How to use your GPU in Jupyter Notebook

Speed up your machine learning algorithms by running them on your graphics card in Jupyter Notebook

BY AYUSH PANDE

UPDATED MAR 21, 2024

If you're unfamiliar with it, Jupyter Notebook is a powerful IDE that lets you create scripts for data analysis, web scraping, machine learning, and tons of other use cases. For the average coder who's into data science, Jupyter Notebook serves as the perfect companion as it lets you create interactive documents for everything from jotting down notes to compiling complex codes. While this IDE can be installed on pretty much any modern laptop, you're going to have a tough time if you try to train AI algorithms on a CPU.

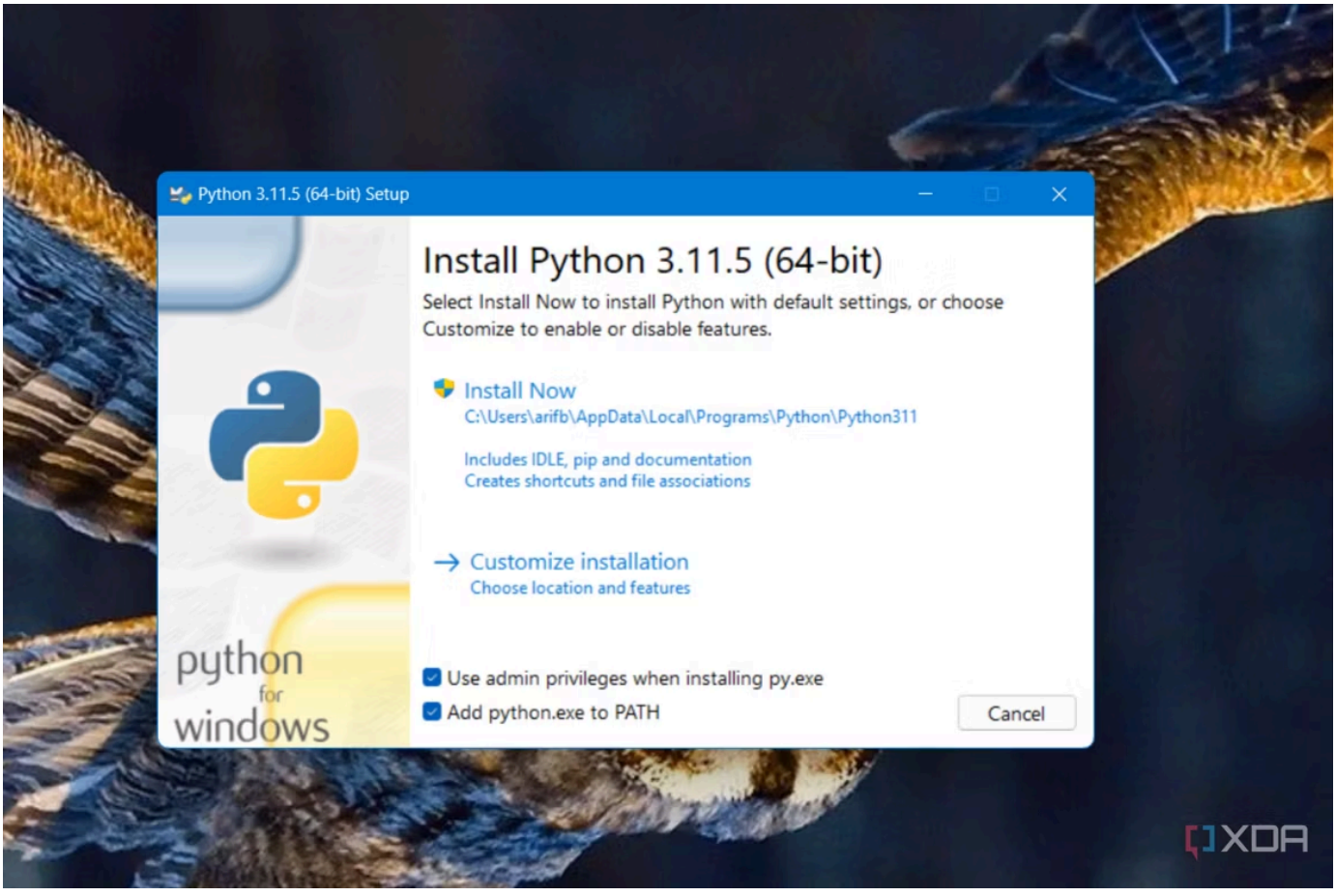
As such, you can configure Jupyter Notebook to relegate the demanding deep-learning workloads to your powerful graphics card instead of the processor. However, you'll have to go through several steps, including setting up Python libraries, creating coding environments, and installing drivers before you can run Jupyter Notebook on your graphics card.

