$ ollama run qwen2:0.5b
pulling manifest
pulling 8de95da68dc4... 100% 352 MB
pulling 62fbfd9ed093... 100% 182 B
pulling c156170b718e... 100% 11 KB
pulling f02dd72bb242... 100% 59 B
pulling 2184ab82477b... 100% 488 B
verifying sha256 digest
writing manifest
removing any unused layers
success
>>> Send a message (/? for help)
```

重新执行指令（不会重新下载，会直接启动）

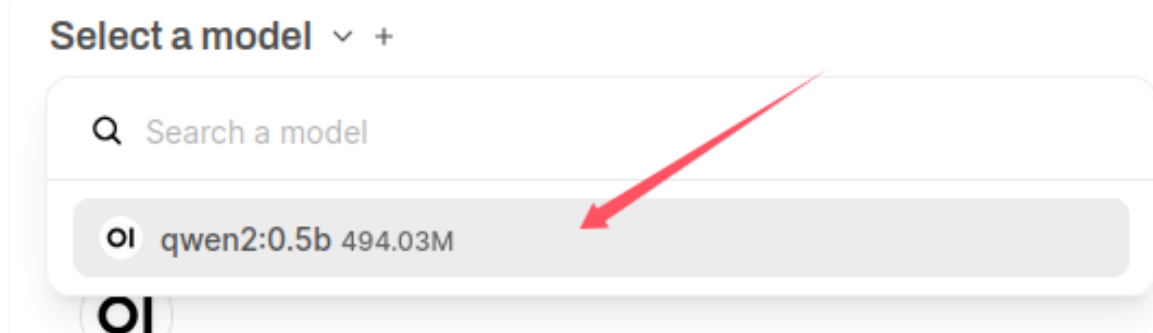
```
1 ollama run qwen2:0.5b
```

然后，打开网页，输入网址

```
1 http://localhost:3000
```



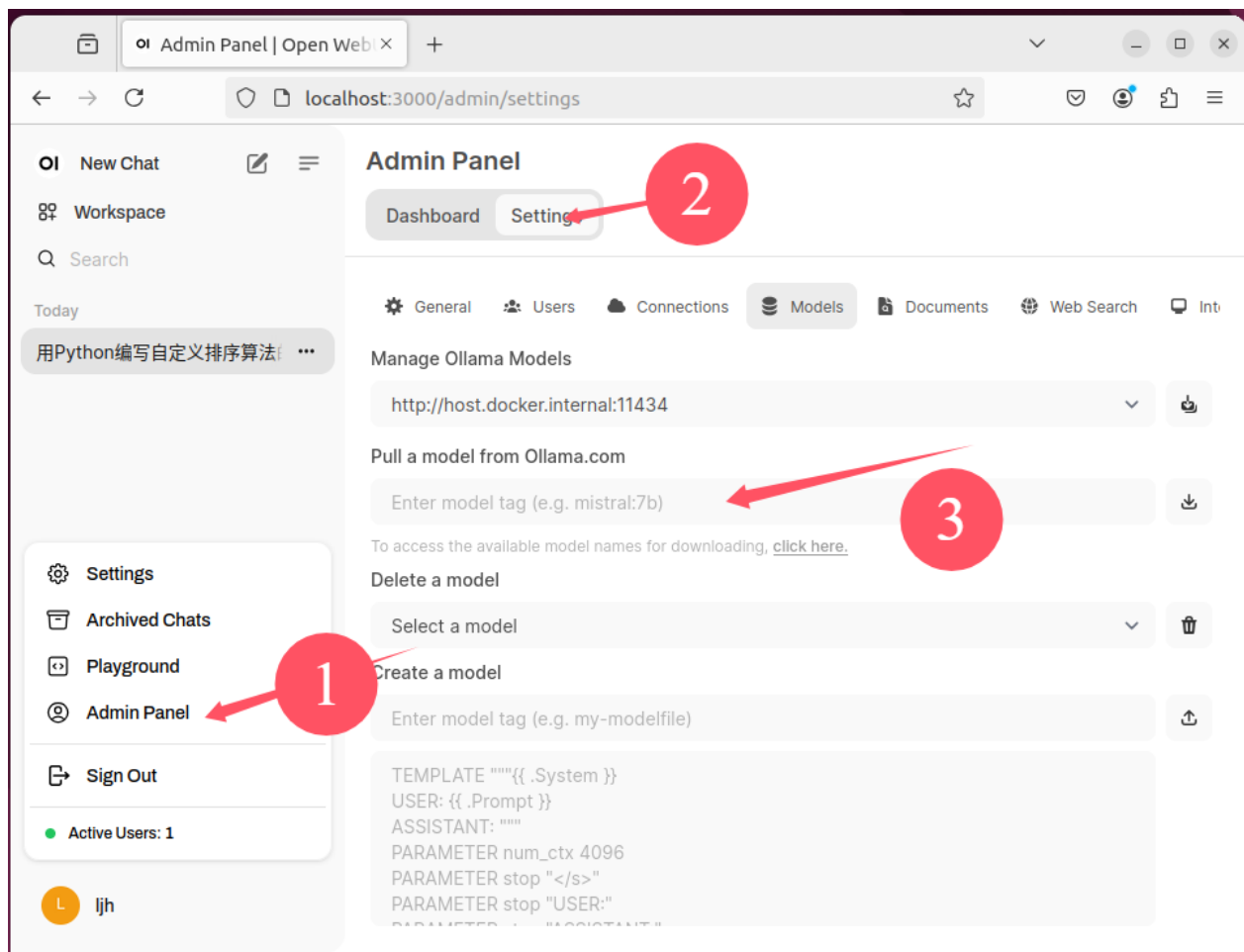
点击 `Select a model`，点击你下载模型就好了



然后你就可以快乐的使用自己私有化的AI了！！！！



这里我再附上另一个可以在OpenWebUI上使用功能的方法



我负责帮你找到下载的地方，其他的操作看：

[适配Ollama的前端界面Open WebUI](#)

OK! 完结!!! *★,°*:.☆(￣▽￣)/\$:*°★*。