

基于华为开发者空间云主机的

Deepseek私有化部署

实验概览

实验目的

1. 理解 **deepseek** 的基本概念
2. 了解 **Ubuntu** 系统的基本用法并在云主机上配置实验环境
3. 掌握 **ollama** 和 **open-webui** 的使用方法
4. 学会如何使用 **ollama** 和 **open-webui** 在 **Ubuntu** 上私有化部署 **deepseek**

实验要求

1. 安装 **ollama** 并下载 **deepseek-R1** 模型，实现命令行交互
2. 安装 **miniconda**，并基于 **miniconda** 创建虚拟环境
3. 安装 **open-webui**，并实现 web 交互
4. 测试5-10 条跟自己专业相关的指令输入，结合专业知识思考现有模型存在的问题

提交内容

以 PDF 格式提交实验文档，内容包括：

测试指令及模型回答截图或记录、逐条指令的专业分析与评价，以及总结与改进建议。

提交地址

<https://icloud.qd.sdu.edu.cn:7777/link/74FA105BD804C362B68A3D22F1A1438C>

截止日期

2025-04-26

实验步骤

启动命令行

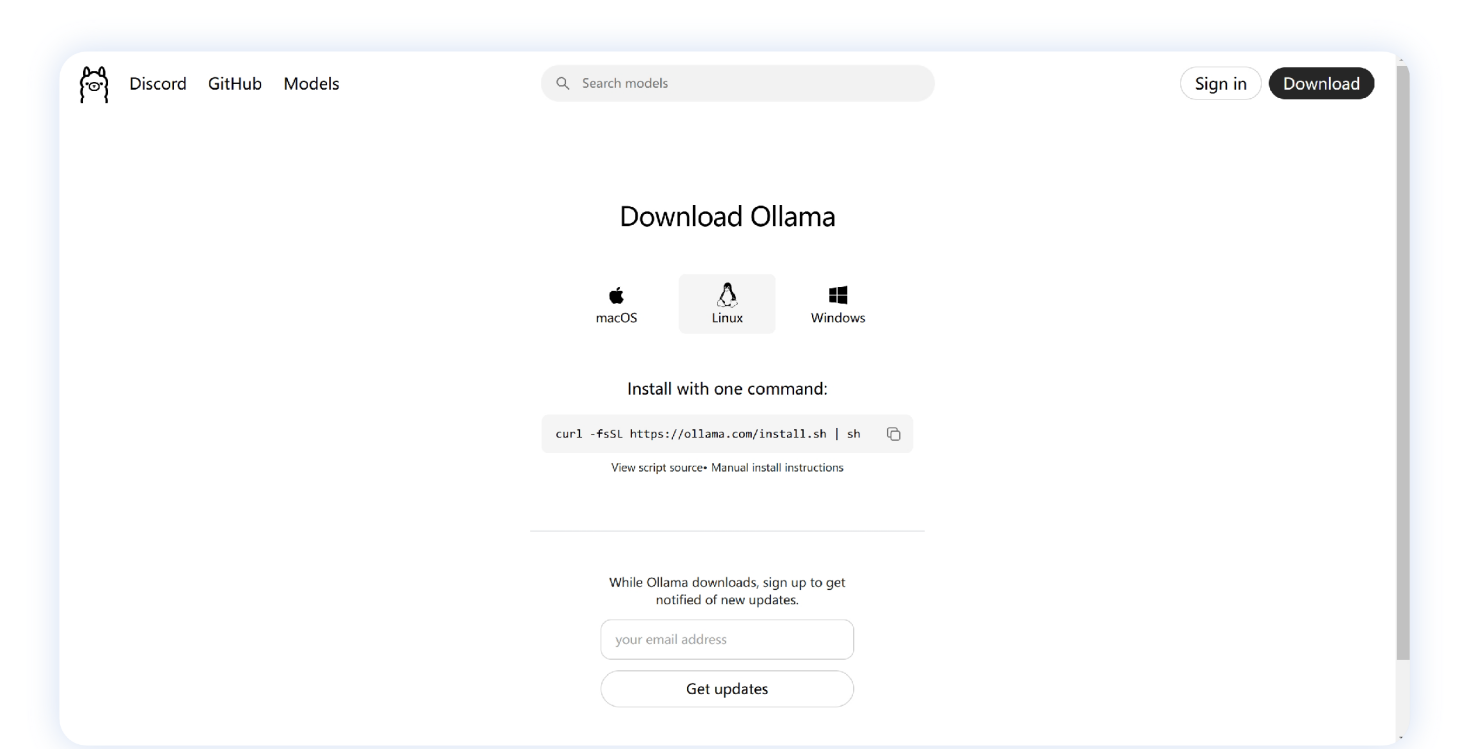
使用快捷键 `Ctrl+ALT+T` 打开命令行，这是用户与 Linux 系统进行交互的主要手段。

