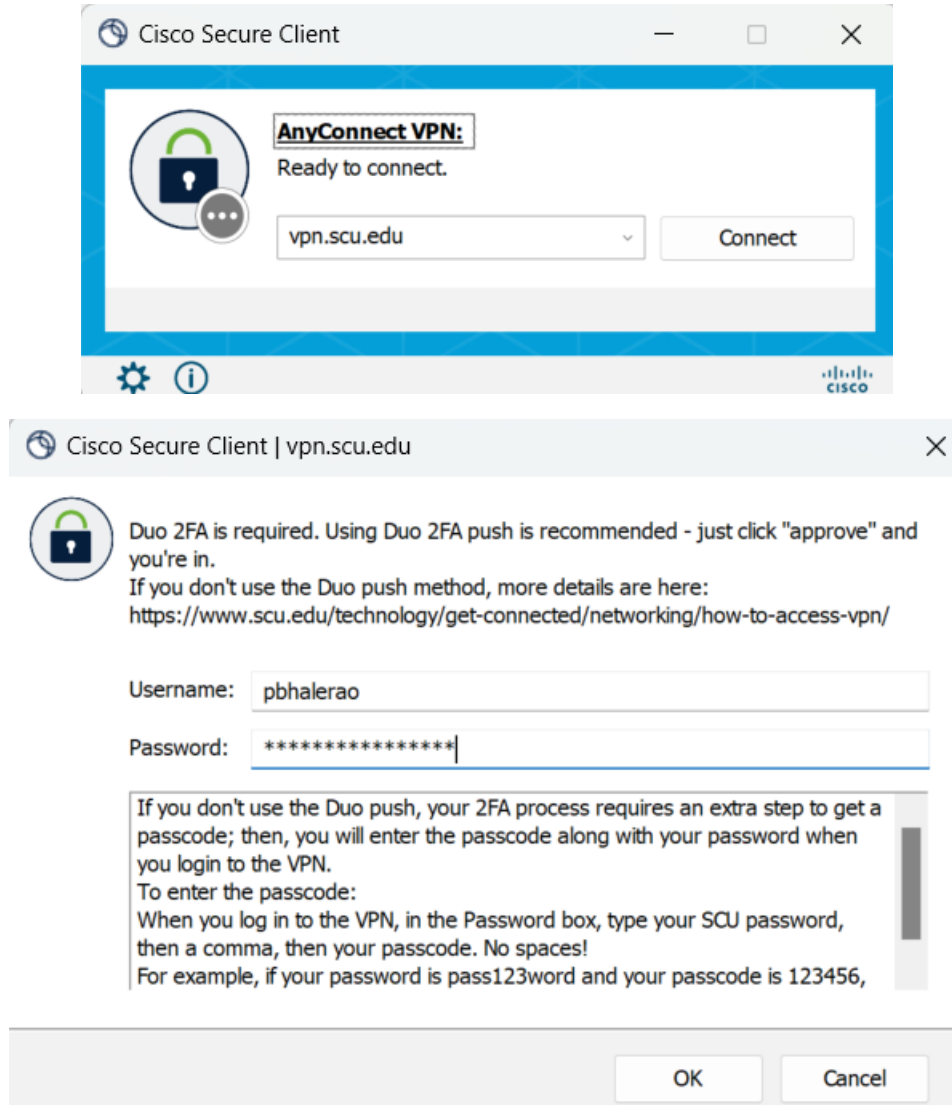


Instructions to Run the Alt-Diffusion-m18

Step-1: Open Cisco Secure Client and Connect to SCU VPN

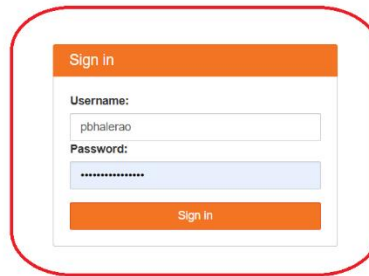
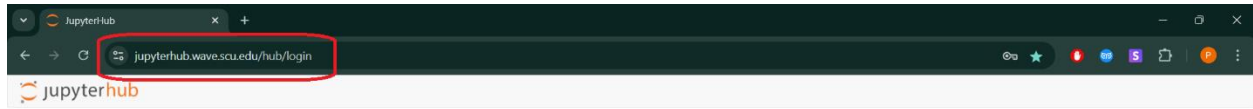


