This comprehensive guide aims to provide university students, particularly those who are non-technical, with the knowledge and tools necessary to set up and utilize various AI platforms to enhance their writing and overall writing skills. By integrating advanced AI tools like Ollama, OpenWebUI, and Stable Diffusion, students can streamline their writing process, generate high-quality content, and gain insights into effective writing techniques.

Goals of This Guide

1. **Introduction to AI Tools:**
    - Familiarise students with AI tools that can assist in writing and content creation.
    - Explain the benefits of using AI for writing, such as idea generation, language refinement, and style improvement.
2. **Setting Up Windows Subsystem for Linux (WSL) with Ubuntu:**
    - Provide step-by-step instructions for installing and configuring WSL on a Windows computer.
    - Guide students through the process of setting up an Ubuntu environment within WSL to run Linux-based AI tools.
3. **Installing and Running Ollama on Ubuntu:**
    - Detailed instructions on downloading, installing, and running Ollama, an AI language model platform, on Ubuntu.
    - Highlight various AI models available within Ollama and their applications for writing and coding.
4. **Integrating Ollama into Obsidian:**
    - Walkthrough of integrating Ollama with Obsidian, a popular note-taking and knowledge management application.
    - Explain how to configure the BMO ChatBot plugin in Obsidian to leverage Ollama for writing assistance.
5. **Installing OpenWebUI on Ubuntu:**
    - Step-by-step guide to installing Docker and running the OpenWebUI Docker container.
    - Instructions on accessing OpenWebUI through a web interface for interactive AI model communication.
6. **Configuring Firewall Settings for Remote Access:**
    - Instructions on adjusting firewall settings to allow remote access to AI services hosted on Ubuntu.
    - Enable students to access their AI tools from any device on their local network.
7. **Installing Stable Diffusion for Image Generation:**
    - Guide on setting up Stable Diffusion with Automatic1111's web interface for AI-driven image generation.
    - Integration of Stable Diffusion with OpenWebUI to enhance functionality.

Benefits for Students

- **Enhanced Writing Skills:**
    - AI models can provide instant feedback on writing style, grammar, and coherence.
    - Generate ideas and expand vocabulary, aiding in the creative writing process.
- **Efficient Content Creation:**
    - AI tools can quickly generate drafts, summaries, and comprehensive content on various topics.
    - Save time on research and drafting, allowing students to focus on refining their work.
- **Improved Technical Writing:**

  - Models like Nous-Hermes2 excel in scientific discussions and coding tasks, aiding technical writing and problem-solving.
  - Access to multilingual support for writing in different languages.
- **Interactive Learning Experience:**
  - Integration with Obsidian allows students to interact with AI directly within their note-taking application.
  - Real-time assistance and interactive chats improve understanding and retention of writing techniques.

About AI Tools

Ollama

Ollama is a free and open-source app, developed by Michael and Jeffrey Morgan, that lets anyone run open LLMs (Large Language Models) locally on your system. It is designed to be user-friendly and accessible to non-technical users. Ollama can assist with a wide range of tasks, including writing, coding, and general language understanding. Originally with a command-line interface, however we will integrate it into Obsidian for a more user-friendly experience.

OpenWebUI

OpenWebUI is a free and open-source web interface for interacting with AI models. It provides a user-friendly way to communicate with AI models through a web browser. OpenWebUI simplifies the process of running AI models and allows users to generate text, images, and other content without needing to interact with the command line. The lead developer of OpenWebUI is Timothy Jaeryang Baek. The interface is similar to ChatGPT, so the user can easily interact with the AI.

BMO ChatBot

Is a free and open-source plugin for Obsidian that allows users to interact with AI models directly within the Obsidian. It provides a chat interface where users can ask questions, generate text, and receive responses from AI models. BMO ChatBot supports various AI models and can be customized to suit different writing tasks and preferences. It has been developed by Long Huynh.

Setting Up Windows Subsystem for Linux (WSL) with Ubuntu on Windows

What is WSL?

Windows Subsystem for Linux (WSL) allows you to run a full-fledged Linux system directly on your Windows computer without needing a virtual machine. By default, it installs Ubuntu, but other Linux distributions are available as well. This guide will walk you through the steps to install and set up WSL with Ubuntu.