本实验会用到的命令如下：

命令	作用
<code>ls</code>	列出目录内容
<code>cd</code>	切换目录
<code>sudo</code>	以管理员身份运行指令
<code>curl</code> / <code>wget</code>	下载文件或者测试 api
<code>bash</code>	执行 <code>sh</code> 脚本

安装 ollama

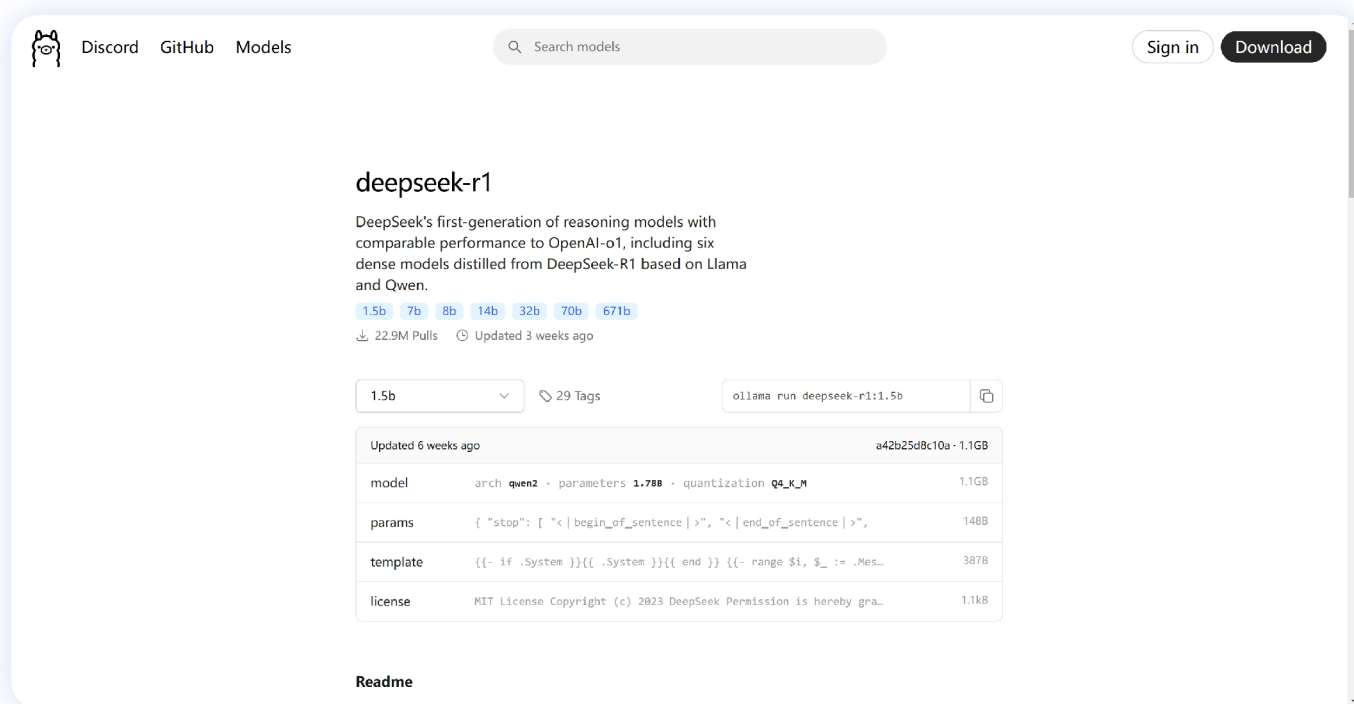
进入ollama 官网并选择Linux 系统



将图中连接输入到命令行中即可进行ollama 的安装。

本地部署 deepseek

在ollama 官网中搜索deepseek-r1 模型，模型大小选择1.5B。



在命令行中输入命令 `ollama run deepseek-r1:1.5b` 安装 deepseek-r1 1.5B 模型。

若命令行出现如下字段则安装完成，在 `>>>` 后输入内容即可与 deepseek 进行对话。

```
developer@developer:~$ ollama run deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8... 100% 1.1 GB
pulling 369ca498f347... 100% 387 B
pulling 6e4c38e1172f... 100% 1.1 KB
pulling f4d24e9138dd... 100% 148 B
pulling a85fe2a2e58e... 100% 487 B
verifying sha256 digest
writing manifest
success
>>> Send a message (/? for help)
```

使用快捷键 `Ctrl + D` 可以退出对话，再次在命令行中输入 `ollama run deepseek-r1:1.5b` 便再次进入对话模式。

安装 anaconda 并配置环境

1. 什么是 anaconda

Anaconda 是一个开源的Python/R 发行版和数据科学平台，专为科学计算、数据分析、机器学习和大规模数据处理设计。它集成了大量预装的科学计算库和工具，并提供 conda 包管理器和虚拟环境管理功能，简化了开发环境的配置流程。我们这里仅使用 conda 包管理器和虚拟环境管理功能。

使用 Python 编程时，我们经常会安装 Python 包辅助编程，如：numpy、matplotlib 等，在不同的项目中，我们也会使用不同的 Python 包，如果将他们安装在同一个环境下，极有可能出现冲突，导致很多难以解决的bug，因此我们需要借助 anaconda 创建虚拟环境，将不同项目的Python包安装在不同的虚拟环境中，避免冲突。

2. 安装 anaconda

访问miniconda 官网找到下载连接，在命令行中输入

```
wget https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-aarch64.sh
```

获取安装包。

检查安装包是否下载到本地，如果输入 ls 后可以在目录里看到安装的 sh 文件则下载成功。

```
developer@developer:~$ wget https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-aarch64.sh
--2025-03-06 16:19:45-- https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-aarch64.sh
正在解析主机 repo.anaconda.com (repo.anaconda.com)... 104.16.191.158, 104.16.32.241, 2606:4700::6810:20f1, ...
