Running Ollama with Open WebUI on Intel Hardware Platform

Installation Guide

June 2024

Document Number: 826081-1.0

Contents

Figures

§

1.0 Introduction

This guide is to help users install and run Ollama with Open WebUI on Intel Hardware Platform on **Windows* 11** and **Ubuntu* 22.04 LTS**.

Throughout this guide, you will learn to:

1. Install Intel GPU Driver and oneAPI Base Toolkit (Windows / Ubuntu).
2. Install Python* 3.11 (Windows / Ubuntu).
3. Install IPEX-LLM for Ollama (Windows / Ubuntu).
4. Install Open WebUI (Windows / Ubuntu).
5. Configure Open WebUI with Ollama.

1.1 Verified Hardware

System Verified Hardware:

- CPU: Intel® Core™ i9-13900K Processor
- GPU: Intel® Arc™ A770 Graphics
- RAM: 32 GB RAM

2.0 Windows 11 Installation Guide

Note: If you are using the Ubuntu operating system, skip this section, and move to section 3.0.

2.1 Step 1: Installing Intel GPU Driver and oneAPI Base Toolkit (Windows)

2.1.1 Installing GPU Driver

Click the link to download and install the latest Intel® Arc™ Graphics GPU Driver from the official Intel download page. A system reboot is necessary after the installation process is complete.

Figure 1: Intel® Arc™ Graphics GPU Driver Official Download Page

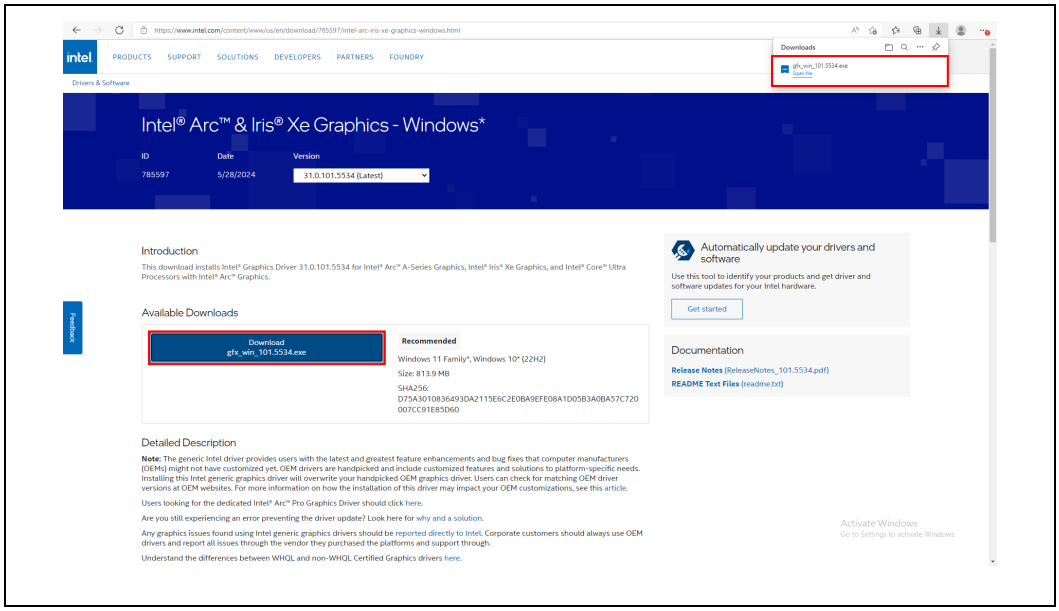
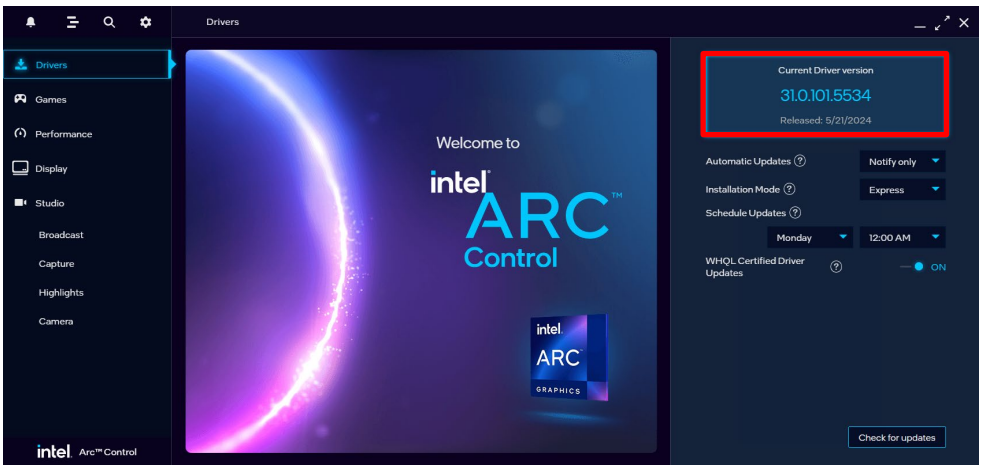


Figure 2: Intel® Arc™ Graphics GPU Driver Installation

2.1.1.1 Successful Installation

Figure 3: Intel® Arc™ Graphics Driver Dashboard



Note: Ensure that your GPU driver is updated to version 31.0.101.5333 or latest, to meet the installation requirements.

2.1.2 Installing Intel oneAPI Base Toolkit

To install the Intel oneAPI Base Toolkit, click the link to begin downloading and installing.

