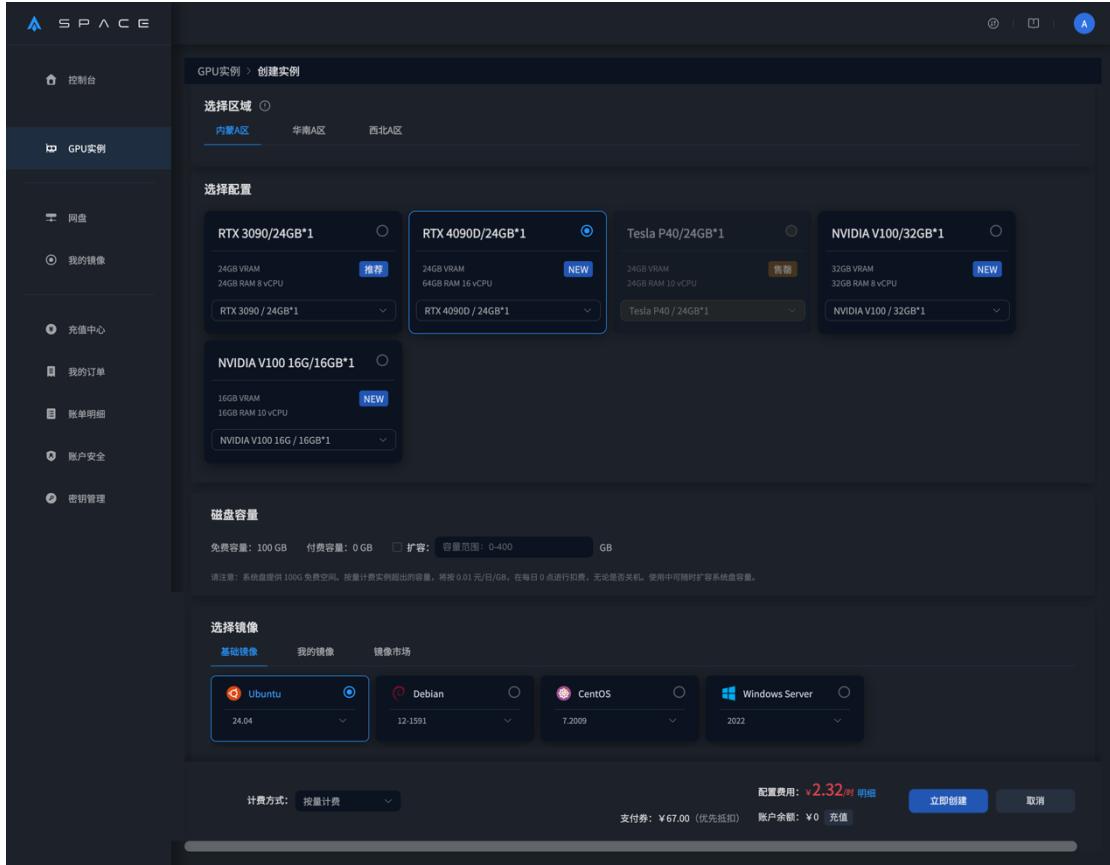


## 一、 云主机选择



1.1 不是 Windows 玩不起 ollama，而是 Linux 更专业。这里选择 Ubuntu24.04，是最新的 Ubuntu LTS (long time support) 版本，其实 22.04、20.04 也一样可以的。

1.2 配置选择的是内蒙 A 区的 RTX4090D/24G\*1 规格，考虑到一般常见不超过 30B (300 亿参数) 开源 LLM (大语言模型) 需要 24G 显存即可在 ollama 环境使用，再大的话就得考虑华南 A 区的 AD103/32GB\*1 甚至 AD102/48GB\*1 规格了，但下边介绍的部署方法没有区别。硬盘 100G 足以，后期下载模型统统放到网盘里，既方便多机共享还节省镜像磁盘空间。

## 二、 远程登录

2.1 我自己用的是 Linux 系统，所以天生自带 ssh 命令，如果本

机用 windows 的话就得安装一些诸如 xshell、Tabby 之类的工具了，本文不多展开。

2.2 执行 ssh-keygen 命令，生成一套 ssh 登录认证的密钥对（如果已经有现成的忽略本步骤）直接默认回车三次即可。

```
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RnVna38UCwyttQlI7o0qUJoAIL/LE+1i/JbgBpq2Nso root@R830-134
The key's randomart image is:
+---[RSA 3072]---+
|=.       ..o.=.o. |
|o.       .o o *..o|
| .. . .. + =..|
| .o+   .. o. + o |
| o+.   So .     o|
|.o.+.   ..         .|
|.+B..o.           |
|+=o+o .          |
|=Eo..          |
+---[SHA256]---+
```