Step-by-Step Guide to Install WSL with Ubuntu

1. **Open the Terminal with Administrator Privileges:**

   1. Press the `Win` key to open the `Search Bar`.
   2. Type `Terminal`.

   3. Right-click on the `Terminal` app and select `Run as administrator`.
   4. Click `Yes` on the prompt that appears.
   5. Your Terminal should now display something like:

   ```
   PS C:\Users\YourUsername>
   ```

2. **Install WSL:**

   1. In the Terminal, type the following command and press `Enter`:

   ```
   wsl --install
   ```

   This command installs WSL along with the default Ubuntu distribution.

3. If you want to see other available Linux distributions, type:

   ```
   wsl --list --online
   ```

   To install a different distribution, use:

   ```
   wsl --install [DISTRIBUTION]
   ```

   For this guide, we'll stick with **Ubuntu**.

4. **Restart Your Computer:**

   1. After the installation process is complete, restart your computer.
   2. The Terminal should open automatically after the restart.

5. **Set Up Your User Account:**

   1. The first time you run Ubuntu, you will be asked to create a new user account.
   2. Enter a username and password when prompted. This username and password are specific to your Ubuntu system and are different from your Windows credentials.

6. **Update and Upgrade Your System:**

   1. Once Ubuntu is set up, it's important to update and upgrade the system to ensure you have the latest software and security patches. In the Terminal, type:

      - `sudo` grants administrative privileges (similar to admin rights on Windows).
      - You might be prompted to enter your password for `sudo` privileges.

      ```
      sudo apt update && sudo apt upgrade -y && sudo apt clean
      ```

   2. This command will update the package lists, upgrade the installed packages, and clean up unnecessary update files.

7. **Launching Ubuntu Again:**

   1. To open Ubuntu anytime after the initial setup, type the following command in the Terminal:

```
wsl -d Ubuntu
```

Additional Tips

- You can switch between Linux distributions by specifying their name in the `wsl -d` command.
- If you encounter any issues, you can always refer to the WSL Documentation for more detailed information and troubleshooting steps.

Potential Troubleshooting Tips for Common Issues

1. **WSL Installation Fails:**

   - Ensure that your Windows version supports WSL 2. WSL 2 is available on Windows 10, version 1903, and higher, or Windows 11.
   - Make sure that Virtual Machine Platform and Windows Subsystem for Linux features are enabled. You can enable them via PowerShell (run as administrator):

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all
/norestart
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-
Linux /all /norestart
```

2. **WSL 2 Kernel Update is Required:**

   - Download and install the latest WSL 2 kernel update package from the Microsoft WSL 2 Kernel Update page.

3. **WSL 2 Not Set as Default:**

   - If you encounter errors related to WSL 2, ensure it is set as your default version:

```
wsl --set-default-version 2
```

4. **Unable to Access WSL After Installation:**

   - Check if your WSL installation is properly initialized by running:

```
wsl
```

   - If WSL doesn't start, restart your computer and try again.

5. **Internet Connectivity Issues in WSL:**

   - If you are unable to access the internet from within WSL, try disabling any VPN or proxy services and check your firewall settings.

6. **Slow Performance:**

   - WSL 2 is designed to be fast, but if you experience slow performance, ensure that your system meets the minimum hardware requirements and that you are not running too many resource-intensive applications concurrently.

Windows Requirements for Running WSL

- **Windows 10:**
  - Version 1903 or higher with Build 18362 or higher for WSL 2.
- **Windows 11:**
  - Any version supports WSL 2.
- **Hardware Requirements:**
  - x64 system with a minimum of 4GB of RAM. - BIOS-level hardware virtualization support must be enabled in the BIOS settings.

Setting Up Ollama on Ubuntu

What is Ollama?

Ollama is a platform that lets you run various AI language models locally. These models can assist with tasks like writing, coding, and general language understanding.

> There is an Ollama Windows Preview Version available, however it is still in development and not recommended for production use.

Step-by-Step Guide to Install Ollama on Ubuntu

1. **Install Ollama:**

   1. Open your terminal and start Ubuntu.

   2. In the terminal, type the following command and press `Enter`. This command will download and install Ollama:

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

   3. Ollama will be installed at: `/usr/local/bin/ollama`

    4. To check if Ollama is installed correctly, type:

   ```
   ls /usr/local/bin/ollama
   ```

   If the path exists, you should see the output confirming the presence of the Ollama executable.

2. **Confirm Ollama is Running:**

    1. Open your web browser and navigate to `localhost:11434`.
    2. You should see a message saying: `Ollama is running`. This confirms that Ollama is up and running.

Importance of Checking System Requirements and Hardware Compatibility

Before proceeding with the installation, ensure your system meets the necessary hardware requirements and is compatible with the AI models you intend to use. Some models are resource-intensive and require significant amounts of VRAM or RAM.

Choose a Model

- Visit Ollama Models to browse available models. Here are a few notable ones:
    - **Llama3**
        - **Made by**: Meta
        - **Good at**: General language understanding, versatile tasks, robust performance in both conversational and instructive contexts.
        - **Lacks**: May not be highly specialized in niche domains compared to more focused models.
    - **Gemma2**
        - **Made by**: Google DeepMind
        - **Good at**: Strong overall language capabilities, good for a wide range of writing and comprehension tasks.
        - **Lacks**: Potentially less specialized in highly specific domains or advanced coding tasks.
    - **Phi3**
        - **Made by**: Microsoft
        - **Good at**: Reasoning and language understanding, lightweight and efficient.
        - **Lacks**: Might lack the depth and breadth of larger, more resource-intensive models.
    - **Aya**
        - **Made by**: Cohere
        - **Good at**: Multilingual support, capable in 23 different languages, useful for global or diverse writing tasks.
        - **Lacks**: May not be as finely tuned for complex English-specific nuances as models focused solely on English.
    - **Nous-Hermes2**
        - **Made by**: Nous Research
        - **Good at**: Excels at scientific discussions and coding tasks, making it highly suitable for technical writing and complex problem-solving.
        - **Lacks**: May not perform as well in general conversational contexts compared to models specifically designed for broad, casual dialogues.

Hardware Resource Requirements

Each model has different versions and resource requirements. For instance:

You should have at least 8 GB of RAM available to run the 7B models, 16 GB to run the 13B models, and 32 GB to run the 33B models.

Install a Model in Ubuntu

1. To install a model, use the `ollama pull` command followed by the model name. For example:

```
# Install Llama3.1 (8b)
ollama pull llama3.1:8b

# Install Gemma2 (9b)
ollama pull gemma2:9b

# Install Phi3 (14b)
ollama pull phi3:14b

# Install Aya (8b)
ollama pull aya:8b

# Install Nous-Hermes2 (10.7b)
ollama pull nous-hermes2:10.7b
```

Run a Model

1. To use a model, run the following command:

```
# Run Phi3 (14b)
ollama run phi3

# Run Llama3
ollama run llama3.1
```

2. Type your input and get responses directly in the terminal interface.

Exit Ollama's AI

To stop using the AI model, type:

```
/bye
```

Exit Terminal

To close the terminal, simply type:

```
exit
```

Remove a Model

```
ollama rm llama3.1
```

Potential Issues and Troubleshooting Tips

1. **Installation Errors:**

    - If the installation script fails, ensure you have curl installed:

    ```
    sudo apt install curl
    ```

    - Check your internet connection and try running the script again.

2. **Model Download Failures:**

    - Ensure you have sufficient disk space for the model files.
    - Check your internet connection and try downloading the model again.

3. **Performance Issues:**

    - Verify that your system meets the hardware requirements for the selected model.

4. **Ollama Not Running:**

    - If you cannot confirm that Ollama is running on `localhost:11434`, restart the service:

    ```
    sudo systemctl restart ollama
    ```

    - Check for any error messages in the terminal that might indicate what went wrong.

GPU Usage

1. Open a new terminal window and type:

    ```
    wsl -d Ubuntu

    watch -n 0.5 nvidia-smi
    ```

Integrating Ollama into Obsidian

BMO ChatBot allows you to communicate with different AI models directly from within Obsidian. You can ask questions, seek advice, or generate text based on your prompts. This guide will walk you through the steps to integrate Ollama into Obsidian.

I have to mention other notable plugins that can integrate Ollama into Obsidian. Their setup is similar to BMO Chatbot.

- Smart Second Brain
  - indexes all of your existing Obsidian Notes and uses them as it's knowledge base.
  - you can ask questions about the content of your Notes
  - when I tried it with roughly 900 notes, indexing took about 30 minutes
  - I tested it with `phi3:14b` and the recommended embedding model
- Local GPT
  - can summarize and generate text
  - fix spelling and grammar in text
  - interpret contents of an image

Obsidian Installation

1. **Download and Install Obsidian:**
   - Visit the Obsidian website and download the app for your operating system.
   - Install Obsidian by following the on-screen instructions.

BMO Chatbot Installation

Through Community Plugins

1. **Open Obsidian:**
2. **Open Settings:**
   1. Click on the gear icon in the bottom left corner to open the Settings.
3. **Install the BMO ChatBot Plugin:**
   1. In the Settings menu, go to the "Community Plugins" tab.
   2. Click on "Browse" to search for plugins. ![[Pasted image 20240708125347.png]]
   3. In the search bar, type `BMO Chatbot` and press `Enter`.
   4. Click on `Install` to install the plugin.
   5. After installation, click on `Enable` to activate the plugin. ![[Pasted image 20240708125226.png]]

Through Github

> After finishing the documentation and following my own instructions, I had some problems installing BMO Chatbot through the Community Plugins. I can't be sure if this problem will remain or even happen for everyone. Therefor I offer an alternative way of installing BMO Chatbot.

1. Open your Ubuntu instance

2. Navigate to your Obsidian Vault (where your Obsidian Notes are stored)

   1. `cd /path/to/your/obsidian/vault/.obsidian/plugins`

    2. for me it is: `cd /mnt/c/Users/twiisker/Documents/DataScience/DS_Notes/.obsidian/plugins`

3. Download the BMO GitHub Repository