The screenshot shows the Cisco Secure Client window. The main window has a title bar "Cisco Secure Client" and a blue border. Inside, there's a "AnyConnect VPN:" section with a "Ready to connect." status. Below this, there's a dropdown menu showing "vpn.scu.edu" and a "Connect" button. At the bottom of the window, there are icons for settings and information, and the Cisco logo.

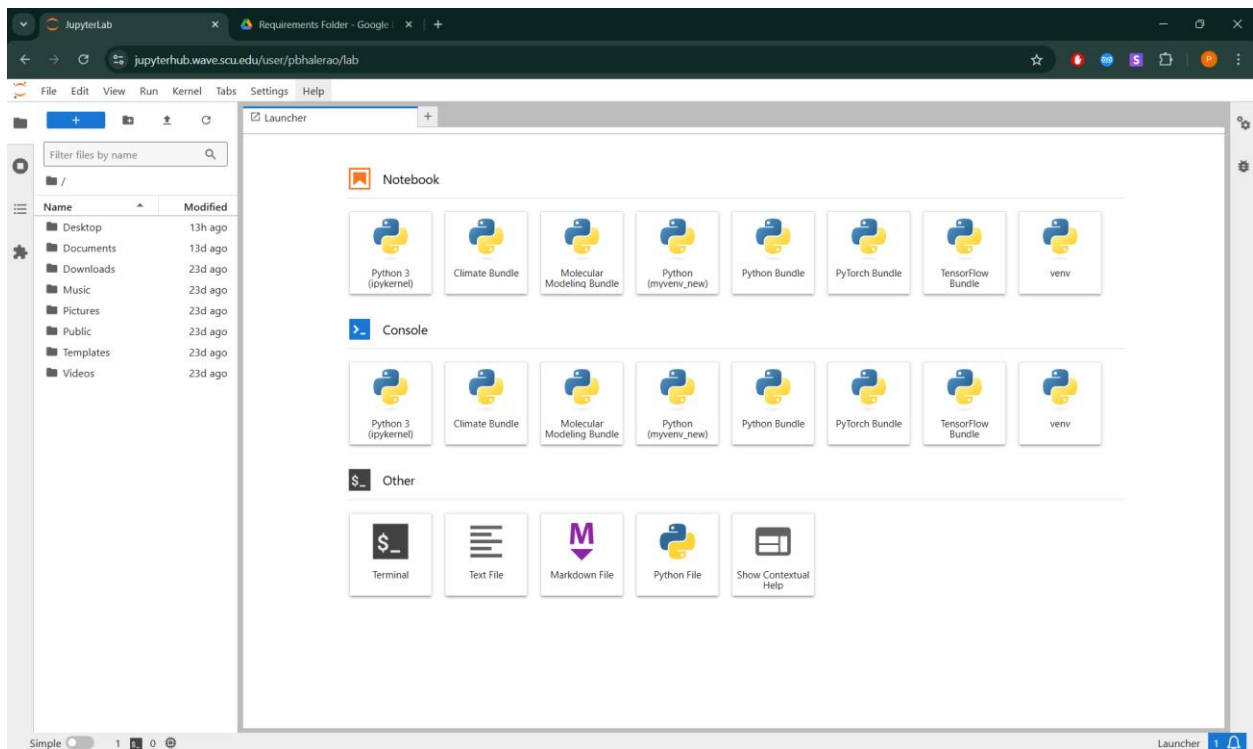
Below the main window, there's a smaller window titled "Cisco Secure Client | vpn.scu.edu". This window contains a Duo 2FA notification. It says: "Duo 2FA is required. Using Duo 2FA push is recommended - just click 'approve' and you're in. If you don't use the Duo push method, more details are here: <https://www.scu.edu/technology/get-connected/networking/how-to-access-vpn/>". Below this, there are input fields for "Username:" (containing "pbhalerao") and "Password:" (containing "*****"). A scrollable text box provides instructions: "If you don't use the Duo push, your 2FA process requires an extra step to get a passcode; then, you will enter the passcode along with your password when you login to the VPN. To enter the passcode: When you log in to the VPN, in the Password box, type your SCU password, then a comma, then your passcode. No spaces! For example, if your password is pass123word and your passcode is 123456,". At the bottom of this window are "OK" and "Cancel" buttons.