2.3 然后查看上步生成的公钥内容（/root/.ssh/id\_rsa.pub），这个是完全公开的可以放置到你想登录的任意 Linux 系统里。但是另外一个私钥文件内容（/root/.ssh/id\_rsa）打死也不要告诉别人^\_^

```
cat /root/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQgQDW/N6+19Junl0dwYSknF8aPp/TcuUKe09U0rnN
PO7eq3ubK4AtdkDxeEdXUmQJMk+TbfgQzVY0S0QMe9Qj2+39uchLYLXrWWSPJlw5NHwE
Y0hVB1lrf0NCvaDouCowaDOwttczRHQTWlg7smBUWLYc8KGBVv7FHLYBnPXAibwFtKuNjj5
8jvsEmRJjFwFGaNLCj/taG3+6OVvoy0ctYa0F7Z5KN9H5S4kcEQWAvtFWzbfmG3UjCSNcfL9G
PeJNrqpP8UVWIQ/XbMs2pc/Bw2psy6nGTsVEc8VbgzOS9g3kEBpsus1ojk/04n8Y3iPSO7xt2
erknWsHiXTxmUuVealuYXDO6W3tkASZhSa1KkBTNkT/ng6nP+G2bEWTLiCRuhpv81A7IKre
CSEvnspdCRQI/Sv84HBWW0jK24fNQRTuFth7xoillrwciaeUEayhRyqsb51FzuQJWURcT2t6SZ
CMv1eL1a0rvHaFiXobUUu81gbT9SBfejLCgbldmqaf0s= root@R830-134
```

2.4 打开云主机的 JupyterLab 服务，并把上一步生成的公钥内容从 ssh-rsa 开始整段复制，然后粘贴到云主机的认证文件里（/root/.ssh/authorized\_keys）

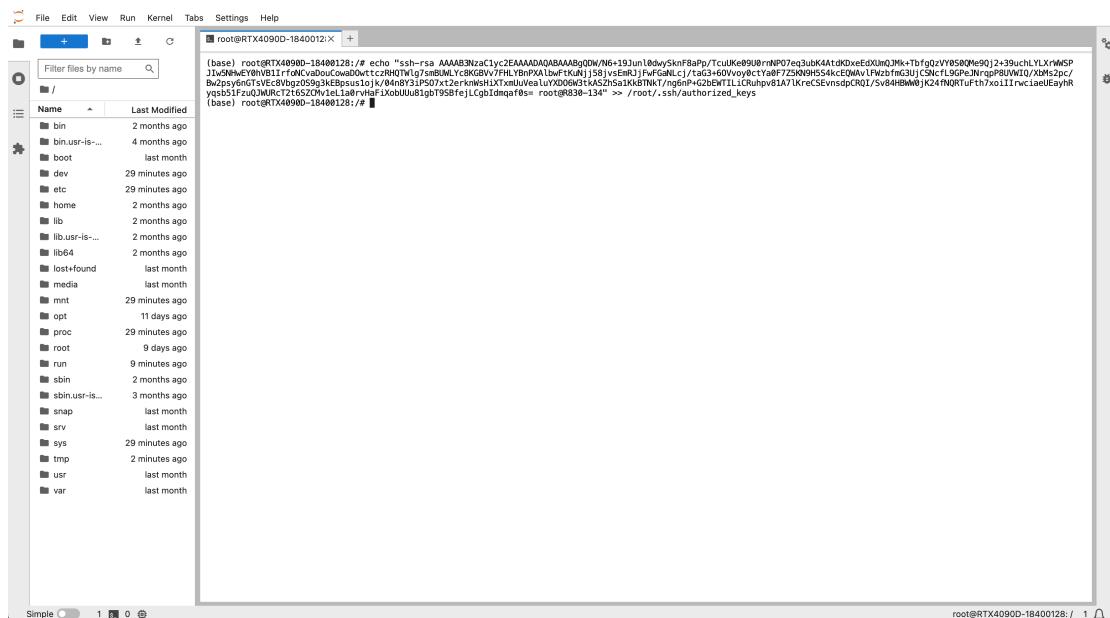
The top screenshot shows the GPU instance management interface. A red arrow points to the 'JupyterLab' link next to the first instance (RTX4090D-e9vPhL). The bottom screenshot shows a terminal window titled 'Launcher'. A red arrow points to the 'Terminal' icon in the 'Other' section of the launcher.