```
wget https://github.com/longy2k/obsidian-bmo-
chatbot/archive/refs/heads/main.zip -O obsidian-bmo-chatbot.zip
```

4. Extract the downloaded ZIP File

```
unzip obsidian-bmo-chatbot.zip
```

5. Rename the extracted folder

```
mv obsidian-bmo-chatbot-main obsidian-bmo-chatbot`
```

6. Navigate into the folder

```
cd obsidian-bmo-chatbot
```

7. In the BMO Chatbot folder run the following:

```
npm install
npm run build
```

8. Open Obsidian and navigate to **Settings** --> **Community Plugins**
    1. Enable **BMO Chatbot**

BMO Chatbot Configuration

1. **Configure BMO ChatBot:**
    1. Go back to the Settings menu and find the BMO ChatBot plugin in the list of installed plugins.
    2. Click on it to open its settings. ![[Pasted image 20240708125436.png]]
    3. Create an Ollama connection by entering the following URL in the "Ollama REST API URL" field: http://localhost:11434
2. **Select Your Model:**
    1. Under the "General" section in the BMO ChatBot settings, find the "Model" option.
    2. Click the reload arrow to refresh the list of available models.
    3. Choose your desired model from the list, such as `llama3` or `phi3:14b`.
3. **Open BMO ChatBot:**

1. On the left navigation bar in Obsidian, you will see a button with a robot face.
2. Click it to open the BMO ChatBot on the right side of the Obsidian interface.

Open BMO	BMO Chatbot
![[Pasted image 20240708125913.png]]	![[Pasted image 20240708130047.png]]

Using BMO ChatBot in Obsidian

Examples of Daily Use Cases

1. **Academic Writing:**
   - **Thesis Writing:** Use the chatbot to generate ideas, refine thesis statements, or draft sections of your thesis.
   - **Research Papers:** Ask for summaries of complex topics, generate citations, or get assistance with structuring your papers.
2. **Creative Writing:**
   - **Story Development:** Generate character descriptions, plot ideas, or dialogue suggestions.
   - **Poetry:** Use the chatbot to create poetic verses, brainstorm themes, or refine your poetry.
3. **Technical Writing:**
   - **Code Documentation:** Generate explanations for code snippets, document APIs, or write technical guides.
   - **Reports:** Get help with drafting technical reports, structuring content, and ensuring clarity.
4. **Daily Planning:**
   - **Task Lists:** Create daily to-do lists, prioritize tasks.
   - **Meeting Notes:** Summarize meeting discussions, create action items, and draft follow-up emails.

Features

1. **Reference Current Note:**
   - You can configure BMO ChatBot to reference the current note you are working on, making it easier to integrate AI responses into your existing notes.
2. **Custom Profiles and Models:**
   - You can switch between different AI models and profiles to suit various tasks and preferences, such as general conversations, technical assistance, or creative writing.
3. **Manage Prompts:**
   - You can save, list, and clear prompts to streamline the chat process and ensure that you are using the most effective prompts for your needs.
4. **Save and Append Chat History:**
   - BMO ChatBot allows you to append chat history to your notes or save it for future reference, helping you keep track of important conversations and ideas.
5. **Control Response Generation:**
   - You can set parameters like maximum tokens and temperature to control the length and creativity of the responses you receive from the chatbot.

Commands

- Use the `/h` Help command to list all commands.

General Commands

Command	Description
/clear or /c	Clears the chat history, removing all previous interactions from the chat window.
/ref on	Enables the "reference current note" feature, allowing the chatbot to reference the note you're currently working on.
/ref off	Disables the "reference current note" feature, stopping the chatbot from referencing the note you're currently working on.
/maxtokens [VALUE]	Sets the maximum number of tokens (words and punctuation marks) that the chatbot can use in its responses. A higher value allows for longer, more detailed responses.
/temp [VALUE]	Adjusts the temperature setting, which controls the randomness of the chatbot's responses. A lower value (closer to 0) makes the responses more predictable and focused, while a higher value (up to 2) makes them more creative.

Profile Commands

Command	Description
/profile	Lists all available profiles, showing the different sets of configurations and preferences you can use.
/profile [PROFILE-NAME]	Switches to a different profile specified by its name, changing the chatbot's behavior and settings to match that profile.
/profile [VALUE]	Changes the current profile based on a provided value, which can be the name or identifier of the profile.

Model Commands

Command	Description
/model	Lists all available models, showing the different AI models you can switch to.
/model [MODEL-NAME]	Switches to a different model specified by its name, allowing you to use a different AI model for responses.
/model [VALUE]	Changes the current model based on a provided value, which can be the name or identifier of the model.

Prompt Commands

Command	Description
/prompt	Lists all available prompts, showing the different starting points or contexts you can use.

Command	Description