Figure 4: Intel oneAPI Base Toolkit Download Page

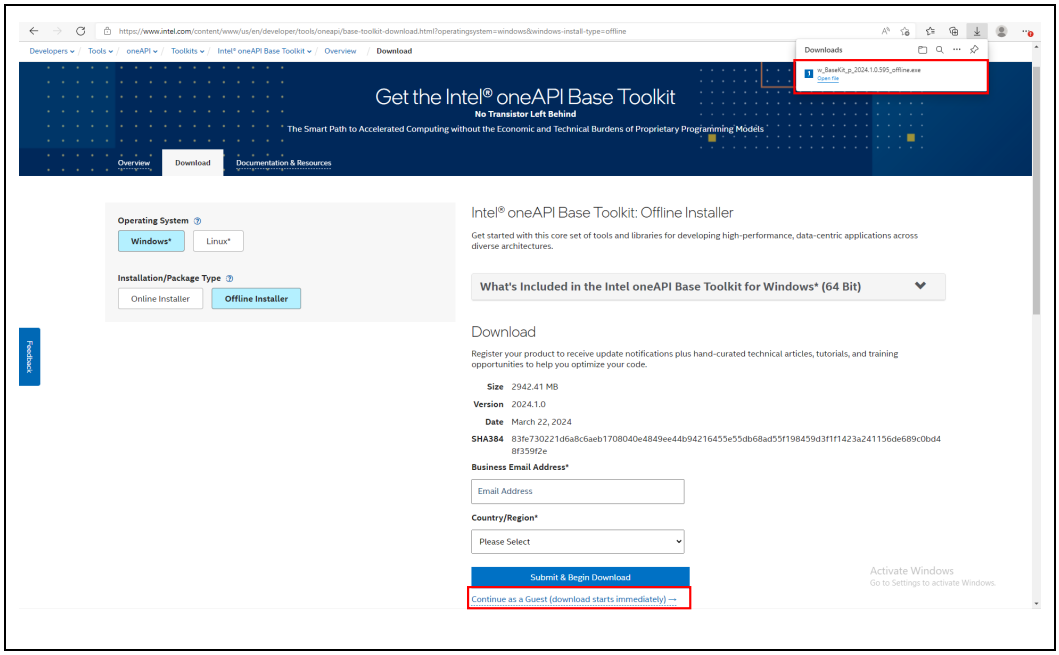


Figure 5: Installation of Intel oneAPI Base Toolkit

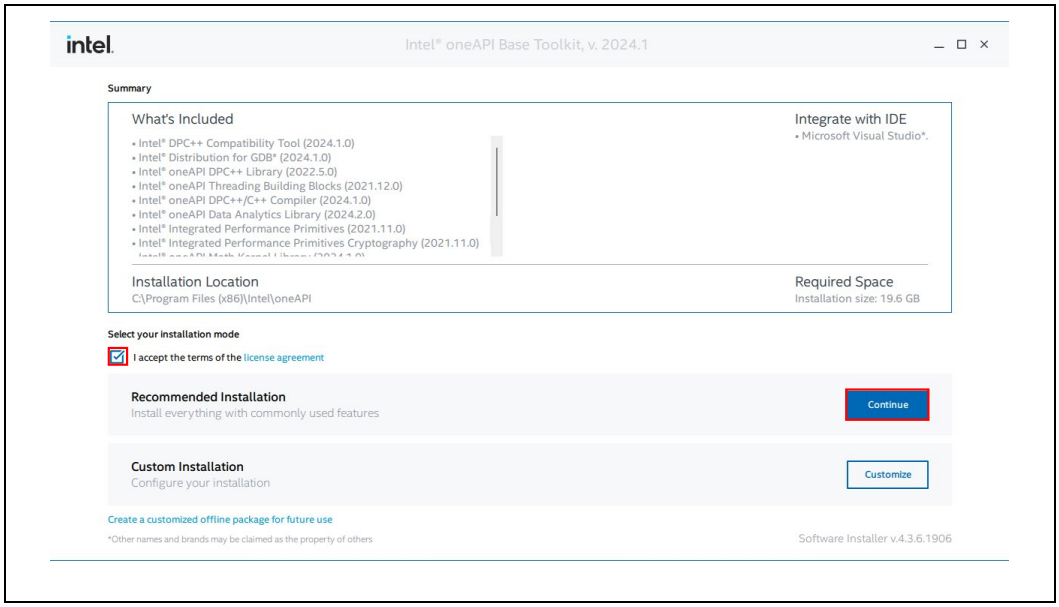


Figure 6: Installation of Intel oneAPI Base Toolkit

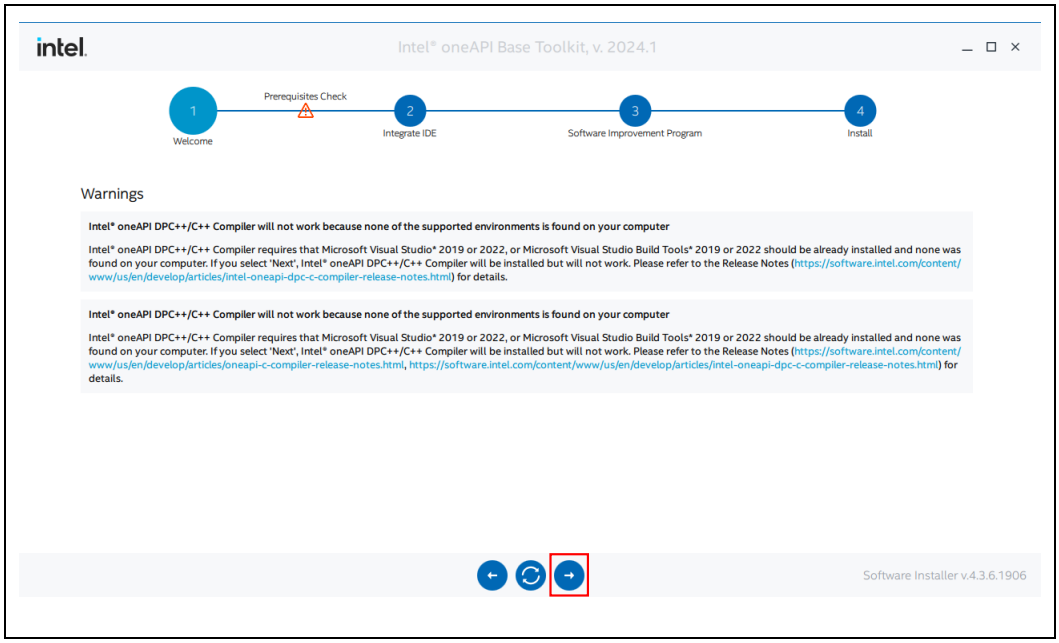
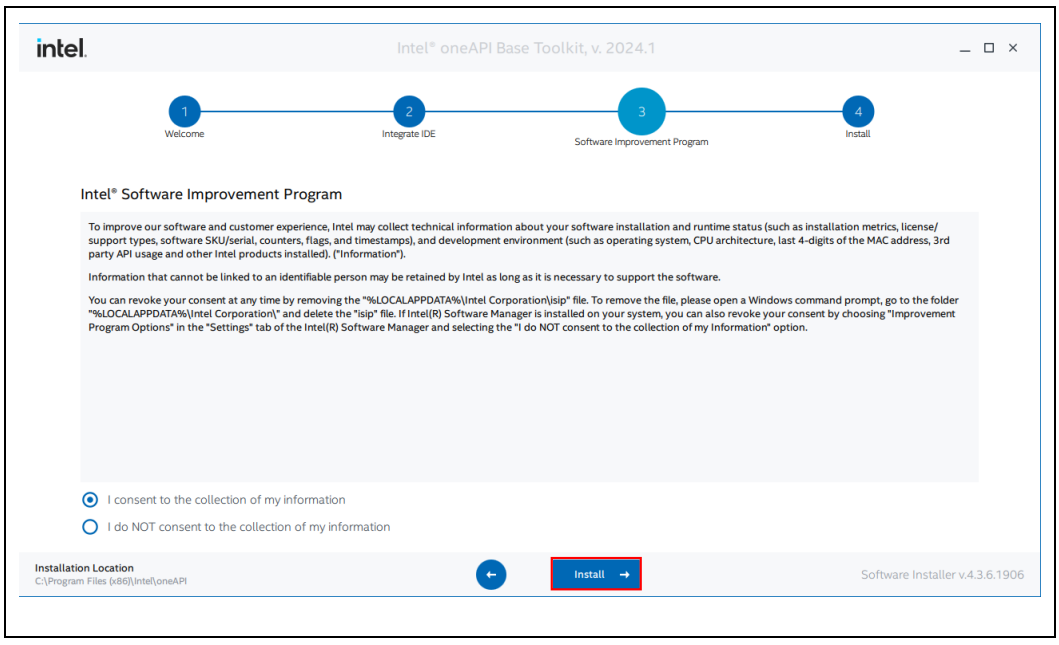
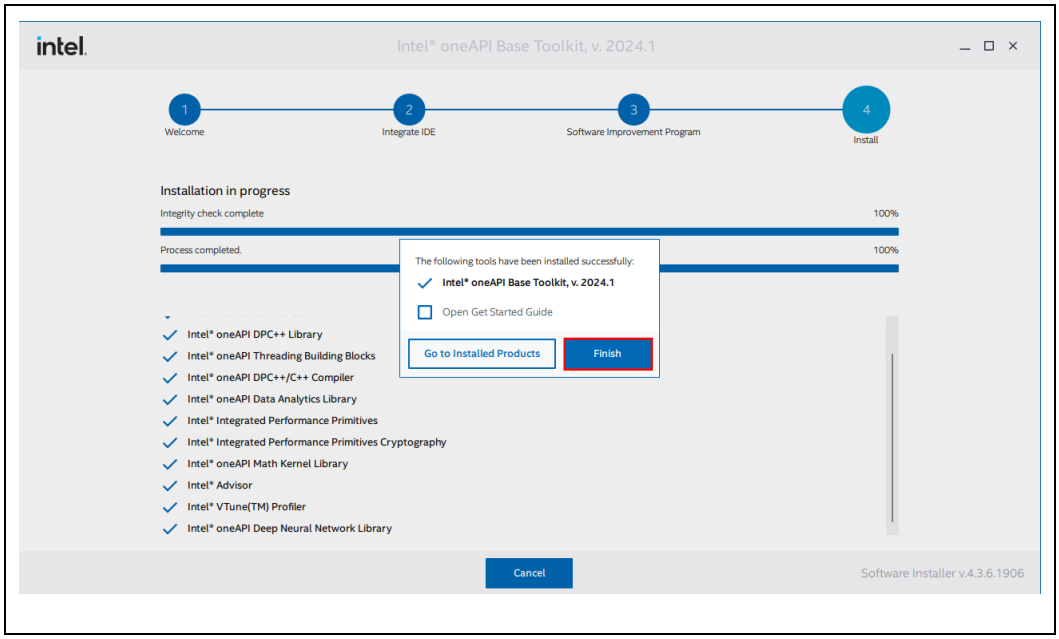


Figure 7: Installation of Intel oneAPI Base Toolkit



2.1.2.1 Successful Installation

Figure 8: Successful Installation of Intel oneAPI Base Toolkit



2.2 Step 2: Installing Python 3.11 (Windows)

To begin installing Python 3.11 for Windows, click the link to download and install Python directly from the official Python download page. Ensure that the option to add python.exe to PATH is enabled before installing.