然后执行命令如下命令，解释下命令四部分的含义，注意每部分之间至少一个空格分开。echo 是显示字符串的命令，随后输入英文半角双引号包着的从 2.3 步骤复制过来的内容，然后两个大于号是重定向输出追加到某个文件最后边，最后是 root 账号默认的认证文件。

```

echo "ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQgQDW/N6+19Junl0dwYSknF8aPp/TcuUKe09U0rn
NPO7eq3ubK4AtdKDxeEdXUmQJMk+TbfgQzVY0S0QMe9Qj2+39uchLYLXrWWSPJlw5N
HwEY0hVB1Irf0NCvaDouCowaDOWttczRHQTWlg7smBUWLYc8KGBVv7FHLYBnPXAibwFt
KuNjj58jvsEmRJjFwFGaNLCj/taG3+6OVvoy0ctYa0F7Z5KN9H5S4kcEQWAvtFWzbmG3UjC
SNcfL9GPeJNrqP8UVWIQ/XbMs2pc/Bw2psy6nGTsVEc8VbgzOS9g3kBpsus1ojk/04n8Y
3iPSO7xt2erknWsHiXTxmUuVealuyXDO6W3tkASZhSa1KkBTNkT/ng6nP+G2bEWTLiCRu
hpv81A7IKreCSEvnspCRQ1/Sv84HBWW0jk24fnQRTuFth7xoillrwciaeUEayhRyqsb51FzuQ
JWURcT2t6SZCMv1eL1a0rvHaFiXobUUu81gbT9SBfejLCgbldmqaf0s=      root@R830-
134" >> /root/.ssh/authorized_keys

```



回车后应该是没有任何反应的，依旧是一个闪烁的命令行提示符。

## 2.5 复制命令，通过 root 账号登录刚刚开通的云主机。

The screenshot shows a cloud provider's GPU instance management interface. On the left sidebar, there are several menu items: 控制台 (Console), GPU实例 (GPU Instances), 网盘 (Cloud Disk), 我的镜像 (My Images), 充值中心 (Top-up Center), 我的订单 (My Orders), 账单明细 (Bill Details), 账户安全 (Account Security), and 密钥管理 (Key Management). The main panel is titled "GPU实例" and displays a list of instances. One instance is highlighted: "RTX4090D-e5vPhL 18400128" which is currently "运行中" (Running) and uses an RTX 4090D\*1 card. It has a status of "已关机" (Powered Off), a price plan of "按量计费" (Pay-as-you-go), and a shutdown time of "手动关机" (Manual Shutdown). The instance is located in "内蒙A区" (Inner Mongolia A Region) and has a connection string "ssh root@116.113.133.20" followed by a password placeholder "\*\*\*\*\*". There are also "JupyterLab" and "资源监控" (Resource Monitoring) links. Below the instance list, there are buttons for "开机" (Power On) and "关机" (Power Off). At the bottom, there are pagination controls for "共 5 条" (5 items total), "10条/页" (10 items per page), and a "前往" (Go To) button.

然后就可以粘贴复制内容，在我自己的 Linux 服务器免密码登录刚刚开通的云主机了，第一次连接要打个 yes 以后就不用了。

```
ssh root@116.113.133.20 -p 10004
The authenticity of host '[116.113.133.20]:10004 ([116.113.133.20]:10004)' can't be established.
ED25519 key fingerprint is SHA256:YaAK49pwge6PTnI5NNuDoWHTm/1tkrfNZW6x9jYz6hM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[116.113.133.20]:10004' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Sat Jun 29 10:07:57 UTC 2024
      System load:  0.29                  Processes:           245
      Usage of /:   10.6% of 95.82GB    Users logged in:       0
      Memory usage: 0%                   IPv4 address for eth0: 10.6.66.9
      Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

(base) root@RTX4090D-18400128:~#
```

### 三、验证 GPU 工作状态、初始化系统

Nvidia 显卡在 Linux 命令行模式可以直接查看硬件工作状态

```
(base) root@RTX4090D-18400128:~# nvidia-smi
Sat Jun 29 10:11:19 2024
+-----+
| NVIDIA-SMI 550.54.14           Driver Version: 550.54.14     CUDA Version: 12.4   |
+-----+
| 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 RTX 4090 D    Off | 00000000:01:00.0 Off |          Off | | | |
| 31% 38C   P0          N/A / 425W | 0MiB / 24564MiB | 0%       Default |
| |               |             |              |           | N/A |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                               |
| GPU  GI  CI          PID  Type  Process name        GPU Memory |
| ID   ID          ID          |          |                 Usage  |
+=====+=====+=====+=====+=====+=====+=====+=====+
| No running processes found               |
+-----+
```