/prompt [PROMPT-NAME]	Changes the current prompt to the one specified by its name, altering the context or style of responses.
/prompt [VALUE]	Sets a new prompt based on a provided value, changing the context or instructions given to the AI.
/prompt clear	Clears the current prompt, removing any specific context or instructions.

Editor Commands

Command	Description
/append	Appends the current chat history to a note, saving the conversation in your Obsidian notes.
/save	Saves the current chat history to a note, preserving the conversation for future reference.

Response Commands

Command	Description
/stop or /s	Stops fetching responses from the AI. **Warning**: Some models, like Anthropic models, cannot be aborted once started. Use with caution.

Best Practices for Using BMO ChatBot in Obsidian

1. **Regular Updates:**
    - Regularly update your BMO ChatBot plugin to access the latest features and improvements. Keeping your tools up-to-date ensures you benefit from the latest enhancements and security updates.
2. **Experiment with Different Models:**
    - Experiment with different AI models to find the one that best fits your writing needs. Different models may offer varying strengths in areas like creative writing, technical documentation, or general conversation.
3. **Save and Append Commands:**
    - Use the `append` and `save` commands to keep track of important information from your chats. This practice helps you organize your notes and ensures that valuable insights are not lost.
4. **Adjusting Settings:**
    - Adjust the `maxtokens` and `temp` settings to control the length and creativity of the AI's responses. Tweaking these parameters can help you get more relevant and useful outputs for your specific needs.
5. **Using Prompts:**
    - Utilize saved prompts to streamline your chat process. Well-crafted prompts can significantly enhance the quality of responses you receive from the AI.
6. **Referencing Notes:**
    - Enable the "reference current note" feature to allow the chatbot to provide context-aware assistance. This feature is particularly useful when working on complex documents or projects that require consistent context.

Autostart Ollama and your model

Since all of this only works when your Ubuntu instance is running and therefor Ollama and your model, you always have to open WSL and start Ubuntu.

We can auto start WSL so it opens up everytime you start your machine.

1. Search your startup folder:
   - `Win` + r
   - type: `shell:startup` and `Enter`
   - an explorer window will open with your startup folder
   - copy the path
2. Create a new file with the text editor of your choice (Notepad, VSCode, Editor, etc.)
3. write the following `batch` script:

   ```
   @echo off
   wsl
   ```

4. save it with the `.bat` extension as a Batch File
5. save it in your startup folder
   1. for example here is my path: `C:\Users\YOUR_USER_NAME\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`
6. Now everytime you start your machine a Commandline (CMD) Window will open and start your Ubuntu instance
   1. with it the Ollama Server will start and you can use your models

Setting Up OpenWebUI

OpenWebUI allows you to interact with AI models through a web interface. This guide will walk you through the steps to install and set up OpenWebUI on your Ubuntu system using Docker.

Step-by-Step Guide to Install OpenWebUI

Prerequisites

- Ensure you have an Ubuntu system with internet access.

Brief Explanation of Docker

Docker is a platform that enables developers to automate the deployment of applications inside lightweight, portable containers. Containers bundle the application and its dependencies together, ensuring that it works seamlessly in any environment. This is particularly useful for running applications like OpenWebUI that have specific dependency requirements.

1. **Install Docker:**

    1. **Add Docker's official GPG key and repository:**

        - Open your terminal and update the package list:

        ```
        sudo apt-get update
        ```

        - Install certificates for secure web communications:

        ```
        sudo apt-get install ca-certificates curl
        ```

        - Create a directory to store GPG keys required by APT, with the proper permissions:

        ```
        sudo install -m 0755 -d /etc/apt/keyrings
        ```

        - Download the official Docker GPG key and save it securely in the created directory:

        ```
        sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
        /etc/apt/keyrings/docker.asc
        sudo chmod a+r /etc/apt/keyrings/docker.asc
        ```

        - Add Docker's official repository for the matching version and architecture of your Ubuntu system:

        ```
        echo \
        "deb [arch=$(dpkg --print-architecture) signed-
        by=/etc/apt/keyrings/docker.asc]
        https://download.docker.com/linux/ubuntu \
        $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
        sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
        ```

        - Update the package list again to include Docker's repository details:

        ```
        sudo apt-get update
        ```

        - Install the latest version of Docker engine along with its necessary components:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
compose-plugin
```