Don't Forget to attend the DUO Push. (It is required)

Step-2: Login on the Jupyter-HUB on SCU WAVE



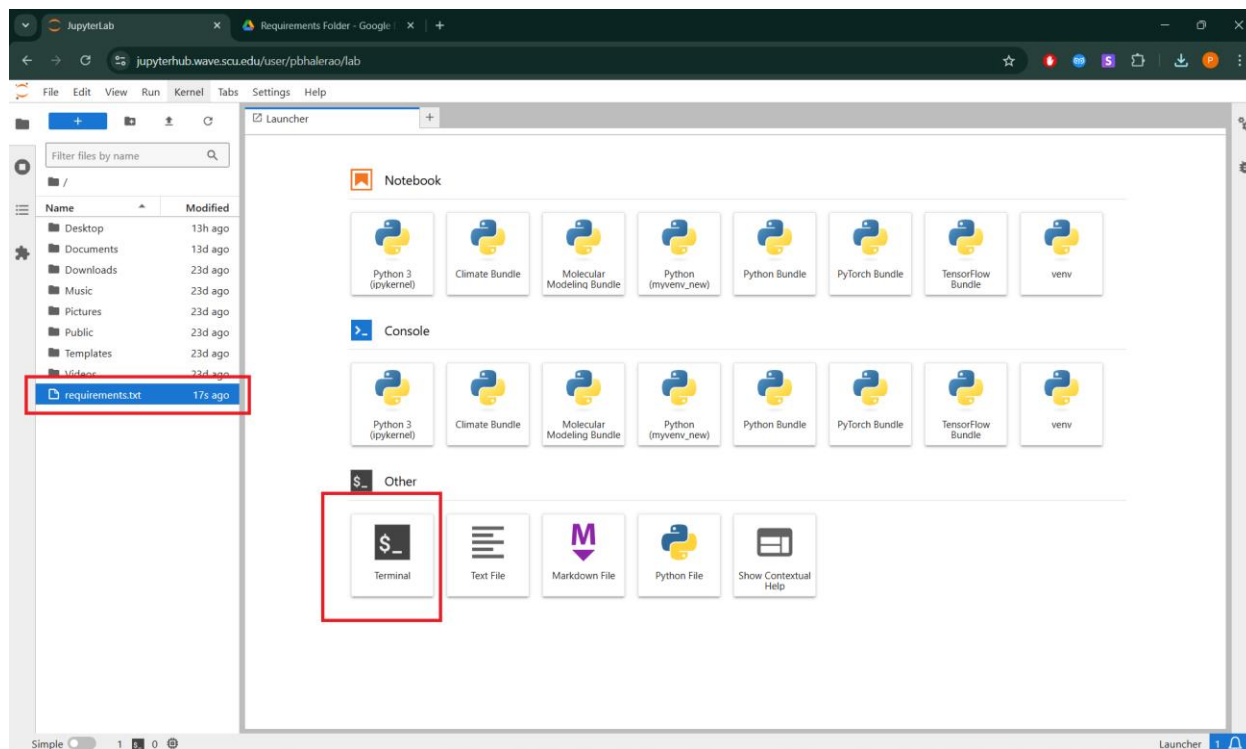
Choose any GPU System.