正在连接 repo.anaconda.com (repo.anaconda.com)|104.16.191.158|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 884179536 (843M) [application/octet-stream]
正在保存至： 'Anaconda3-2024.10-1-Linux-aarch64.sh'

Anaconda3-2024.10-1 100%[=====>] 843.22M 21.5MB/s 用时 42s

2025-03-06 16:20:28 (20.2 MB/s) - 已保存 'Anaconda3-2024.10-1-Linux-aarch64.sh'
[884179536/884179536])

developer@developer:~$ ls
1 icon Public
Anaconda3-2024.10-1-Linux-aarch64.sh IDEProjects run.log
DeployHuaweiCloudSession Music snap
Desktop node_modules Templates
Documents package.json Videos
Downloads package-lock.json
HuaweiCloudSession Pictures
```

此时，在命令行中输入 `bash Anaconda3-2024.10-1-Linux-aarch64.sh` 进行安装，按住 enter 浏览证书和安装协议直至出现下图字样

8. Pynacl. A Python binding to the Networking and Cryptography library, a crypto library with the stated goal of improving usability, security and speed.

9. Cryptography A Python library. This exposes cryptographic recipes and primitives.

10. Definitions.

1. "Anaconda Distribution", shortened form "Distribution", is an open-source distribution of Python and R programming languages for scientific computing and data science. It aims to simplify package management and deployment. Anaconda Distribution includes: (1) conda, a package and environment manager for your command line interface; (2) Anaconda Navigator; (3) 250 automatically installed packages; (3) access to the Anaconda Public Repository.

2. "Anaconda Navigator" means a graphical interface for launching common Python programs without having to use command lines, to install packages and manage environments. It also allows the user to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

3. "Anaconda Public Repository", means the Anaconda packages repository of 8000 open-source data science and machine learning packages at repo.anaconda.com.