2. **Run OpenWebUI Docker Container:** Now that Docker is installed, you can run the OpenWebUI Docker container:

   - Start the Docker container for OpenWebUI, providing Web UI access on your local machine:

```
sudo docker run -d --network=host -v open-webui:/app/backend/data -e
OLLAMA_BASE_URL=http://127.0.0.1:11434 --name open-webui --restart always
ghcr.io/open-webui/open-webui:main
```

3. **Check Docker Container Status:** To ensure that the Docker container is running correctly, check its status:

   - Show running Docker containers:

```
sudo docker ps
```

   - You should see the OpenWebUI container listed. The output should look similar to this:

```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
abc123xyz456    ghcr.io/open-webui/open-webui:main   "/bin/sh -c 'python3…"
30 seconds ago    Up 29 seconds             open-webui
```

4. **Access OpenWebUI:**

   - Open your web browser and navigate to: http://127.0.0.1:8080
   - Sign up to create an account. The first account created will have admin privileges.
   - **Note:** Credentials are only stored locally.

Using OpenWebUI

1. **Adding Documents:**
   - Click on "Documents" in the web interface.
   - Add your own documents.
   - Reference them in any chat using `#` and selecting the correct document.

Tips and Best Practices

- **Ensure Docker is Running:** Make sure Docker is running before starting the OpenWebUI container.
- **Regular Updates:** Regularly check for updates to Docker and OpenWebUI for the latest features and security improvements.
- **Local Storage:** Remember that all credentials and data are stored locally on your machine.

- **Resource Usage:** Docker containers can consume significant system resources, including CPU, memory, and storage. Monitor your system's performance and ensure you have sufficient resources to run the containers smoothly.

Documents

1. Click on Documents.
2. Add your own documents.
3. Reference them in any chat using `#` and selecting the correct documents.

Stable Diffusion Image Generation

This guide will help you install and set up Stable Diffusion with Automatic1111's web interface on your Ubuntu system. It is designed for non-technical university students.

Purpose of Stable Diffusion

Stable Diffusion is an AI-driven image generation tool that creates high-quality images from textual descriptions. It is particularly useful for enhancing writing and content creation by providing visual aids, illustrations, and creative visuals that complement textual content. Whether you're working on academic papers, creative writing, or multimedia projects, Stable Diffusion can significantly enhance your work by generating custom images tailored to your needs.