Link copied to clipboard

RELATED

How to use Jupyter Notebook on Windows, Linux, and macOS

Installing Python



This step may sound redundant if you're already knee-deep into programming, but you'll need to install Python on your PC to use GPU-accelerated AI in Jupyter Notebook. Simply download the Python.exe file from the official website and click on the install button after granting admin privileges to the installer.

RELATED
Link copied to clipboard

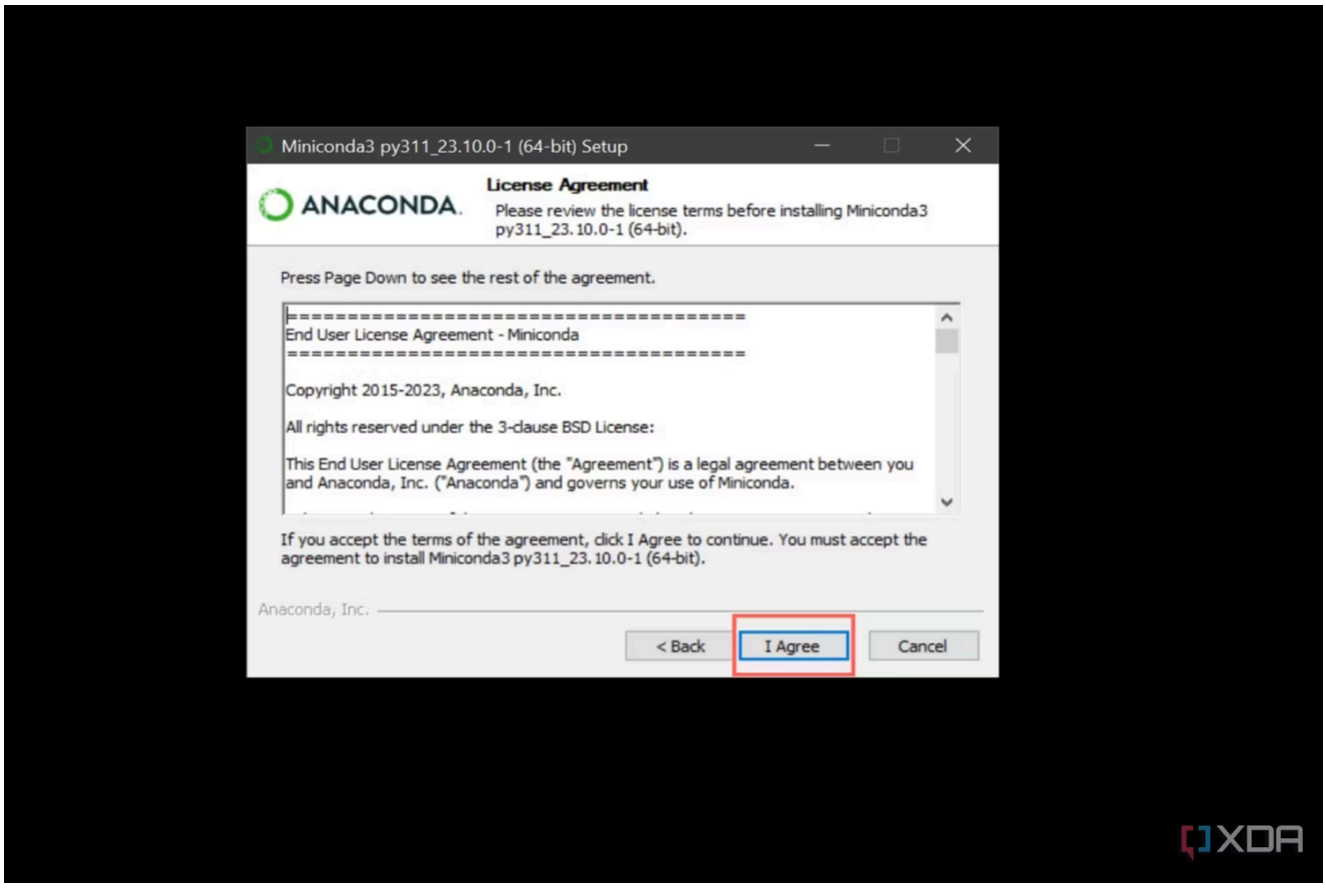
How to install Python on Windows, Linux, and macOS

For most users, I recommend choosing the Disable Path Length Limit to avoid future headaches caused by the 260-character limit on the length of file paths set by Windows 11.