对于初学者来说能看到这个内容就可以了，假如异常的话得找客服去解决硬件或者云主机的系统问题。

3.2 然后执行 `apt update && apt upgrade` 命令去升级现有 Ubuntu 官方的系统软件、显卡驱动、各个应用软件。注意此命令 5 个部分之间依旧需要至少一个空格分开，期中前两部分是一个命令，中间`&&`代表第一个命令执行成功后，才会去支持后两部分的命令。执行后需要输入 Y 回车，因为输出内容比较多，就不完全复制到文档里了。此命令大概需要执行 3-5 分钟左右，最后完成的时候最后几行应该显示如下：

```
(base) root@RTX4090D-18400128:~# apt update && apt upgrade
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
中间省略若干行……
No containers need to be restarted.

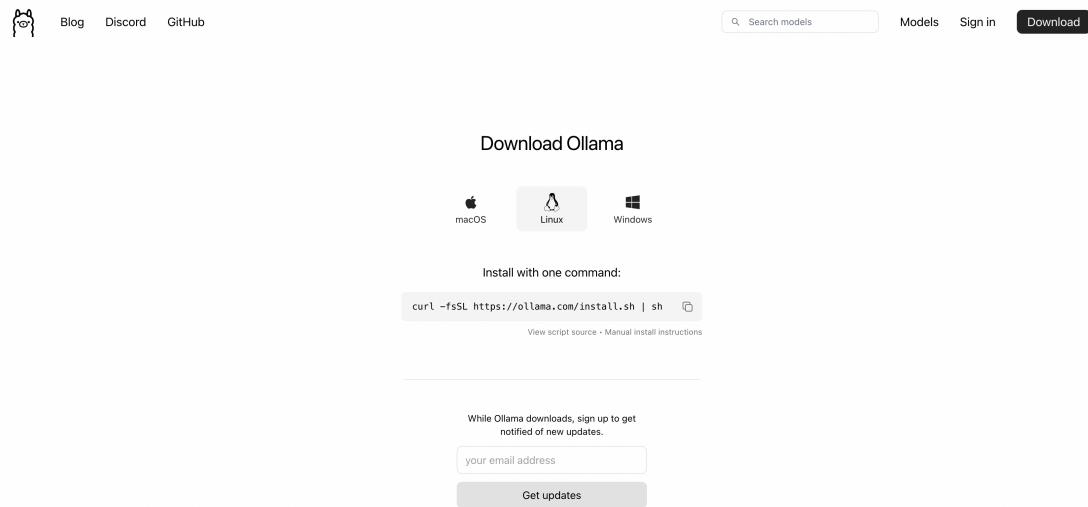
User sessions running outdated binaries:
root @ session #4: apt[4106], bash[2986], sshd[2868]
root @ user manager service: systemd[2873]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
(base) root@RTX4090D-18400128:~#
```

执行完成后输入 reboot 命令重启系统，需要等待 1 分钟左右再次执行 2.5 步骤中复制的 ssh 远程登录命令，所有准备工作就完成了。

#### 四、 安装 ollama 服务

复制 ollama 官网提供的 Linux 系统一键安装脚本即可自动完成本机的 ollama 服务安装 (<https://ollama.com/download/linux>)



注意：为了加快速度一定先要执行下 source /etc/network\_turbo 然后再执行从 ollama 官网复制的安装命令。详见官方帮助文档 <https://spacehpc.feishu.cn/wiki/KamLwQyM0i0Sehkx8hFcu4W1nzh>

```
(base) root@RTX4090D-18400128:~# source /etc/network_turbo
(base) root@RTX4090D-18400128:~# curl -fsSL https://ollama.com/install.sh | sh
>>> Downloading ollama...
#####
##### 100.0%##O=#
#####
##### 100.0%^@
>>> Installing ollama to /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.
>>> NVIDIA GPU installed.
(base) root@RTX4090D-18400128:~# ollama -v
ollama version is 0.1.48
```

安装完成后执行 ollama -v 命令能看到输出当前 ollama 版本号就是成功了。

## 五、 配置模型下载地址到网盘共享目录

### 5. 1 先停止已经启动的 ollama 服务

```
(base) root@RTX4090D-18400128:~# systemctl stop ollama
(base) root@RTX4090D-18400128:~# ollama -v
Warning: could not connect to a running Ollama instance
Warning: client version is 0.1.48
```