Prerequisites

Before you start, you need to install some necessary packages. Open your terminal and run the following command:

```
sudo apt install wget git python3 python3-venv libgl1 libglib2.0-0
```

Step-by-Step Guide to Install Stable Diffusion

1. **Create a Folder for Stable Diffusion:** First, create a directory where you will install Stable Diffusion:

   - Open your terminal and type:

     ```
     mkdir stablediff

     cd stablediff/
     ```

2. **Download the WebUI Script:** Next, download the script needed to run the web interface:

   - In the terminal, type:

   ```
   wget -q https://raw.githubusercontent.com/AUTOMATIC1111/stable-diffusion-webui/master/webui.sh
   ```

3. **Make the Script Executable:** Change the permissions of the script to make it executable:

   - In the terminal, type:

   ```
   chmod +x webui.sh
   ```

4. **Run the WebUI Script:** Finally, run the script to start the Stable Diffusion web interface:

   - In the terminal, type:

   ```
   ./webui.sh --listen --api
   ```

   This will start the web server, and you should see the output indicating the server is running.

Integrating Stable Diffusion into OpenWebUI

Once you have Stable Diffusion up and running, you can integrate it into OpenWebUI for enhanced functionality.

1. **Open OpenWebUI:**

   - Launch OpenWebUI in your web browser by navigating to http://127.0.0.1:8080

2. **Access the Admin Panel:**

   - Click on your username in the top-right corner and then click on **Admin Panel**.

3. **Configure Image Generation:**

   - In the Admin Panel, go to **Settings**.
   - Select the **Images** tab.
   - Set the URL to `http://127.0.0.1:7860` (the default port for Stable Diffusion).
   - Click the **Arrow Button** on the right to reload the settings.
   - Activate the `Image generation experimental` option.
   - Click **Save** and ensure you see a message indicating `Server connection verified`.

4. **To use Stablediffusion:**

   - launch it in your Ubuntu terminal with

   ```
   ./webui.sh --listen --api
   ```

   - you can then write a prompt in the OpenWebUI and press the small image icon to generate an image
   - in the admin settings you can change the model and values

- in the official dashboard `http://127.0.0.1:7860` you have a lot more options to change and configure your image generation
- you can also install other models from Huggingface and use them in the dashboard

5. **Improve performance:**

   - Installation of **xformers:**
     - `sudo apt install python3-pip`
     - `pip install xformers`
   - Installation of **TCMalloc:**
     - `sudo apt -y install libtcmalloc-minimal4`

6. For more information visit: AUTOMATIC1111

How Stable Diffusion Can Benefit Writing and Content Creation

Stable Diffusion offers numerous benefits for various writing and content creation projects by providing high-quality, custom-generated images. Here are some specific use cases:

1. **Creative Writing:**
   - Generate character illustrations and scene visuals to complement stories and novels.
   - Design book covers and promotional images for self-publishing authors.
2. **Educational Materials:**
   - Develop educational content with detailed illustrations and infographics.
   - Create visual aids for presentations and teaching materials.
3. **Marketing and Social Media:**
   - Design engaging visuals for blog posts, social media updates, and marketing campaigns.
   - Create eye-catching graphics for advertisements and promotional materials.
4. **Project Reports:**
   - Illustrate project reports with relevant images and diagrams.
   - Enhance the visual appeal of presentations and documentation.

Accessing OpenWebUI from Anywhere by Changing Ubuntu Firewall Settings

To access OpenWebUI from other devices on your network, you need to adjust the firewall settings on your Ubuntu system running in WSL. Follow these steps to allow traffic through the necessary ports and enable the firewall.

Step-by-Step Guide to Adjust Firewall Settings

1. **Open Your WSL Ubuntu Terminal:**

   - Press the `Win` key and type `Terminal`.
   - Open your Ubuntu instance.

2. **Allow Traffic on Necessary Ports:** You need to open ports 11434 and 8080 to allow access to Ollama and OpenWebUI with UFW. In Ubuntu, type the following commands:

   - Allow traffic on port 11434 (Ollama):

```
sudo ufw allow 11434
```

- Allow traffic on port 8080 (OpenWebUI):

```
sudo ufw allow 8080
```

3. **Enable the Firewall:** If your firewall is not already enabled, you need to enable it to apply the changes. Type the following command:

```
sudo ufw enable
```

Why Adjusting Firewall Settings is Necessary

Adjusting firewall settings is essential for allowing external devices on your network to communicate with services running on your Ubuntu system in WSL. By default, firewalls block incoming traffic to protect your system from unauthorized access. Opening specific ports ensures that only the required services are accessible from other devices, enabling functionalities like remote access to AI models.