Figure 9: Python 3.11 Official Download Page

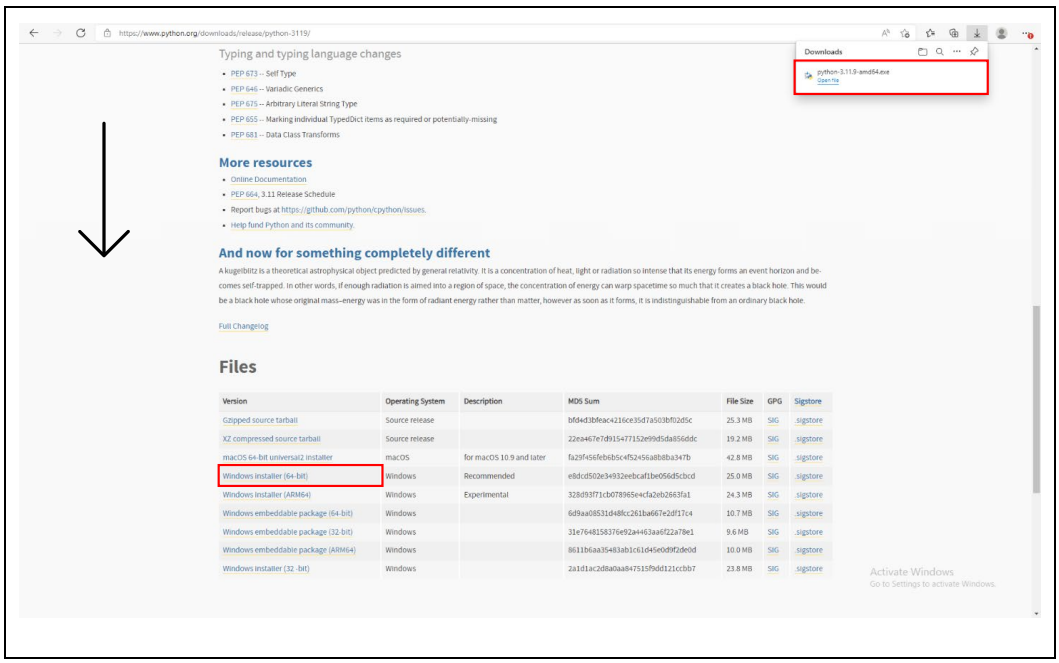


Figure 10: Python 3.11 Installation Process

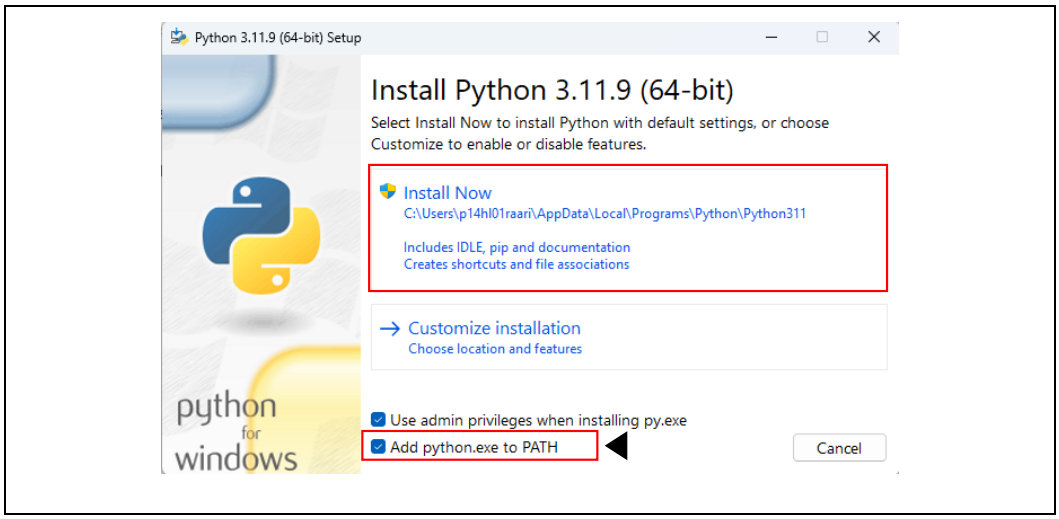


Figure 11: Python 3.11 Installation Process

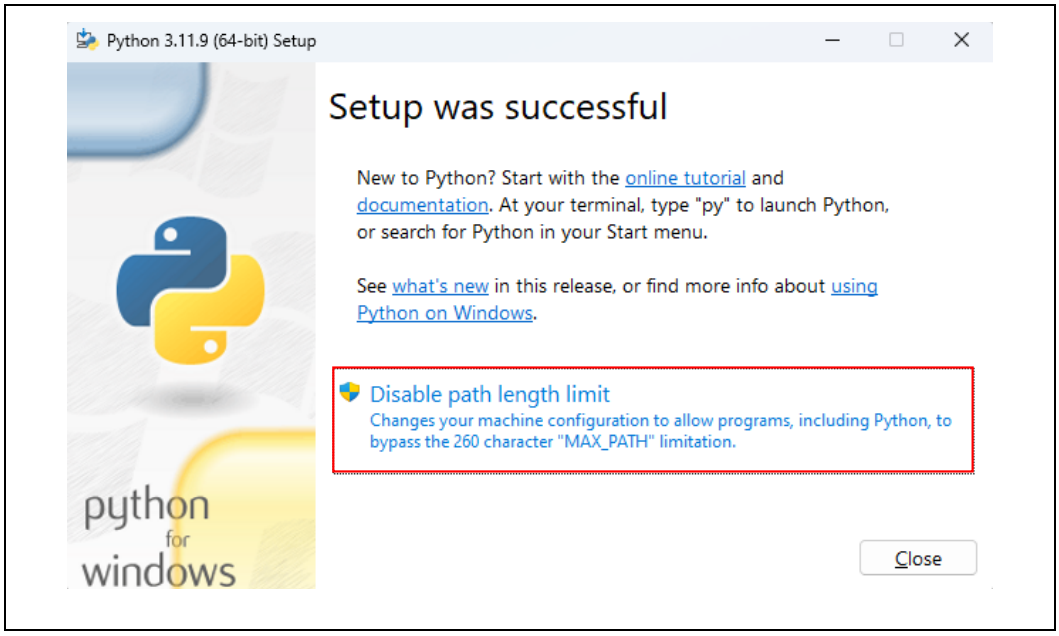
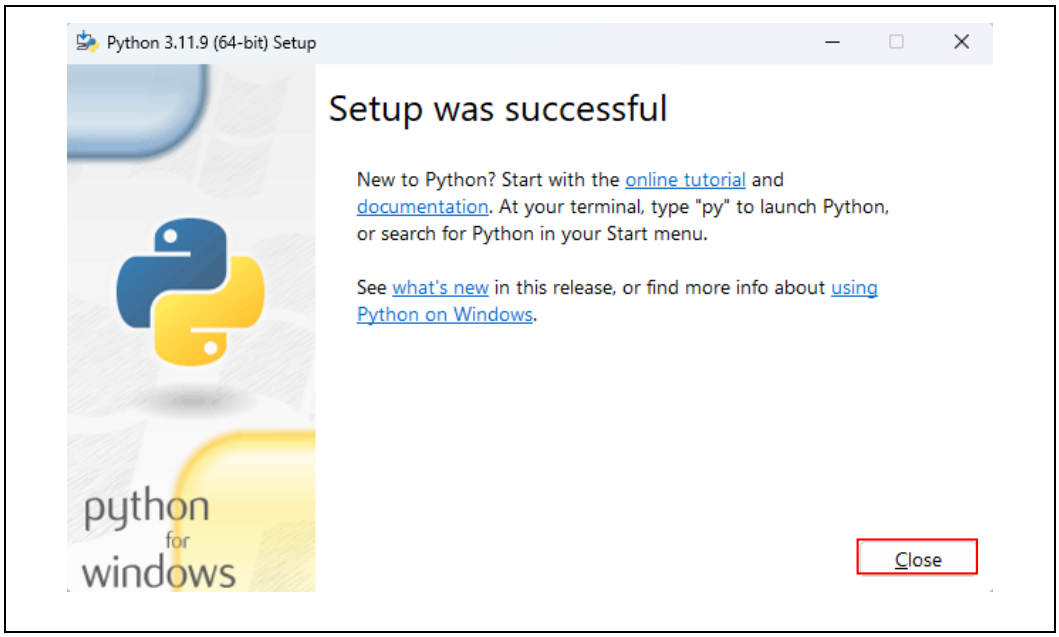