5. 2 然后迁移 ollama 软件安装目录到网盘目录，这样既不占用云主机的本地硬盘，假如多开服务器的话，又可以共享给自己账号里其他云主机使用 ollama 模型，无需重复下载。注意执行 mv 命令的时候千万不要中断，否则就得重新安装 ollama 才能确保不丢文件了。再次提醒，必须看到 Warning: client version is 0.1.48 才能继续下边的操作。

```
(base) root@RTX4090D-18400128:~# mv /usr/share/ollama/ /mnt/storage/
(base) root@RTX4090D-18400128:~# ln -s /mnt/storage/ollama/ /usr/share/
(base) root@RTX4090D-18400128:~# systemctl start ollama
(base) root@RTX4090D-18400128:~# ollama -v
ollama version is 0.1.48
```

### 5.3 解释下几个命令的作用：

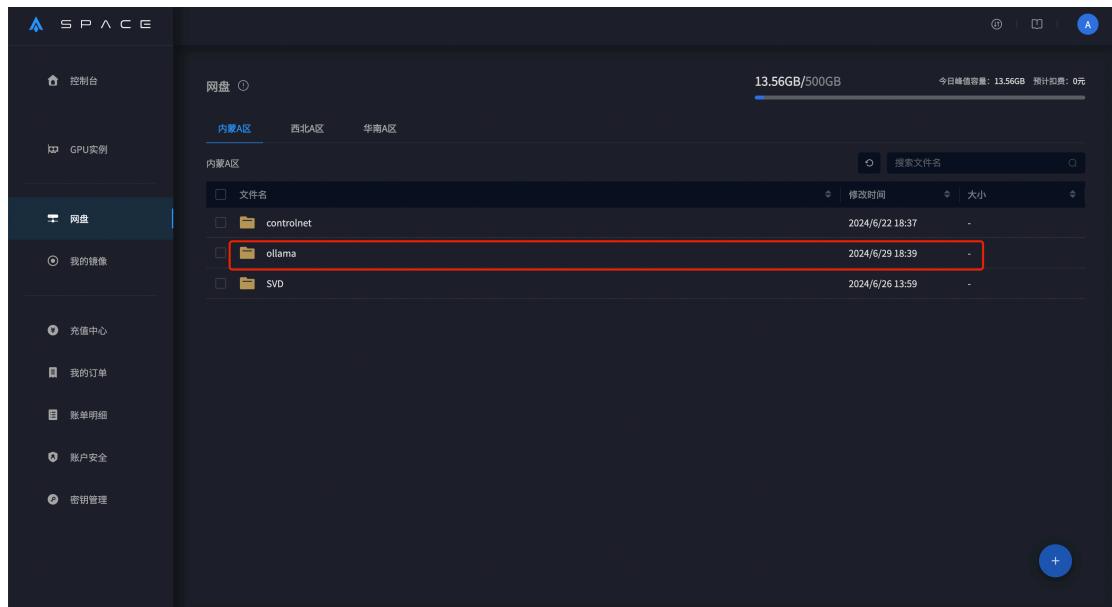
mv 是把 ollama 默认安装的位置剪切到网盘目录去

ln 命令是把网盘目录里的 ollama 再创建一个软连放回默认路径

systemctl 是再次启动 ollama 服务

最后 ollama -v 看到的结果就不像 5.1 最后显示有 Warning 了

5.4 然后我们去看看自己的网盘，应该多了一个叫 ollama 目录，这个文件夹就不要随便动了啊，删除、改名、移动位置，都会导致 ollama 服务无法使用。



注：截图里的 controlnet 和 SVD 是我其他 ai 项目使用的共享文件夹，与本次 ollama 服务安装没有任何关系，忽略即可。

## 六、 安装 open-webui

这个安装是为了方便通过 web 方式去使用 ollama 服务，否则只能在命令行里玩耍那就太不方便了。

6.1 先通过 conda 创建一个 python3.11 版本的虚拟环境，方便以后多个环境在同一个云主机里并存互不影响，需要按一次 y 同意。

```
(base) root@RTX4090D-18400128:~# conda create -n webui python=3.11
Channels:
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
中间省略若干行……
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate webui
#
# To deactivate an active environment, use
#
#     $ conda deactivate
(base) root@RTX4090D-18400128:~#
```

解释下 conda 命令中 5 个部分的意义。首先是 conda，然后是 create 表示要创建，然后是-n 表示给这个虚拟环境起个名字，后边的 webui 就简单明了，个人方便好记就可以了，最后的 python=3.11 代表这个虚拟环境用的 python 版本。查看当前 conda 环境也很容易，看每行提示符最左边的括号里 (base) 代表就是默认的系统环境。使用命令 conda activate webui 切换后就是-n 后边第 4 部分的名字了。

```
(base) root@RTX4090D-18400128:~# conda activate webui  
(webui) root@RTX4090D-18400128:~#
```