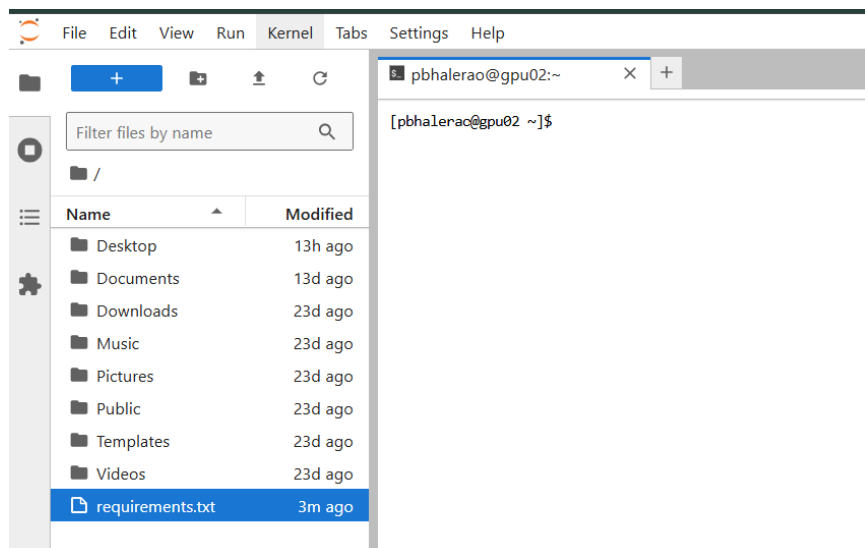
Step-3: Get Requirements.txt file and Load it to Jupyter.

Link to Requirements file: [Click Here](#)

This has the exact required library configs required by model
Just drag and drop the requirements file into the Left Panel.