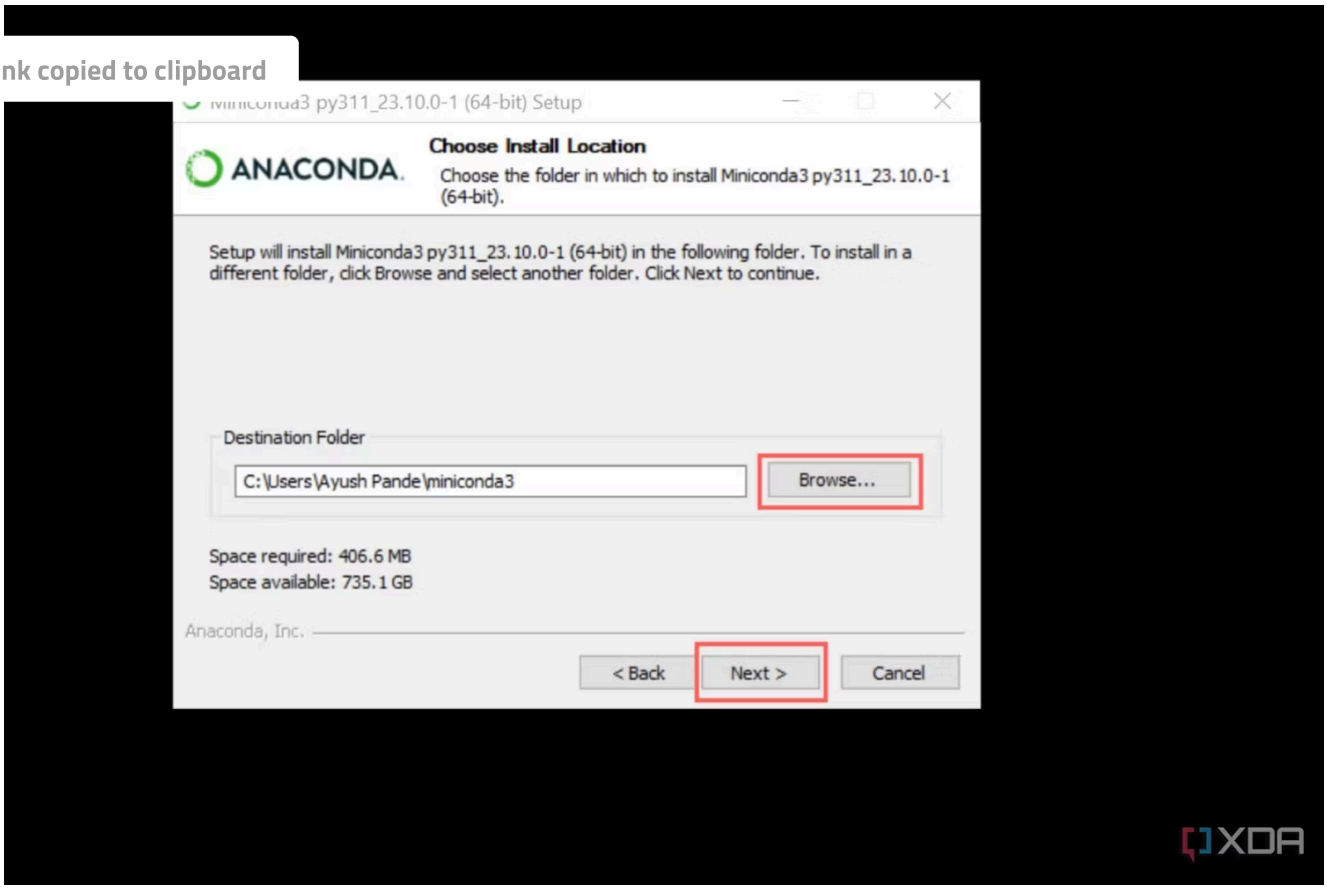
Installing Miniconda

Miniconda is a toolkit that contains important Python libraries, environments, and packages necessary to enable your GPU. It also lets you create a Jupyter Notebook.