Figure 12: Python 3.11 Installation Process



2.2.1 Successful Installation

To ensure the successful installation of Python 3.11, follow these steps:

1. Open the Start menu and type **cmd** into the search bar.
2. Right-click on "Command Prompt" in the search results.
3. Select "Run as administrator".
4. Execute the command: **$** ***python -V***

Figure 13: Steps on Opening Command Prompt as Administrator

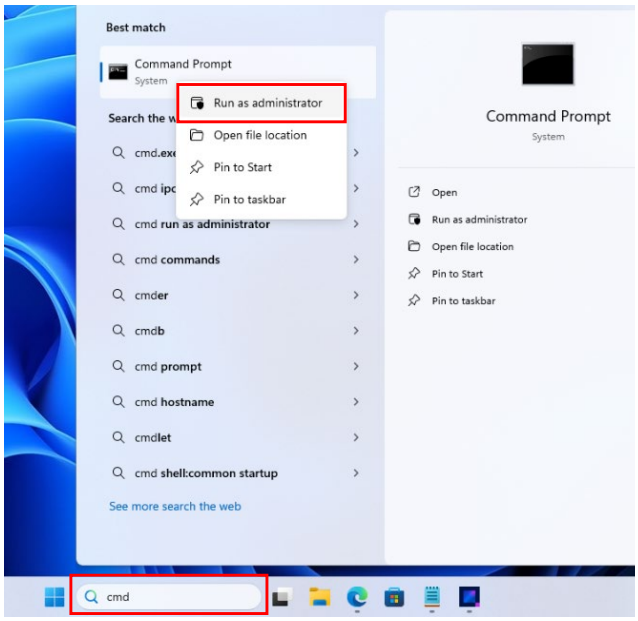
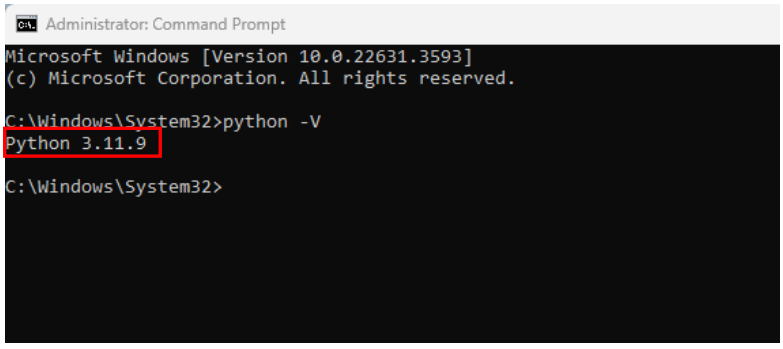


Figure 14: Command Prompt Output to Ensure Successful Installation of Python 3.11



2.3 Step 3: Installing IPEX-LLM for Ollama (Windows)

2.3.1 Installing and Initializing Ollama with Intel GPU

To use Ollama with Intel GPU, ensure that ipex-llm[cpp] is installed. To achieve this, enter the command below onto the same command prompt from before with administrator mode to install and initialize Ollama.

```
$ python -m venv llm_env
$ C:\Windows\System32\llm_env\Scripts\activate.bat
$ pip install --pre --upgrade ipex-llm[cpp]
$ mkdir llama-cpp
$ cd llama-cpp
$ init-ollama.bat
```

2.3.2 Run Ollama Serve with Intel GPU

You may execute the commands below to launch the Ollama service:

```
$ "C:\Program Files (x86)\Intel\oneAPI\setvars.bat"
$ set OLLAMA_NUM_GPU=999
$ set no_proxy=localhost,127.0.0.1
$ set ZES_ENABLE_SYSMAN=1
$ set SYCL_CACHE_PERSISTENT=1
$ ollama serve
```

Note: Set the OLLAMA_NUM_GPU to 999 to ensure all layers of your models are running on Intel GPU, otherwise, some layers may run on CPU.

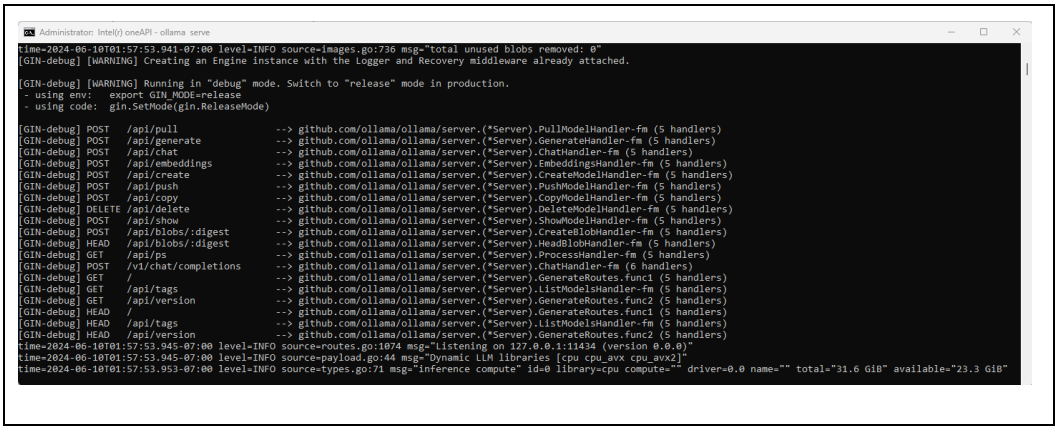
Note: If you're under a network proxy, execute the following command before running ollama serve to set up network proxy for Ollama:

```
$ set HTTP_PROXY=http://proxy.com:port
$ set HTTPS_PROXY=http://proxy.com:port
```

2.3.3 Successful Installation

The console will display message similar to the following indicating a successful launch of the ollama service. Ensure that this terminal is not closed:

Figure 15: Successful Installation of Ollama Service



2.4 Step 4: Installing Open WebUI (Windows)

2.4.1 Install Open WebUI

Open a new command prompt window with **administrator mode** and run the following command:

```
$ C:\Windows\System32\llm_env\Scripts\activate.bat
$ pip install open-webui==0.2.5
```