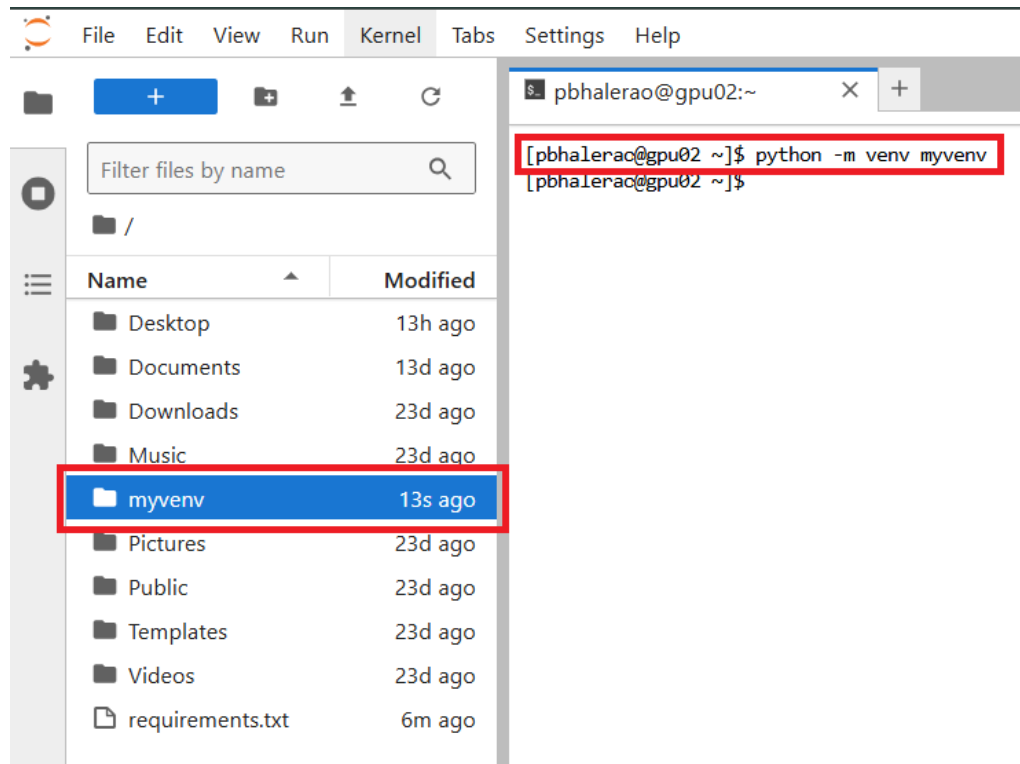


Once this is done, open the terminal.



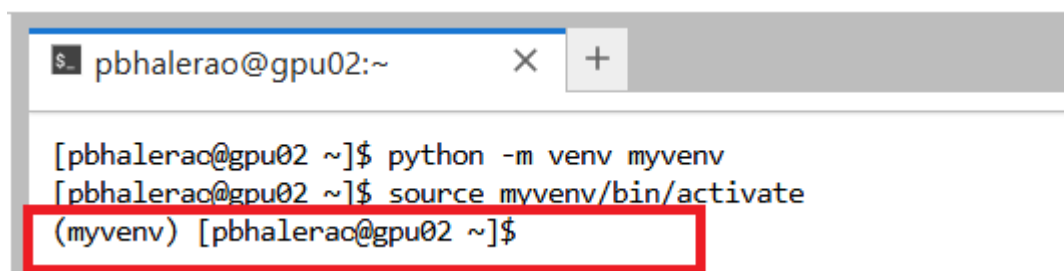
Step-4: Create a virtual Environment "myvenv"

Command: **python -m venv myvenv**



Activate virtual Environment.

Command: **source myvenv/bin/activate**



Step-4: Run Command to install PyTorch but with CUDA-12.1 Specifically.

`pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121`

(Remember this is inside your virtual environment)

```
(myvenv) [pbhalerao@gpu02 ~]$ pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121
```

```
Looking in indexes: https://download.pytorch.org/whl/cu121
```

```
Collecting torch
```

```
  Downloading https://download.pytorch.org/whl/cu121/torch-2.5.1%2Bcu121-cp39-cp39-linux_x86_64.whl (780.4 MB)
```

```
780.4/780.4 MB 8.2 MB/s eta 0:00:00
```

```
Successfully installed MarkupSafe-2.1.5 filelock-3.13.1 fsspec-2024.2.0 Jinja2-3.1.3 mpmath-1.3.0 networkx-3.2.1 numpy-1.26.3 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.0.5 nvidia-cuda-nvrtc-cu12-12.1.1.105 nvidia-cuda-runtime-cu12-12.1.1.105 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-11.4.5.1 07 nvidia-cuspars-cu12-12.1.0.106 nvidia-nccl-cu12-2.21.5 nvidia-nvjitlink-cu12-12.1.1.105 nvidia-nvtx-cu12-12.1.1.105 pillow-10.2.0 sympy-1.13.1 torch-2.5.1+cu121 torchaudio-2.5.1+cu121 torchvision-0.20.1+cu121 triton-3.1.0 typing-extensions-4.9.0
```