1. Download the setup.exe file from the official website and run it with admin privileges.
2. Choose the **I Agree** option when the installer asks you to agree to the licensing terms and hit **Next**.



3. Choose the directory where you wish to install Miniconda, and click on the **Next** button.



4. Hit the **Install** button and press **Finish** once the installation is complete.

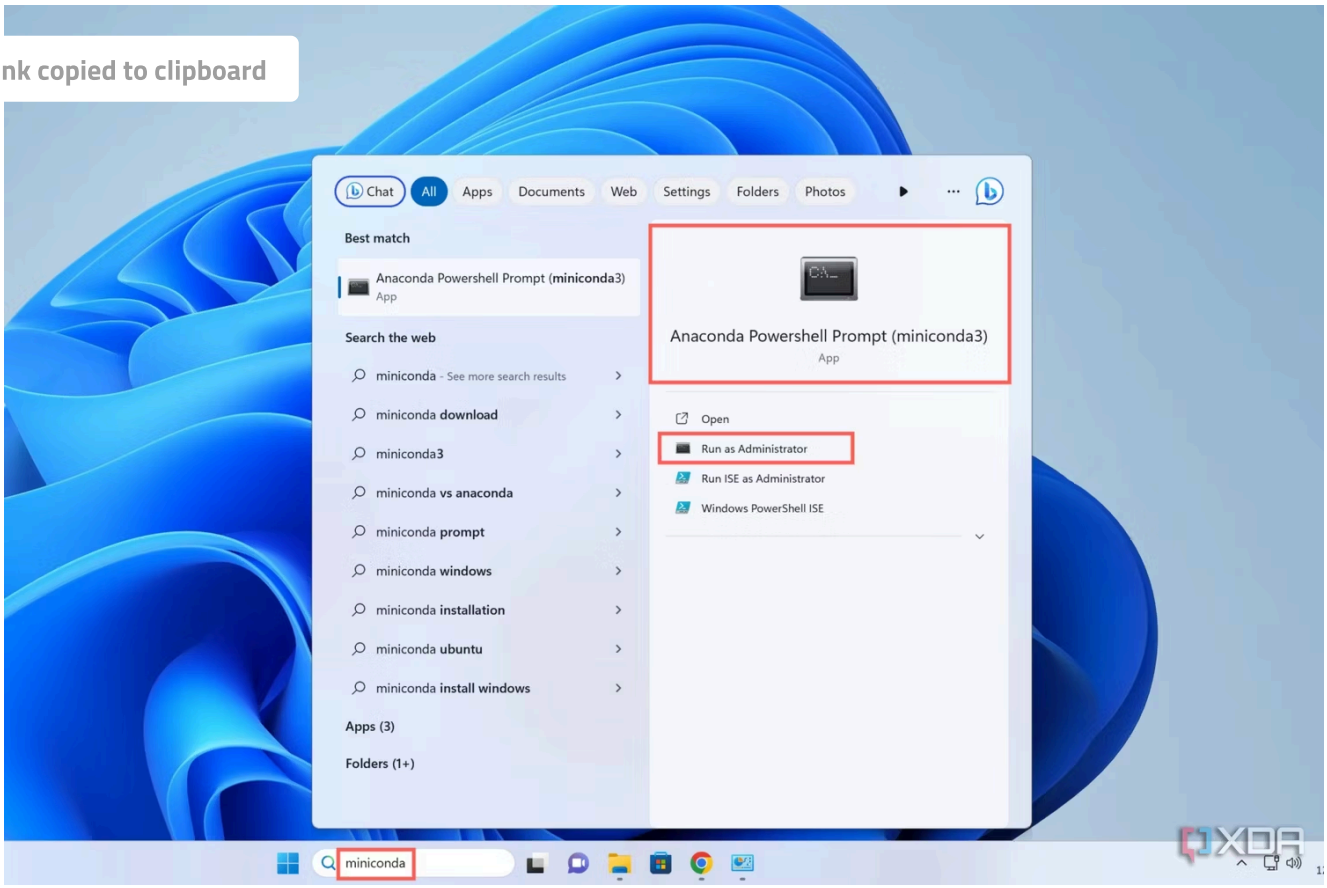
Setting up a Conda environment

Now that you've installed Python and Miniconda, it's time to configure a coding environment for your machine learning projects. I recommend creating a separate enviroment as we'll be using older packages in this tutorial.