Version 4.0 | Last Modified: March 31, 2024 | ANACONDA TOS

Do you accept the license terms? [yes|no]

>>>

如下图所示，在交互处输入 yes 接受证书和安装协议，并在安装目录处直接输入 enter 使用默认目录即可。

Do you accept the license terms? [yes|no]

>>> yes

Anaconda3 will now be installed into this location:

/home/developer/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/developer/anaconda3] >>>

PREFIX=/home/developer/anaconda3

Unpacking payload ...

如下图所示，直接输入 enter 选择 no，这里的选择是禁止 `conda` 环境自动启动，这是因为 `anaconda` 本身存在一个小bug，如果选择自动启动，在开启命令行时可能会同时启动两个环境。

```
zlib-ng          pkgs/main/linux-aarch64::zlib-ng-2.0.7-h998d150_0
zope             pkgs/main/linux-aarch64::zope-1.0-py312hd43f75c_1
zope.interface   pkgs/main/linux-aarch64::zope.interface-5.4.0-py312h998d150_0
_0
zstandard        pkgs/main/linux-aarch64::zstandard-0.23.0-py312hc476304_0
zstd             pkgs/main/linux-aarch64::zstd-1.5.6-h6a09583_0

Downloading and Extracting Packages:

Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated
.
If you'd prefer that conda's base environment not be activated on startup,
    run the following command when conda is activated:

conda config --set auto_activate_base false

You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>>
```

这样 `anaconda` 就安装完成了，但是此时在命令行输入 `conda` 命令却发现报错未找到命令，这是因为我们刚刚禁止 `conda` 环境自动启动，所以我们还需要手动为 `anaconda` 设置启动路径。

在命令行输入 `vim ~/.bashrc` 使用 `vim` 编辑器编辑 `.bashrc` 文件。`vim` 是一款经典的命令行编辑器，`.bashrc` 是管理命令行初始化的隐藏文件。

`vim` 共分为三种模式，命令模式、输入模式和底线命令模式。刚进入 `vim` 界面时，我们正处于命令模式，此状态下敲击键盘动作会被 `vim` 识别为命令，而非输入字符此时敲击键盘上 `Shift + G` 跳转光标到文件末尾。再敲击键盘上 `I` 便进入了输入模式，在文件末尾输入 `export PATH=$PATH:/home/developer/anaconda3/bin`。

```
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

source /usr/share/cangjie/envsetup.sh

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH=$PATH:/home/developer/anaconda3/bin
```

-- 插入 --

117,27

86%

此时敲击键盘 **Esc** 退出输入模式进入命令模式。在命令模式下，按下 **:** 进入底线命令模式，输入 **wq** 保存修改并退出 **vim**。


```
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

source /usr/share/cangjie/envsetup.sh

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH=$PATH:/home/developer/anaconda3/bin

:wq
```

此时 `anaconda` 路径已经配置完毕，在命令行中输入 `source ~/.bashrc` 执行一遍 `.bashrc`，使得刚刚的修改生效。随后输入 `conda init` 进行 `conda` 的初始化，随后按照指引重启命令行。