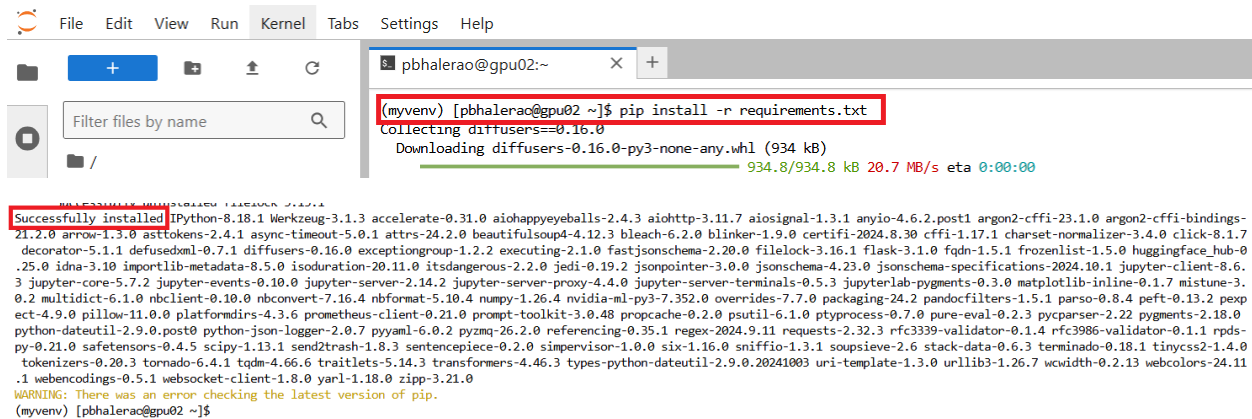
```
WARNING: There was an error checking the latest version of pip.
```

```
(myvenv) [pbhalerao@gpu02 ~]$
```

Ignore Warnings.

Step-5: Install from Requirements.txt

Command: `pip install -r requirements.txt`



```
File Edit View Run Kernel Tabs Settings Help
pbhalerao@gpu02:~
(myvenv) [pbhalerao@gpu02 ~]$ pip install -r requirements.txt
Collecting diffusers==0.16.0
  Downloading diffusers-0.16.0-py3-none-any.whl (934 kB)
934.8/934.8 kB 20.7 MB/s eta 0:00:00

Successfully installed IPython-8.18.1 Werkzeug-3.1.3 accelerate-0.31.0 aiohappyeyeballs-2.4.3 aiohttp-3.11.7 aiosignal-1.3.1 anyio-4.6.2.post1 argon2-cffi-23.1.0 argon2-cffi-bindings-21.2.0 arrow-1.3.0 asttokens-2.4.1 async-timeout-5.0.1 attrs-24.2.0 beautifulsoup4-4.12.3 bleach-6.2.0 blinker-1.9.0 certifi-2024.8.30 cffi-1.17.1 charset-normalizer-3.4.0 click-8.1.7 decorator-5.1.1 defusedxml-0.7.1 diffusers-0.16.0 exceptiongroup-1.2.2 executing-2.1.0 fastjsonschema-2.20.0 filelock-3.16.1 flask-3.1.0 fqn-1.5.1 frozenlist-1.5.0 huggingface-hub-0.25.0 idna-3.10 importlib-metadata-8.5.0 isoduration-20.11.0 itsdangerous-2.2.0 jedi-0.19.2 jsonpointer-3.0.0 jsonschema-4.23.0 jsonschema-specifications-2024.10.1 jupyter-client-8.6.3 jupyter-core-5.7.2 jupyter-events-0.10.0 jupyter-server-2.14.2 jupyter-server-proxy-4.4.0 jupyter-server-terminals-0.5.3 jupyterlab-pygments-0.3.0 matplotlib-inline-0.1.7 mistune-3.0.2 multidict-6.1.0 nbclient-0.10.0 nbconvert-7.16.4 nbformat-5.10.4 numpy-1.26.4 nvidia-ml-py3-7.352.0 overrides-7.7.0 packaging-24.2 pandocfilters-1.5.1 parso-0.8.4 peft-0.13.2 pexpt-4.9.0 pillow-11.0.0 platformdirs-4.3.6 prometheus-client-0.21.0 prompt-toolkit-3.0.48 propcache-0.2.0 psutil-6.1.0 ptyprocess-0.7.0 pure-eval-0.2.3 pycparser-2.22 pygments-2.18.0 python-dateutil-2.9.0.post0 python-json-logger-2.0.7 pyyaml-6.0.2 pyzmq-26.2.0 referencing-0.35.1 regex-2024.9.11 requests-2.32.3 rfc3339-validator-0.1.4 rfc3986-validator-0.1.1 rpds-py-0.21.0 safetensors-0.4.5 scipy-1.13.1 send2trash-1.8.3 sentencepiece-0.2.0 supervisor-1.0.0 six-1.16.0 sniffio-1.3.1 soupsieve-2.6 stack-data-0.6.3 terminado-0.18.1 tinycss2-1.4.0 tokenizers-0.20.3 tornado-6.4.1 tqdm-4.66.6 traitlets-5.14.3 transformers-4.46.3 types-python-dateutil-2.9.0.20241003 uri-template-1.3.0 urllib3-1.26.7 wcwidth-0.2.13 webcolors-24.11.1 webencodings-0.5.1 websocket-client-1.8.0 yarl-1.18.0 zipp-3.21.0

WARNING: There was an error checking the latest version of pip.
(myvenv) [pbhalerao@gpu02 ~]$
```