2.4.2 Start Open WebUI Server

Once installed, you may start the server using the following command:

```
$ open-webui serve
```

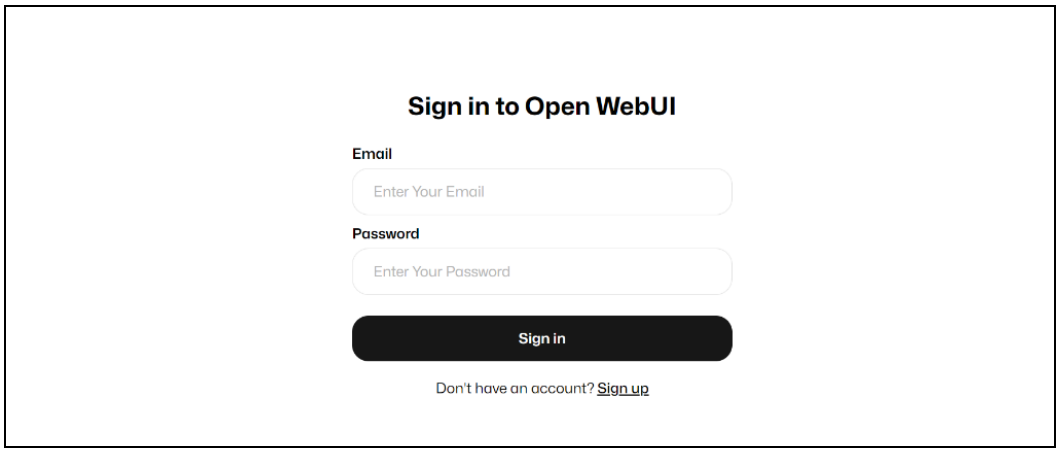
Note: If you're under a network proxy, execute the following command before running open-webui serve to set up network proxy for Open WebUI:

```
$ set http_proxy=http://proxy.com:port
$ set https_proxy=http://proxy.com:port
$ set no_proxy=localhost,127.0.0.1
```

2.4.3 Successful Installation

In this step, you may access the Open WebUI using http://localhost:8080 once the installation is successful.

Figure 16: Successful Installation of Open WebUI



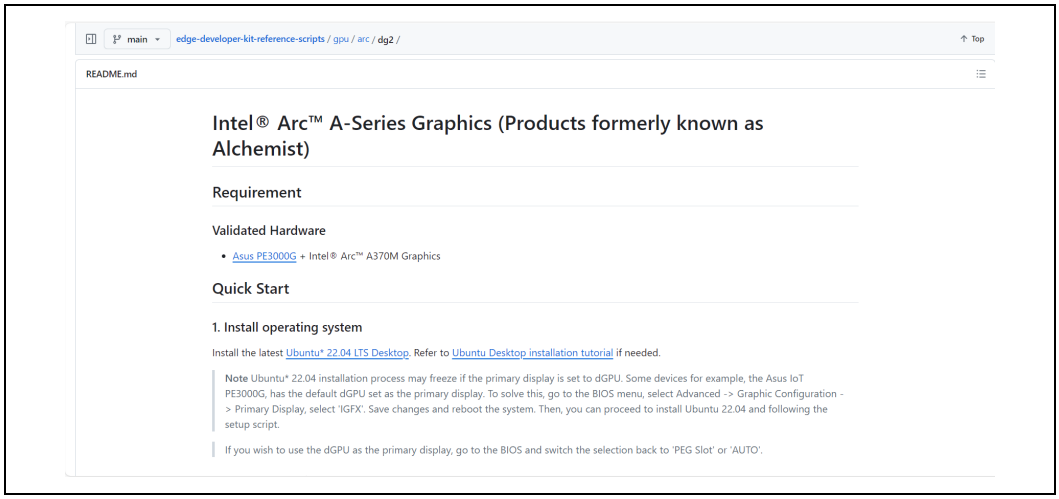
3.0 Ubuntu Installation Guide

3.1 Step 1: Installing Intel GPU Driver and oneAPI Base Toolkit (Ubuntu)

3.1.1 Installing GPU Driver

To start, click the link and follow the steps in the README.md to begin installing the Intel® Arc™ Graphics GPU Driver.

Figure 17: GPU Driver Installation Page



3.1.2 Installing Intel oneAPI Base Toolkit

To install the Intel oneAPI Base Toolkit, click the link and follow the steps provided on the webpage.