注意看上边行最左括号里是 base，执行命令后就变成 webui 了。

6.2 使用 pip 命令安装 open-webui 服务，一定要看到左边括号里是 webui 后方可执行如下命令。这命令大概需要 10 分钟才能完成。

```
(webui)      root@RTX4090D-18400128:~#     pip      install      open-webui      -i  
https://pypi.tuna.tsinghua.edu.cn/simple  
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple  
Collecting open-webui  
  Downloading  
https://pypi.tuna.tsinghua.edu.cn/packages/49/5c/6042ca1d65682a14fe3d0e792cd78e616  
6d3bd258b94c331bbde7c7a2d5b/open_webui-0.3.6-py3-none-any.whl (61.5 MB)  
  
-----  
— 61.5/61.5 MB 11.7 MB/s eta 0:00:00  
Collecting aiohttp==3.9.5 (from open-webui)  
中间省略若干行……  
ujson-5.10.0 unstructured-0.14.0 unstructured-client-0.23.8 uritemplate-4.1.1 urllib3-2.2.2  
uvicorn-0.22.0 uvloop-0.19.0 validators-0.28.1 watchfiles-0.22.0 websocket-client-1.8.0  
websockets-12.0 wrapt-1.16.0 wsproto-1.2.0 xlrld-2.0.1 yarl-1.9.4 youtube-transcript-api-  
0.6.2 zipp-3.19.2  
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting  
behaviour with the system package manager. It is recommended to use a virtual environment  
instead: https://pip.pypa.io/warnings/venv  
(webui) root@RTX4090D-18400128:~#
```

解释下这个命令 5 部分的含义，pip 是 python 扩展安装命令，第 2 部分 install 代表让 pip 安装，第 3 部分 open-webui 就是我们想要安装的服务名，后边的-i 是代表用哪个软件源，最后一串网址是清华大学的 pip 源地址，国内用户使用这个最快。

当然也可以通过其他方式安装 open-webui，比如说 docker。但是安装 docker，修改 docker 镜像拉取使用国内的地址，或者配置科学

上网使用 docker 对于非专业人员来说会更复杂。

### 6.3 检查 open-webui 安装是否成功。正常安装后应该显示如下

```
(webui) root@RTX4090D-18400128:~# pip show open-webui
```

Name: open-webui

Version: 0.3.6

Summary: Open WebUI (Formerly Ollama WebUI)

Home-page:

Author:

Author-email: Timothy Jaeryang Baek <tim@openwebui.com>

License: MIT License

Copyright (c) 2023 Timothy Jaeryang Baek

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
SOFTWARE.

Location: /root/miniconda3/envs/webui/lib/python3.11/site-packages

Requires: aiohttp, apscheduler, argon2-cffi, authlib, bcrypt, black, boto3, chromadb, docx2txt, duckduckgo-search, extract-msg, fake-useragent, fastapi, faster-whisper, flask, flask-cors, fpdf2, google-generativeai, langchain, langchain-chroma, langchain-community, langfuse, markdown, opencv-python-headless, openpyxl, pandas, passlib, peewee, peewee-migrate, psycopg2-binary, pydantic, pydub, pyjwt, pymysql, pypandoc, pypdf, python-jose, python-multipart, python-socketio, pytube, pyxlsb, rank-bm25, rapidocr-onnxruntime, requests, sentence-transformers, unstructured, unicorn, validators, xlrd, youtube-transcript-api

Required-by:

## 七、 安装 open-webui 依赖的 ffmpeg，否则启动时会报错

```
(webui) root@RTX4090D-18400128:~# apt install -y ffmpeg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
中间省略若干行.....
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.10+dfsg-3ubuntu3.1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
(webui) root@RTX4090D-18400128:~#
```

## 八、 启动 open-webui 服务

第一次启动应该会自动去下载 huggingface 依赖环境。所以一定要先执行加速命令。

```
(webui) root@RTX4090D-18400128:~# source /etc/network_turbo
(webui) root@RTX4090D-18400128:~# open-webui serve
Loading WEBUI_SECRET_KEY from file, not provided as an environment variable.
Loading WEBUI_SECRET_KEY from /root/.webui_secret_key
/root/miniconda3/envs/webui/lib/python3.11
中间省略若干行.....
v0.3.6 - building the best open-source AI user interface.

https://github.com/open-webui/open-webui

INFO:     Started server process [6296]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

看到最后 running on <http://0.0.0.0:8080> 就说明启动成功了。

注意在使用过程中不要关闭 ssh 窗口也不要按 CTRL + C。

## 九、 ssh 隧道映射 web 服务

9.1 因为平台默认只给一共一个 ssh 连接端口，得想办法把上一步 open-webui 服务映射出来，这样就可以方便本机浏览器访问了。

按照 2.5 步新开一个 ssh 窗口，登录成功后执行如下命令并保证不断