Ignore Warnings.

Step-6: Set path to environment.

`echo 'export PATH="/WAVE/users2/unix/pbhalerao/myvenv/bin:$PATH"' >> ~/.bashrc && source ~/.bashrc`

A terminal window with the title 'pbhalerao@gpu02:~'. The command 'echo 'export PATH="/WAVE/users2/unix/pbhalerao/myvenv/bin:\$PATH"' >> ~/.bashrc && source ~/.bashrc' is entered and executed. The prompt changes from '(myvenv) [pbhalerao@gpu02 ~]\$' to '(myvenv) [pbhalerao@gpu02 ~]\$' after the command is run.

```
(myvenv) [pbhalerao@gpu02 ~]$ echo 'export PATH="/WAVE/users2/unix/pbhalerao/myvenv/bin:$PATH"' >> ~/.bashrc && source ~/.bashrc
(myvenv) [pbhalerao@gpu02 ~]$
```

The above export path will change as per your system.

Run the above command to set path permanently to your virtual environment. This is necessary, to solve the conflict of similar libraries from the parent environment.

Once, Done, please verify the version of urllib3 and it should say **1.26.7 specifically.**

Command: pip show urllib3

```
(myvenv) [pbhalerao@gpu02 ~]$ pip show urllib3
Name: urllib3
Version: 1.26.7
Summary: HTTP library with thread-safe connection pooling, file post, and more.
Home-page: https://urllib3.readthedocs.io/
Author: Andrey Petrov
Author-email: andrey.petrov@shazow.net
License: MIT
Location: /WAVE/users2/unix/pbhalerao/myvenv/lib/python3.9/site-packages
Requires:
Required-by: requests
(myvenv) [pbhalerao@gpu02 ~]$
```

The setup is complete and this is just one time process. From Next time onwards, just activate your myvenv and run the code. So, from next time, we can jump directly to Step-7.

Reminder to activate myvenv: source myvenv/bin/activate

To come out of virtual environment: deactivate

Step-7: Run the Code Directly.

The model has been setup already and we just need to run the following python command.