Figure 18: Intel oneAPI Base Toolkit Download Page

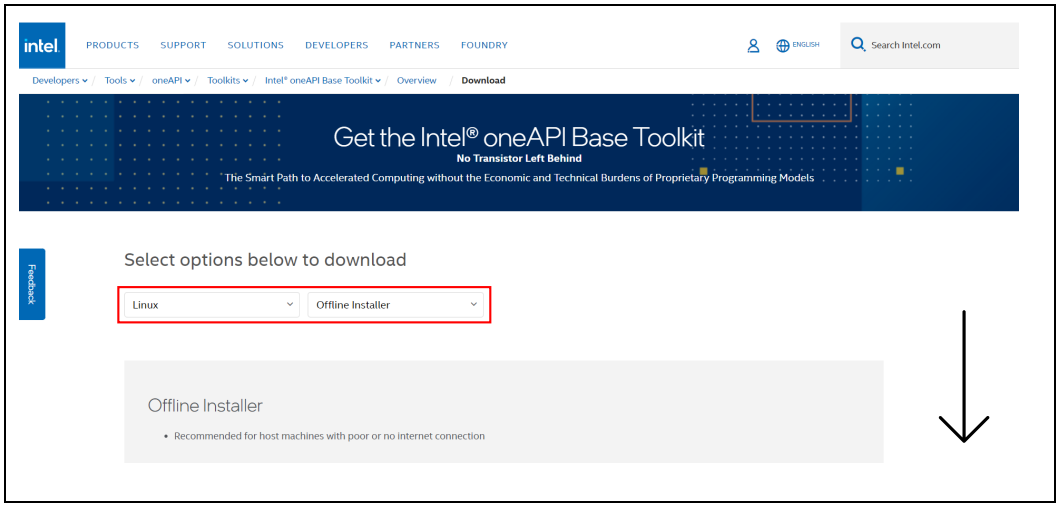


Figure 19: Intel oneAPI Base Toolkit Command Line Guide

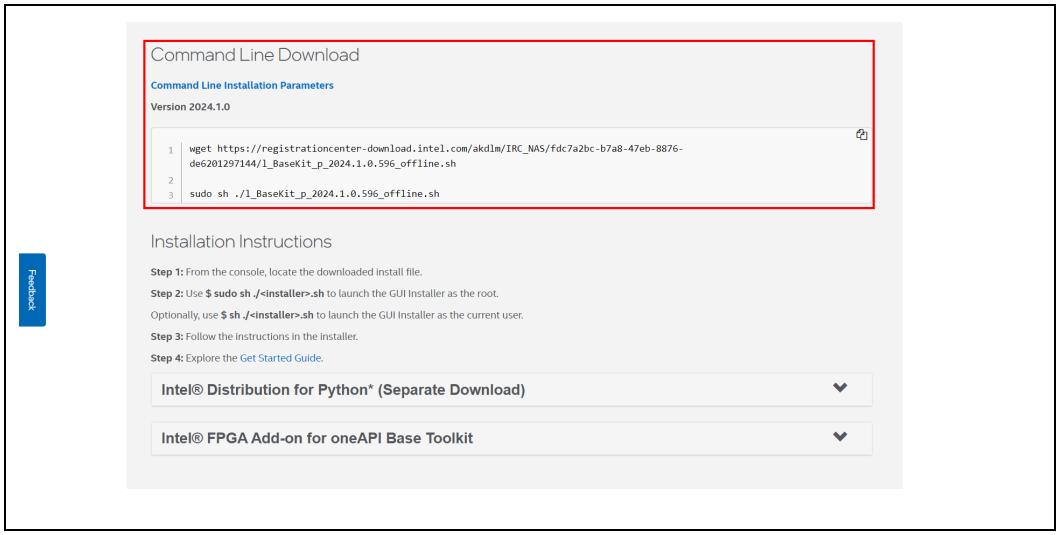


Figure 20: Installation of Intel oneAPI Base Toolkit

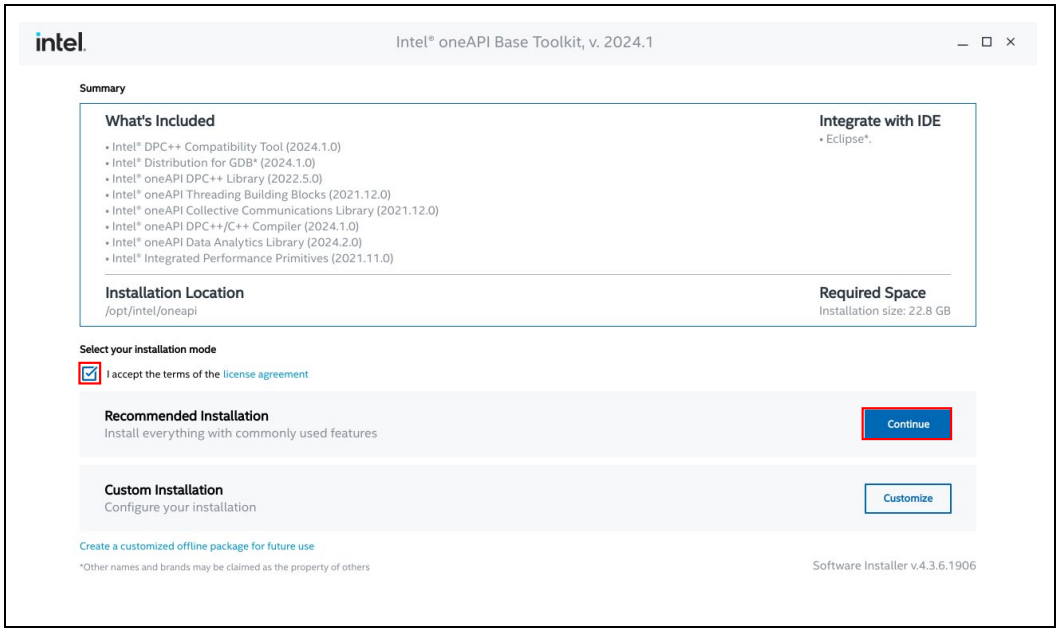
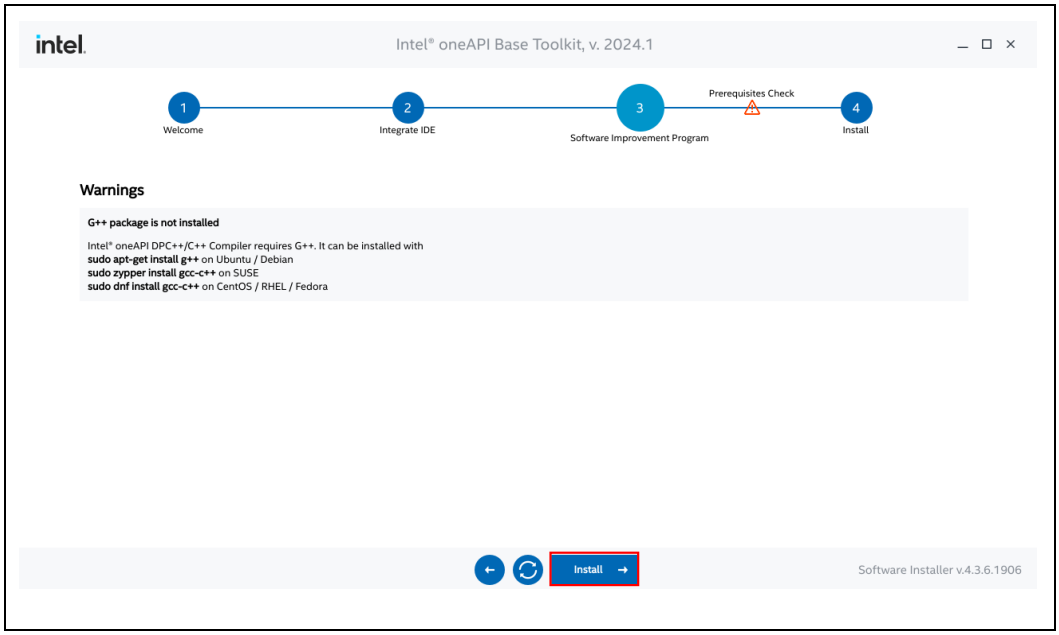
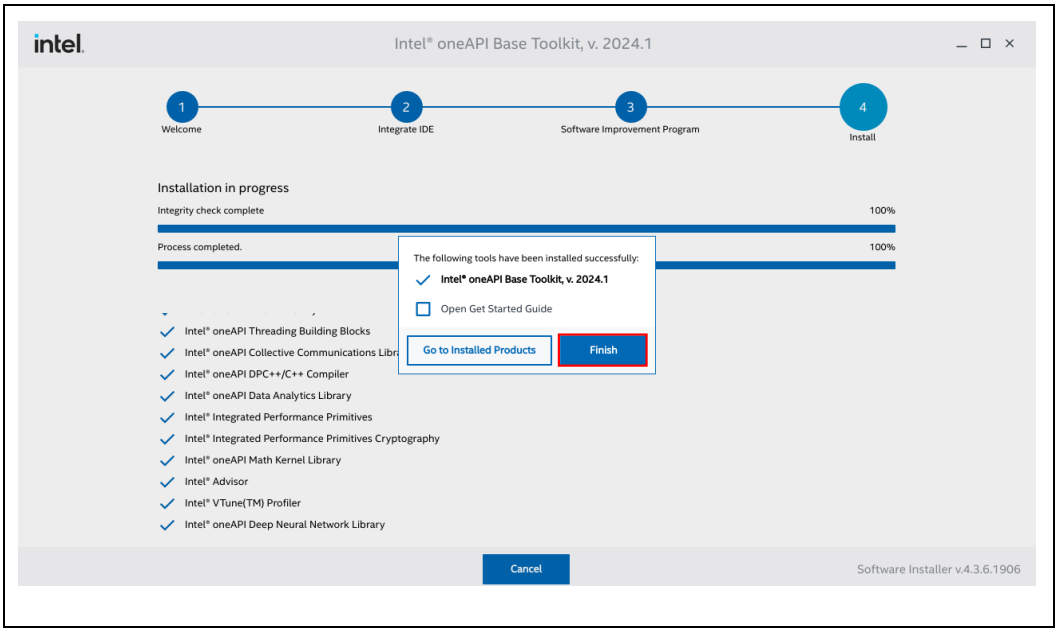


Figure 21: Installation of Intel oneAPI Base Toolkit



3.1.2.1 Successful Installation

Figure 22: Successful Installation of Intel oneAPI Base Toolkit



3.2 Step 2: Installing Python 3.11 (Ubuntu)

3.2.1 Update System Packages

Next, update the system packages in the Ubuntu 22.04 terminal using:

```
$ sudo apt update
$ sudo apt upgrade
```

3.2.2 Adding PPA Repository

Add the deadsnakes PPA repository to the system:

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
```

3.2.3 Python 3.11 Installation

Use this command to install Python 3.11 and required dependencies on Ubuntu 22.04:

```
$ sudo apt install python3.11 -y
$ sudo apt install python3.11-venv -y
```

3.2.4 Validate Python Version

Use this command to validate Python 3.11 installation on Ubuntu 22.04:

```
$ python3.11 -V
```

Output:

```
Python 3.11.xx
```

3.3 Step 3: Installing IPEX-LLM for Ollama (Ubuntu)

3.3.1 Installing and initializing Ollama with Intel GPU

To use Ollama with Intel GPU, ensure that ipex-llm[cpp] is installed. To achieve this, enter the command below onto the terminal to install and initialize Ollama.