```
ssh -p 10004 -N -L 8888:localhost:8080 root@116.113.133.20
```

解释下这个命令相对有些难懂：

ssh 这个无需解释了

-p 10004 是从第 2.5 步复制过来的平台给什么端口就是什么

-N 是不要登录只保留当前 ssh 链接隧道即可

-L 是在本机创建一个映射的监听端口

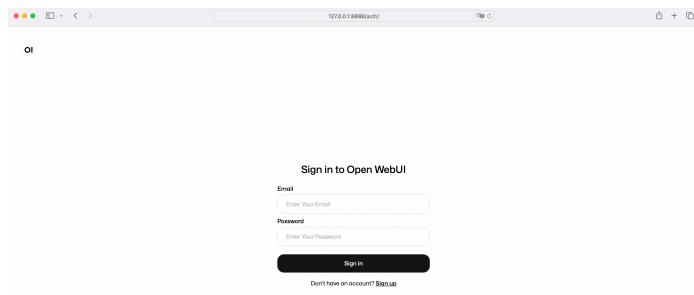
8888 是指定本机端口号，不要小于 1024 不要大于 65534，并且不重复已有本机监听端口

localhost 是监听端口映射到本机

8080 是刚刚 open-webui 启动的服务端口

root@116.113.133.20 是从 2.5 步复制过来的，系统给云主机分配的公网 ip

9.2 验证映射是否成功，浏览器直接输入 <http://127.0.0.1:8888>



## 十、 登录 webui 下载模型并使用

### 10.1 第一次登录可以直接注册一个登录邮箱和密码

#### Sign up to Open WebUI

ⓘ Open WebUI does not make any external connections, and your data stays securely on your locally hosted server.

Name

ollama

Email

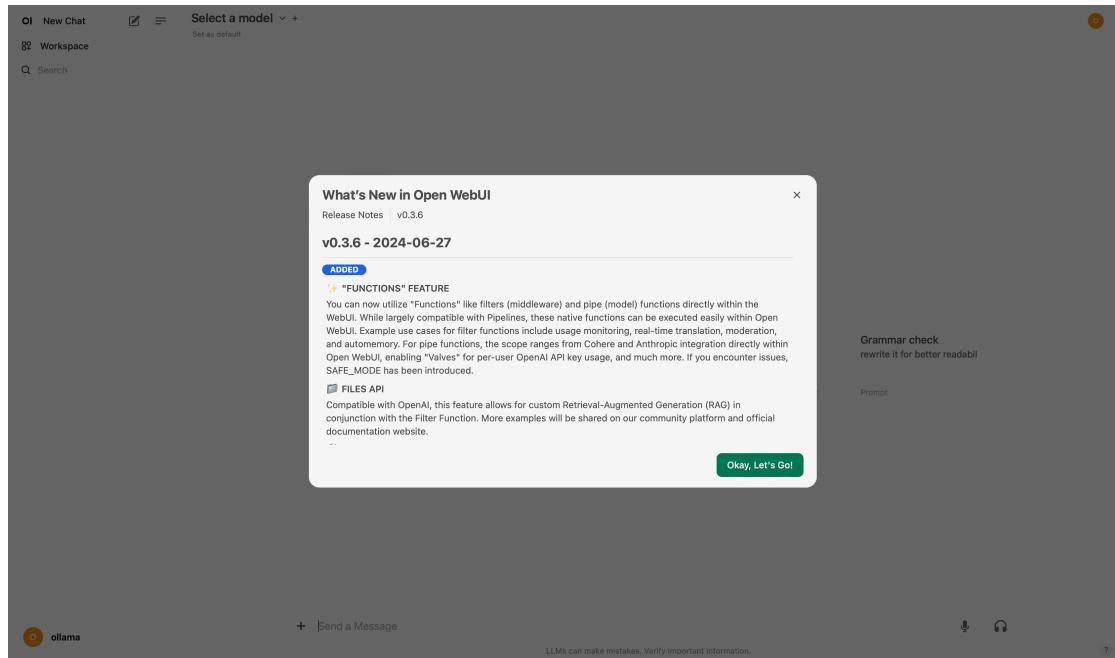
ollama@spacehpc.com

Password

.....

Create Account

Already have an account? [Sign in](#)



恭喜你终于可以开始愉快的玩耍了。

## 10.2 修改成中文界面

点左下角的头像，然后在 Setting 配置菜单里选择“简体中文”并保存。



## 10.3 下载第一个 LLM 模型

点左下角的头像，然后在管理员面板里找到“模型”菜单。输入模型名字下载 (<https://ollama.com/library>) 我们这里选择一个 Google 刚刚开源的 gemma2，点右边的下载按钮耐心等待下载完成。

The image consists of three vertically stacked screenshots of the Ollama Admin Panel. Each screenshot shows the 'Models' section of the interface.