Command: **python /WAVE/scratch2/oignat_lab/ParthBhaleraoWork/app.py**

The above command remains same for all users.

```
pbhalerao@gpu02:~$ python /WAVE/scratch2/oignat_lab/ParthBhaleraoWork/app.py

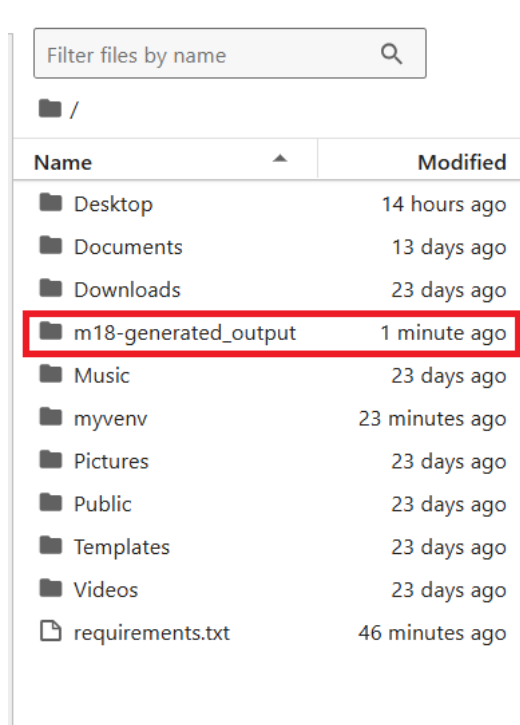
Loading the model (this will take a few minutes)...

=== Initial GPU Status ===
/WAVE/users2/unix/pbhalerao/myenv/lib/python3.9/site-packages/torch/cuda/__init__.py:716: UserWarning: Can't initialize NVML
  warnings.warn("Can't initialize NVML")
GPU: Tesla V100-PCIE-32GB
Total GPU Memory: 32494.12 MB
Allocated GPU Memory: 0.00 MB
Cached GPU Memory: 0.00 MB
Free GPU Memory: 32494.12 MB
=====

Clearing GPU memory...
GPU memory cleared!

=== GPU Status After Clearing Memory ===
GPU: Tesla V100-PCIE-32GB
Total GPU Memory: 32494.12 MB
```

The code will start running perfectly and images will be stored in the following output directory.



How to Generate Images ? and what are the next steps ?

The next important task, which we all need to figure out collectively as a team is, we need to find out best parameter settings which will generate a good quality image.

Following are the parameters for now:

- **Width:** I kept it to 768 (but we can vary from 512 to 1024)
- **Height:** I kept it to 768 (but we can vary from 512 to 1024)
- **Seed:** any value between 0 to $2^{32}-1$ (Right now 1337 is giving somewhat good results, but please try other values, we need to figure out best settings.)
- **Guidance Scale:** Any value between 5.0 to 15.0 (Read instructions printed on terminal for more information)
- **Inference Steps:** Any value between 20 to 150. (Higher value leads to better quality of images, and do not worry about the speed, we have the best hardware with us, so you can try maxxing out the settings, if required to test)
- **Prompt:** 18 Languages Supported. (Read more info on terminal, about the supported languages)
- Generated images can be downloaded from the output folder.


```
pbhalerao@gpu02:~  
3. Guidance Scale:  
• Recommended range: 5.0-15.0  
• Higher values = stronger adherence to prompt  
• Lower values = more creative freedom  
• Default: 7.5  
4. Inference Steps:  
• Range: 20-150  
• More steps = better quality but slower generation  
• Default: 50  
SUPPORTED LANGUAGES:  
-----  
English, Chinese, Japanese, Thai, Korean, Hindi, Ukrainian, Arabic,  
Turkish, Vietnamese, Polish, Dutch, Portuguese, Italian, Spanish,  
German, French, and Russian  
PROMPT GUIDELINES:  
-----  
• Maximum length: 77 tokens  
• Exceeding tokens will be truncated  
• Be specific and yet descriptive  
• Use commas to separate different aspects  
COMMANDS:  
-----  
• Type 'exit' or 'quit' at any prompt to end the program  
• Type 'guidelines' to see these guidelines again  
=====
```

```
=====
```

```
NEW IMAGE GENERATION
```

```
=====
```

```
Enter width (multiple of 8): 768  
Enter height (multiple of 8): 768  
Enter seed (integer): 1337  
Enter guidance scale (5.0-15.0): 8.5  
Enter number of inference steps (20-150): 100  
  
Enter your prompt (or 'exit' to quit):  
(Remember: supports multiple languages, max 77 tokens)  
> |
```

We need to figure out which setting of parameters gives the best quality of image. It can be the same or different for different languages as well, so happy to receive output from everyone.

Please convey your best results, so that we can integrate those model settings in the further Multi-AI part on the backend side.

Please let me know if anyone is stuck, and happy playing around with model settings !!

THANK YOU !!