Note

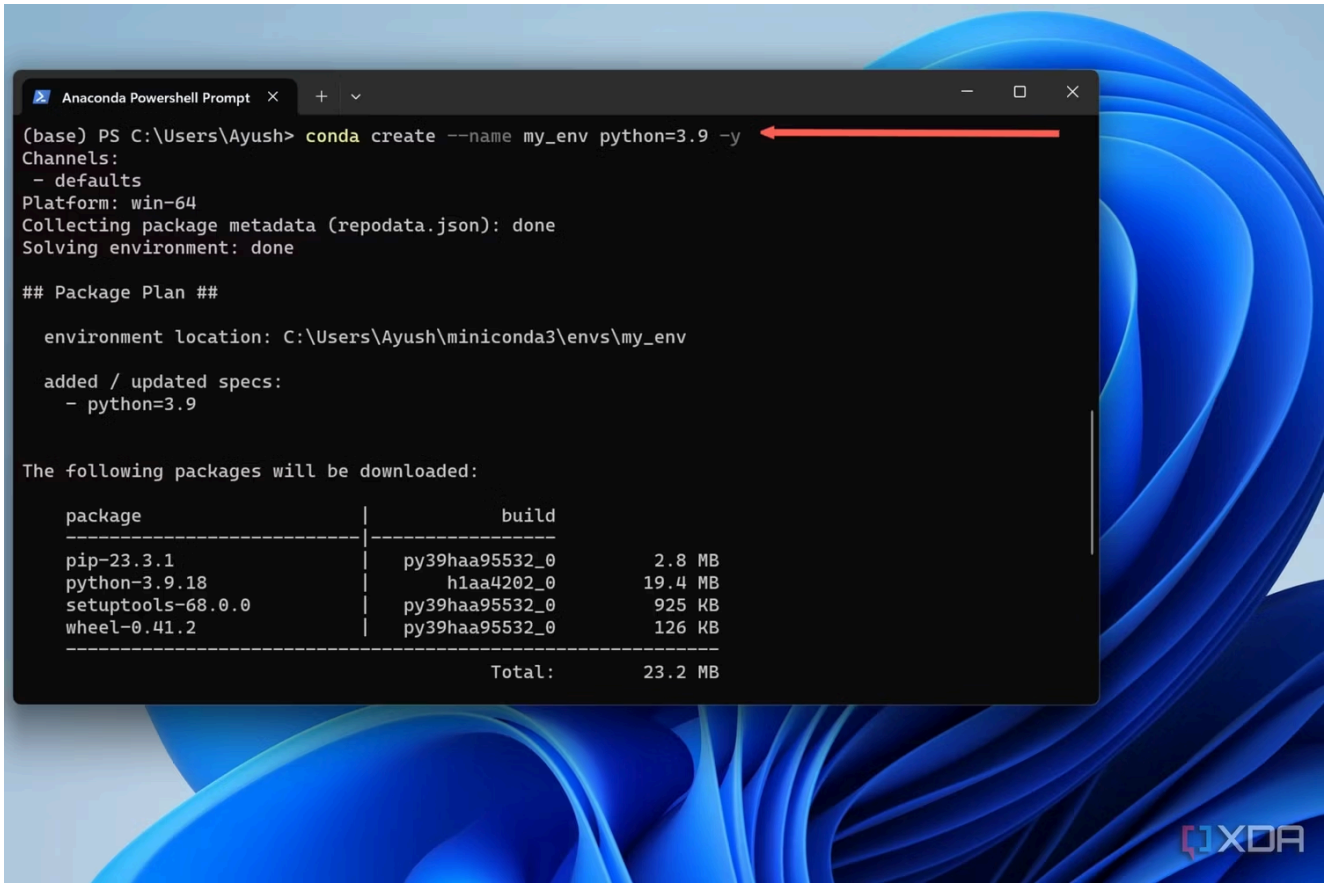
Since the latest version of TensorFlow doesn't work on Windows 11 systems that don't have WSL pre-configured, you'll have to install a build that's older than TensorFlow v2.11. The same goes for Python, so you'll have to downgrade to Python 3.9 in the new Conda environment.