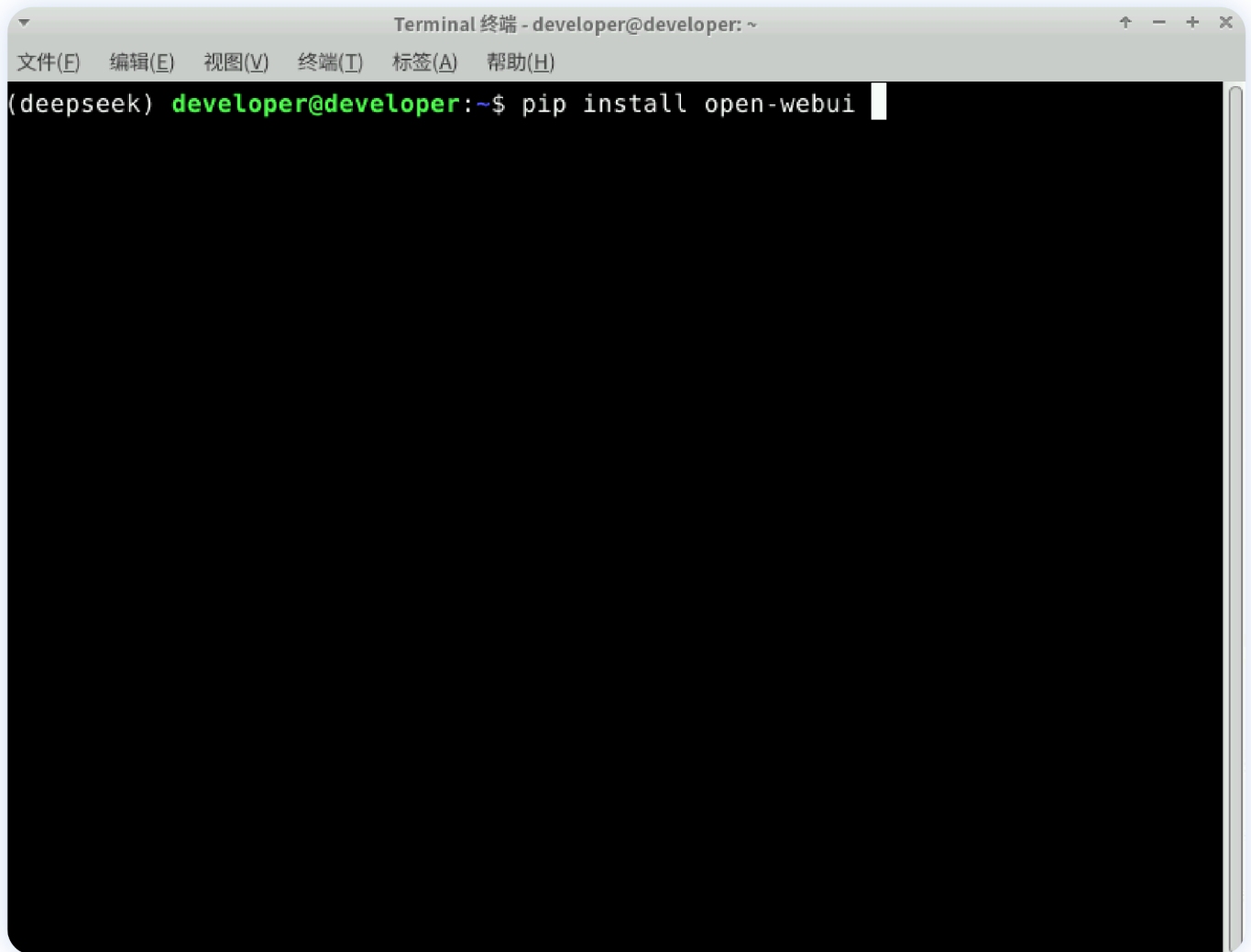
若下图所示，路径前面出现 (base) 字样，则代表 `conda` 环境已经成功激活。

```
(base) developer@developer:~$
```

3. 配置环境

在命令行中输入 `conda create -n deepseek python=3.11` 创建一个 python 版本为 3.11 名称为 **deepseek** 的虚拟环境。

创建完毕后，输入 `conda activate deepseek` 激活 **deepseek** 环境，此时如果路径前面的 (base) 变成 (deepseek) 则代表环境激活成功。

A screenshot of a terminal window titled "Terminal 终端 - developer@developer: ~". The window has a menu bar with options: 文件(E), 编辑(E), 视图(V), 终端(T), 标签(A), 帮助(H). The terminal prompt is "(deepseek) developer@developer:~\$". The command "pip install open-webui" has been entered, and a white cursor is visible at the end of the command line.

```
Terminal 终端 - developer@developer: ~
文件(E) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
(deepseek) developer@developer:~$ pip install open-webui
```

在命令行中输入 `pip install open-webui` 安装 `open-webui` 包。安装过程比较久，请耐心等待... 如果安装过程中卡住，可以使用 `Ctrl + C` 强制终止命令，并重新输入安装命令。