Potential Security Risks

Opening ports on your firewall can expose your system to potential security risks, such as unauthorized access or attacks from other devices on the same network. It is crucial to ensure that only trusted devices have access to these ports and to monitor for any suspicious activity. Using strong passwords and enabling other security measures, such as network encryption and regular system updates, can help mitigate these risks.

Accessing OpenWebUI from Another Device

Now, you can access your AI services from another device on your network using the IP address of your laptop (running WSL).

1. **Find Your Laptop's IP Address:**

   - On your laptop, open a Windows terminal and type:

```
ipconfig | findstr IPv4
```

   - Look for the IP address associated with your network connection (it usually starts with `192.168`).

2. **Access the Services:**

   - From another device on your network, open a web browser and type the following addresses, replacing `<your-laptop-ip>` with the actual IP address of your laptop:
   - For Ollama: `http://<your-laptop-ip>:11434`

  - For OpenWebUI: `http://<your-laptop-ip>:8080`

Example

If your laptop's IP address is `192.168.1.100`, you would access the services like this:

- Ollama: `http://192.168.1.100:11434`
- OpenWebUI: `http://192.168.1.100:8080`

Reverting Firewall Changes

If you need to revert the firewall changes and close the opened ports, follow these steps:

1. **Open Your WSL Ubuntu Terminal:**
    - Press the `Win` key and type `Terminal`.
    - Select the Ubuntu terminal from the search results.
2. **Revoke Port Permissions:** To close the previously opened ports, type the following commands:
    - Deny traffic on port 11434 (Ollama):

    ```
    sudo ufw deny 11434
    ```

    - Deny traffic on port 8080 (OpenWebUI):

    ```
    sudo ufw deny 8080
    ```

3. **Disable the Firewall:** If you want to disable the firewall completely, type the following command:

    ```
    sudo ufw disable
    ```

By following these steps, you can adjust the firewall settings to enable or disable access to your AI services as needed, ensuring both functionality and security for your system.

Glossary of Terms (Alphabetical Order)

AI (Artificial Intelligence)

The simulation of human intelligence processes by machines, especially computer systems. AI applications include expert systems, natural language processing, machine vision, and autonomous systems. AI can learn and adapt over time, making it a powerful tool for various tasks ranging from data analysis to creative endeavors.

API (Application Programming Interface)

A set of protocols and tools for building software applications. An API defines how software components should interact, allowing different software systems to communicate with each other.

BIOS (Basic Input/Output System)

Firmware used to perform hardware initialization during the booting process and to provide runtime services for operating systems and programs.

Firewall

A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.

GPU (Graphics Processing Unit)

A specialized electronic circuit designed to accelerate the processing of images and calculations involving visual data.

GPG (GNU Privacy Guard)

A free software implementation of the OpenPGP standard, allowing you to encrypt and sign your data and communications.

IP Address (Internet Protocol Address)

A numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

Kernel

The core part of an operating system, managing system resources and communication between hardware and software components.

Linux

An open-source operating system modelled on UNIX. It is widely used on servers, desktops, and mobile devices.

RAM (Random Access Memory)

A type of computer memory that can be accessed randomly. RAM is used by the system to store data that is being used or processed.

Repository

A central location where data is stored and managed. In software development, a repository is a storage location for software packages.

Terminal

A text-based interface used to interact with the computer's operating system. Commands are typed in by the user and the output is displayed in the terminal window.

Ubuntu

A popular Linux distribution based on Debian, known for its user-friendly interface and wide range of applications.

UFW (Uncomplicated Firewall)

A front-end for managing iptables firewall rules on Linux. It aims to provide an easy-to-use interface for people unfamiliar with firewall concepts.

URL (Uniform Resource Locator)

A reference (an address) to a resource on the Internet. URLs are used to access websites and other resources.

Virtualization

The creation of a virtual version of something, such as a server, storage device, or network resources. Virtualization allows multiple operating systems to run on a single physical machine, optimizing the use of resources.

VRAM (Video Random Access Memory)

A type of RAM used specifically for storing video data. It is used by the GPU to render graphics.