```
$ python3.11 -m venv llm_env
$ source llm_env/bin/activate
$ pip install --pre --upgrade ipex-llm[cpp]
$ mkdir llama-cpp
$ cd llama-cpp
$ init-ollama
```

3.3.2 Run Ollama Serve with Intel GPU

You may execute the commands below to launch the Ollama service:

```
$ export OLLAMA_NUM_GPU=999
$ export no_proxy=localhost,127.0.0.1
$ export ZES_ENABLE_SYSMAN=1
$ source /opt/intel/oneapi/setvars.sh
$ export SYCL_CACHE_PERSISTENT=1
$ ./ollama serve
```

Note: Set the **OLLAMA_NUM_GPU** to **999** to ensure all layers of your models are running on Intel GPU, otherwise, some layers may run on CPU.

Note: To allow the service to accept connections from all IP addresses, use:

```
$ OLLAMA_HOST=0.0.0.0 ./ollama serve
```

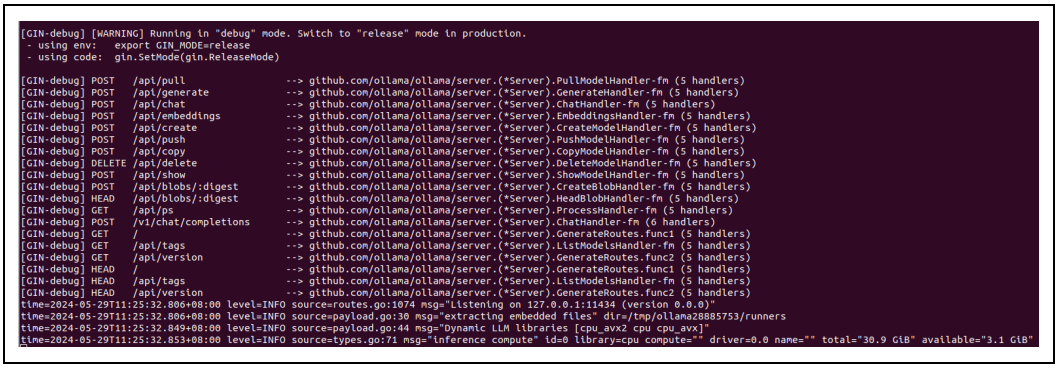
Note: If your local LLM is running on Intel® Arc™ A-Series Graphics with Linux OS (Kernel 6.2), it is recommended to additionally add the following environment command for optimal performance before executing ./ollama serve:

```
$ export SYCL_PI_LEVEL_ZERO_USE_IMMEDIATE_COMMANDLISTS=1
```

3.3.3 Successful Installation

The console will display message similar to the following indicating a successful launch of the ollama service. Ensure that this terminal is not closed:

Figure 23: Successful Installation of Ollama Service



3.4 Step 4: Installing Open WebUI (Ubuntu)

3.4.1 Install Open WebUI

Open a new terminal using **CTRL + ALT + T** and run the following command:

```
$ source llm_env/bin/activate
$ pip install open-webui==0.2.5
```

3.4.2 Start Open WebUI Server

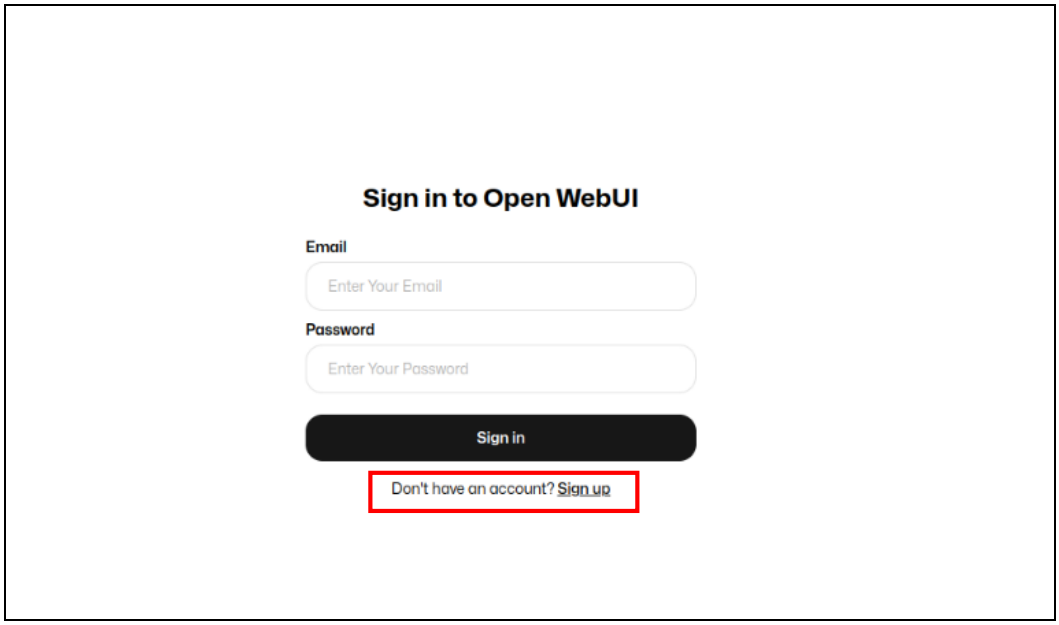
Once installed, you may start the server using the following command:

```
$ open-webui serve
```

3.4.3 Successful Installation

In this step, you may access the Open WebUI using http://localhost:8080 once the installation is successful.

Figure 24: Successful Installation of Open WebUI



4.0 Configuring Open WebUI with Ollama

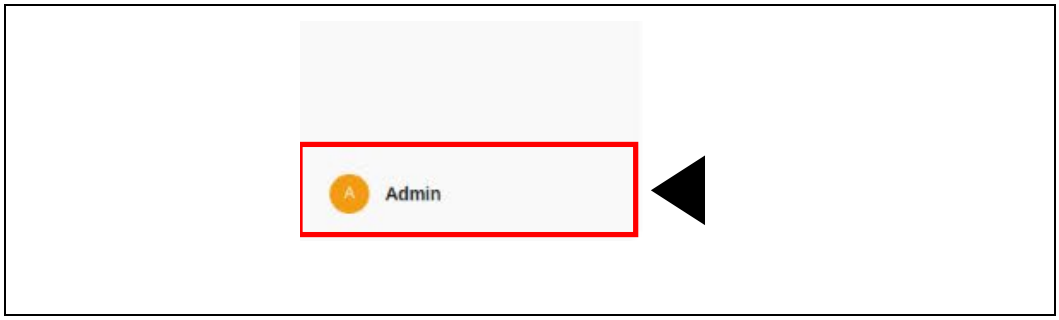
4.1 Step 5: Configuring Open WebUI with Ollama

4.1.1 Setting Up Ollama with Open WebUI

To set up Ollama with Open WebUI, once logged in, follow the steps below:

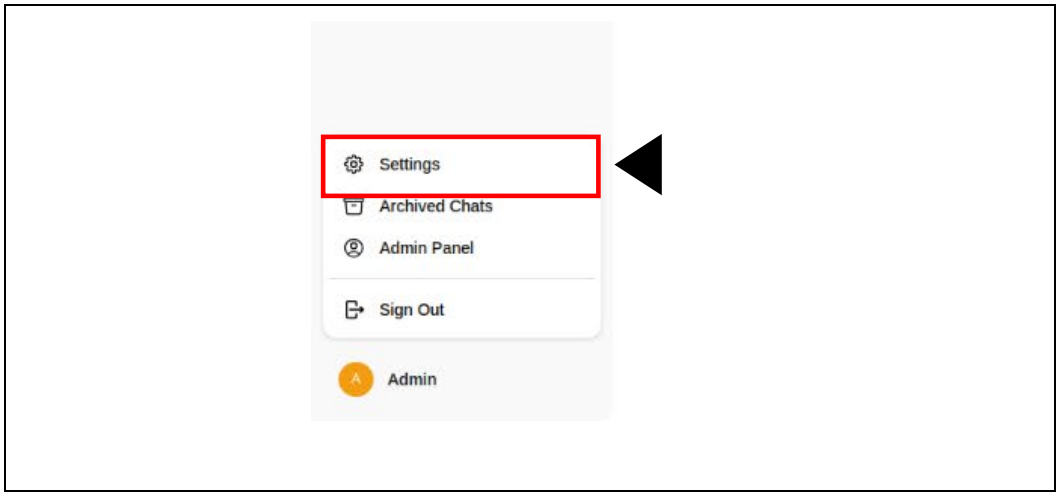
1. Click on your profile at the bottom left of the page.

Figure 25: Step 1 of Setting Up Ollama with Open WebUI



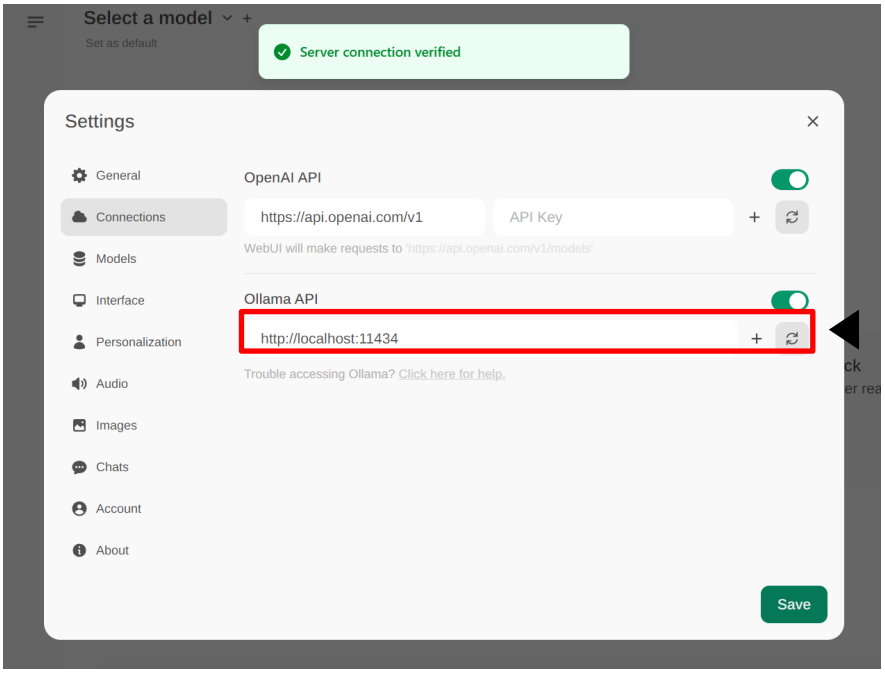
2. Navigate to Settings

Figure 26: Step 2 of Setting Up Ollama with Open WebUI



3. Go to Connections to link the Open WebUI with your Ollama service

Figure 27: Step 3 of Setting Up Ollama with Open WebUI



4.2 Download Model and Inferencing

To download your desired model, you may navigate to **settings > models** to pull your model and start to perform inferences once the model has completed downloading:

Figure 28: Downloading Model for Inferencing

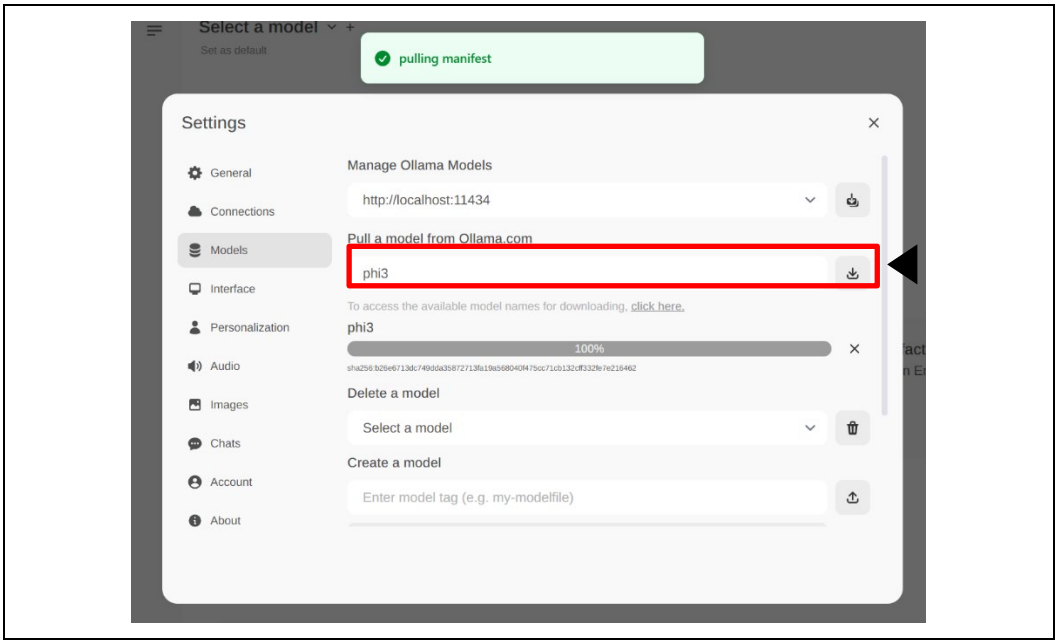


Figure 29: Selecting Model for Inferencing

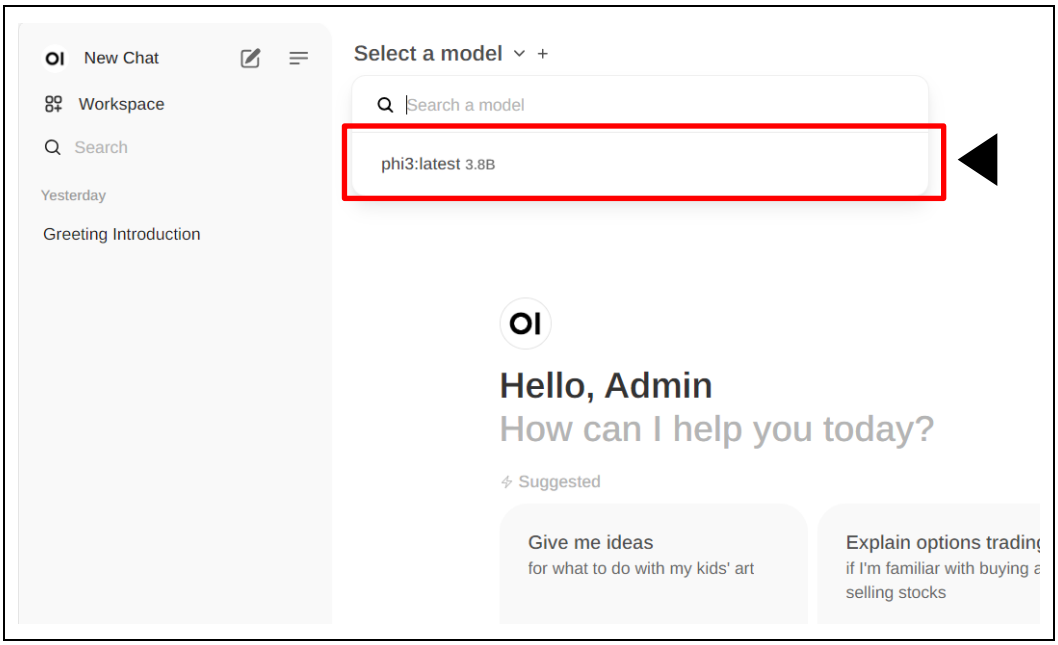


Figure 30: Inferencing with Open WebUI and Ollama

§

Revision History

Date	Revision	Description
June 2024	1.0	Initial release.

§