实现 web 交互

安装完毕后，在命令行中输入 `open-webui serve` 命令，此时便启动了 `webui` 的后端程序。由于服务器没有使用 vpn 服务，这一步可能报错连接失败，此时可以重启输入 `open-webui serve` 命令，或者多等待一会，直到出现图中绿色字体 INFO 的字样。

```

Traceback (most recent call last):
  File "/home/developer/anaconda3/envs/deepseek/lib/python3.11/site-packages/open_webui/retrieval/utils.py", line 543, in get_model_path
    model_repo_path = snapshot_download(**snapshot_kwargs)
                        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/developer/anaconda3/envs/deepseek/lib/python3.11/site-packages/huggingface_hub/utils/_validators.py", line 114, in _inner_fn
    return fn(*args, **kwargs)
            ^^^^^^^^^^^^^^^^^
  File "/home/developer/anaconda3/envs/deepseek/lib/python3.11/site-packages/huggingface_hub/_snapshot_download.py", line 235, in snapshot_download
    raise LocalEntryNotFoundError(
huggingface_hub.errors.LocalEntryNotFoundError: An error happened while trying to locate the files on the Hub and we cannot find the appropriate snapshot folder for the specified revision on the local disk. Please check your internet connection and try again.
WARNI [sentence_transformers.SentenceTransformer] No sentence-transformers model found with name sentence-transformers/all-MiniLM-L6-v2. Creating a new one with mean pooling.
INFO:      Started server process [34747]
INFO:      Waiting for application startup.
2025-03-06 17:36:20.696 | INFO      | open_webui.utils.logger:start_logger:140 -
GLOBAL_LOG_LEVEL: INFO - {}
2025-03-06 17:36:26.396 | INFO      | uvicorn.protocols.http.httptools_impl:send

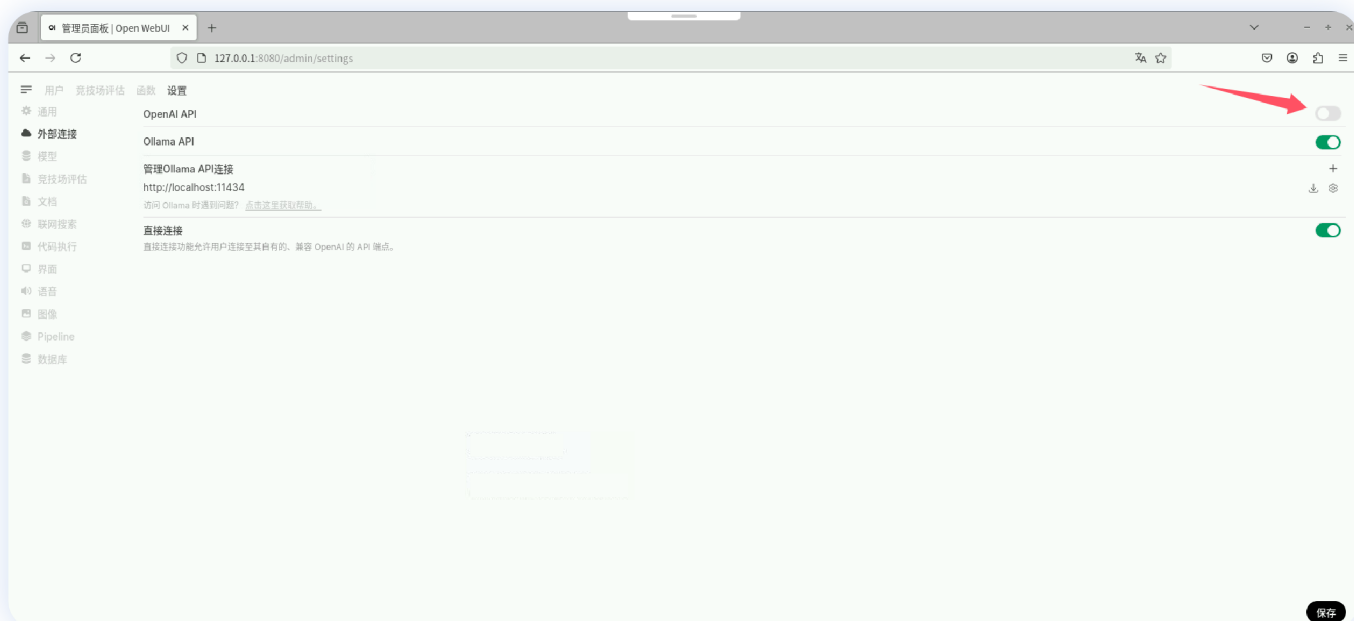
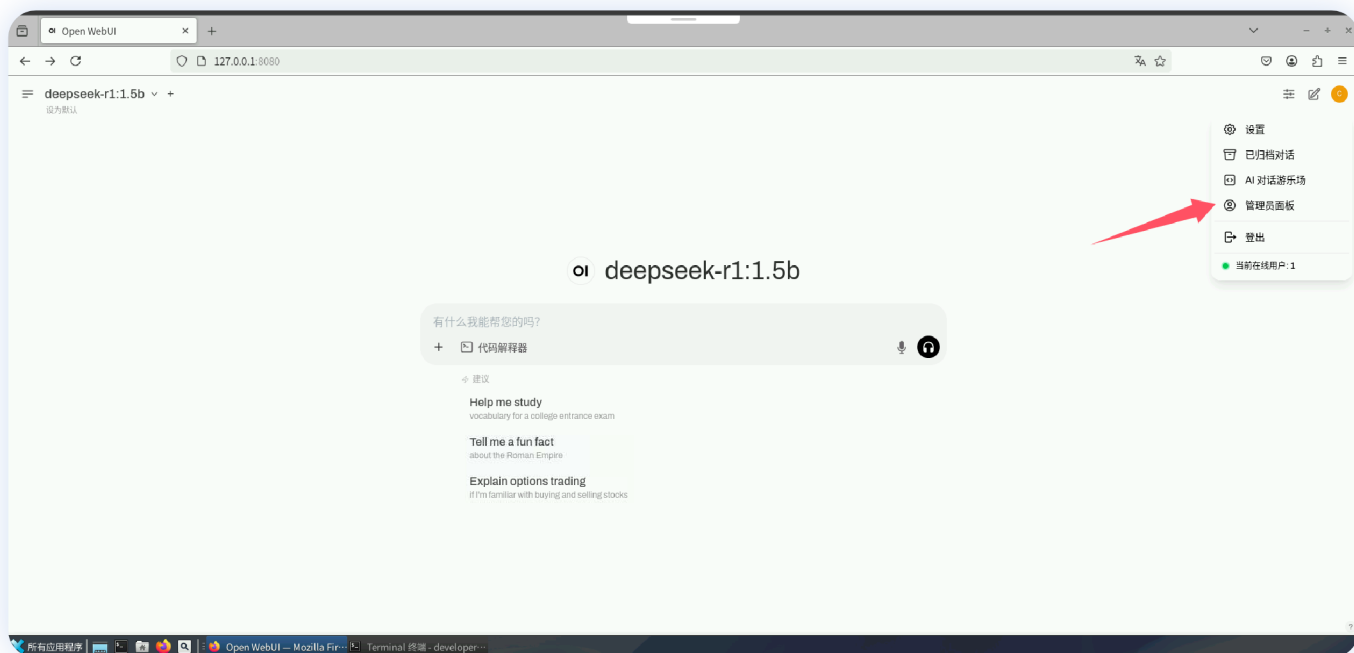
```

打开浏览器，访问<http://localhost:8080/>，进入前端 web 页面。如下图所示，注册一个管理员账号，并记住账号密码。



点击创建管理员账号后需要等待 **open-webui** 访问 api。**open-webui** 默认访问 **openai.com** 获取 api，连接国内的网络无法直接获取，因此需要等待较长一段时间，等待 **openai.com** 申请超时。

如图所示，进入页面后，我们需要进入 **管理员面板->设置->外部连接**，关闭 OpenAI API，这样下次进入就会默认调用 ollama API，无需再等待申请超时。



返回工作空间，可以发现 **open-webui** 已经自动调用了 ollama API 加载 **deepseek-r1-1.5B** 模型，在对话框中输入问题，可以正确返回回答。这样我们就完成了web 交互。