- Screenshot 1:** Shows the 'gemma2' model listed under the 'Models' category. A red arrow points to the download button on the right side of the model card.
- Screenshot 2:** Shows the same model card with a progress bar indicating the download is at 31.8% completion. A red arrow points to the download button.
- Screenshot 3:** Shows the same model card with a green success message: '模型'gemma2'已成功下载。' (Model 'gemma2' has been successfully downloaded.)

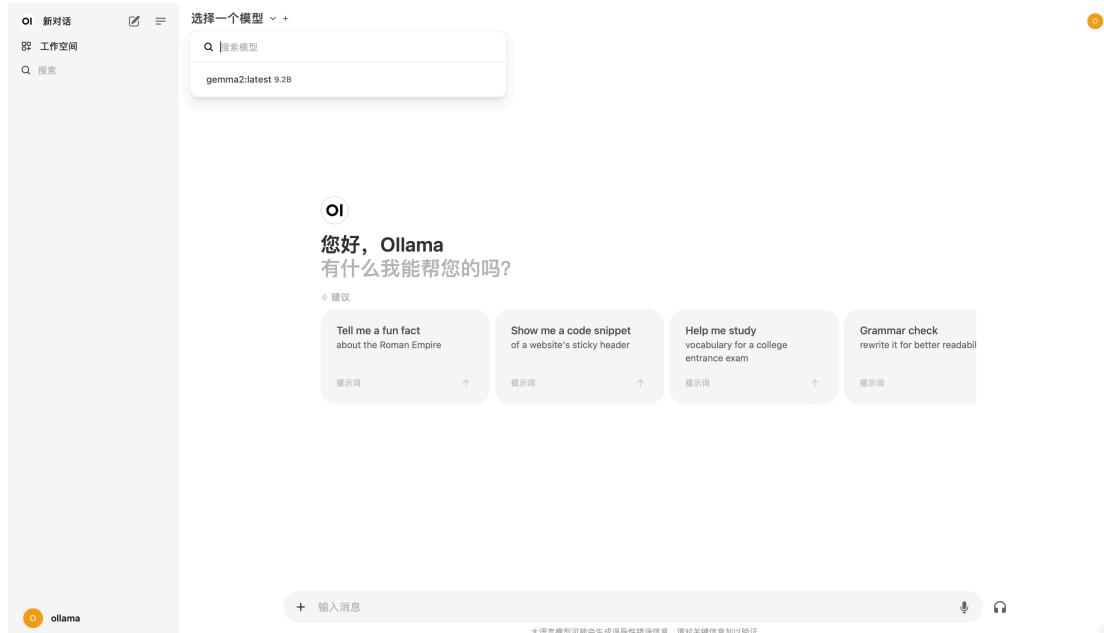
一定要耐心等到提示成功下载。然后可以去执行命令去论证下是否下载安装成功。

```
(base) root@RTX4090D-18400128:~# ollama list
NAME           ID      SIZE     MODIFIED
gemma2:latest  c19987e1e6e2  5.4 GB  12 seconds ago

(base) root@RTX4090D-18400128:~# du -sh /mnt/storage/ollama/
5.0G   /mnt/storage/ollama/
```

第 1 个命令是验证当前系统里的 ollama 已经成功安装了那些模型，第 2 个命令是去看在 5.2 中执行的 ollama 安装目录现在有多大了。

## 10.4 开始对话互动



点 web 页面左上角的“新对话”菜单，然后再点“选择一个模型”应该能看到刚刚下载安装成功的 gemma2，9.2B 就是这个模型的参数量。开始你的私有模型享受吧^\_^



写在最后：

这个文档适合于稍微有些 Linux 基础使用常识或者特别喜欢钻研的人。遇到问题的时候应该多去 google 搜，涉及的知识点已经精简的不能再少了。当然也希望官方能出个部署好的公共镜像，这样对于初学者上手来说就容易多了。