1. Type **miniconda** in the **Windows Search Bar** and pick the **Run as Administrator** option under the **Anaconda Powershell Prompt**.



2. Paste the following code into the terminal and press enter:

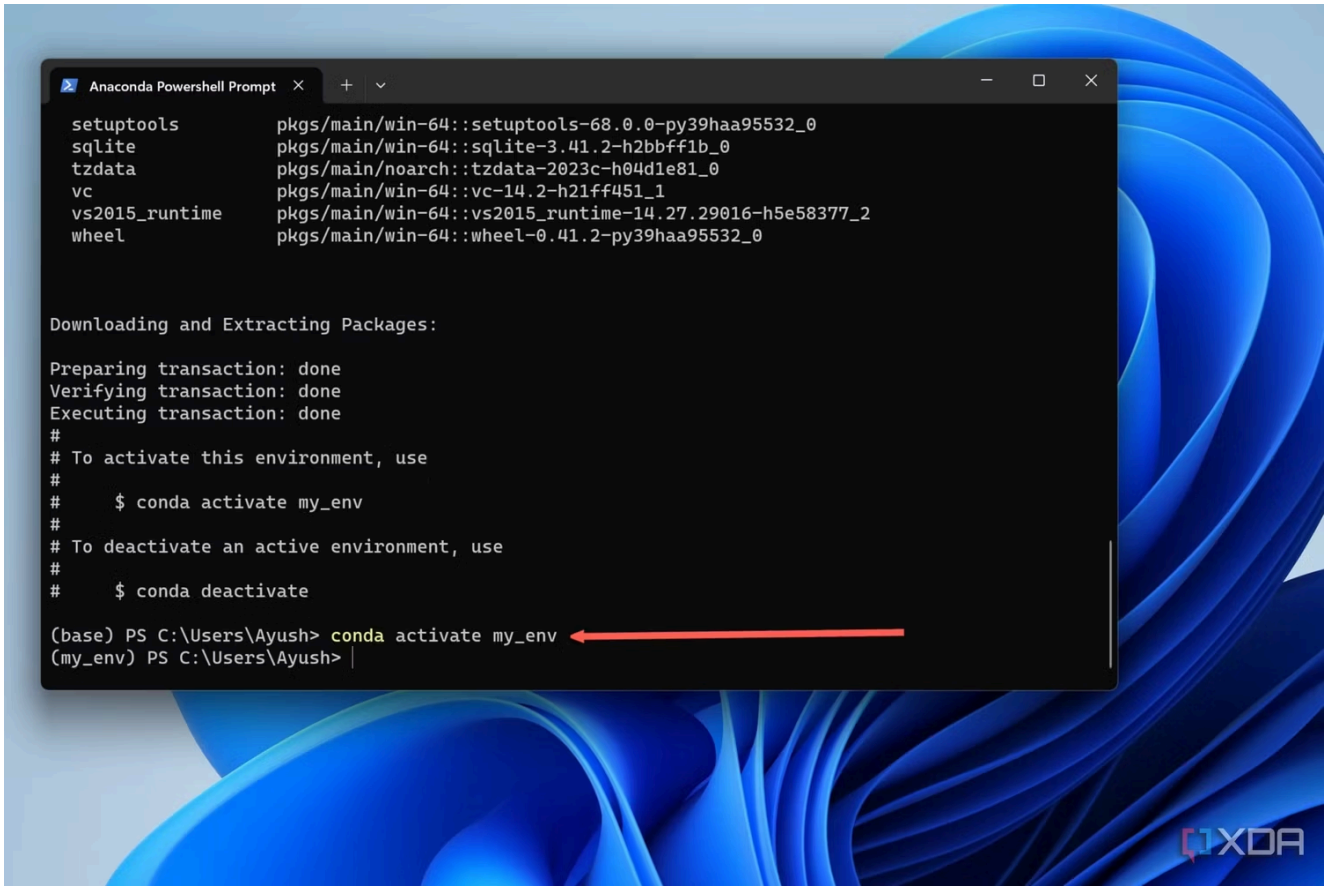
```
conda create --name my_env python=3.9 -y
```



3. Activate the newly created environment using the following command:

Link copied to clipboard

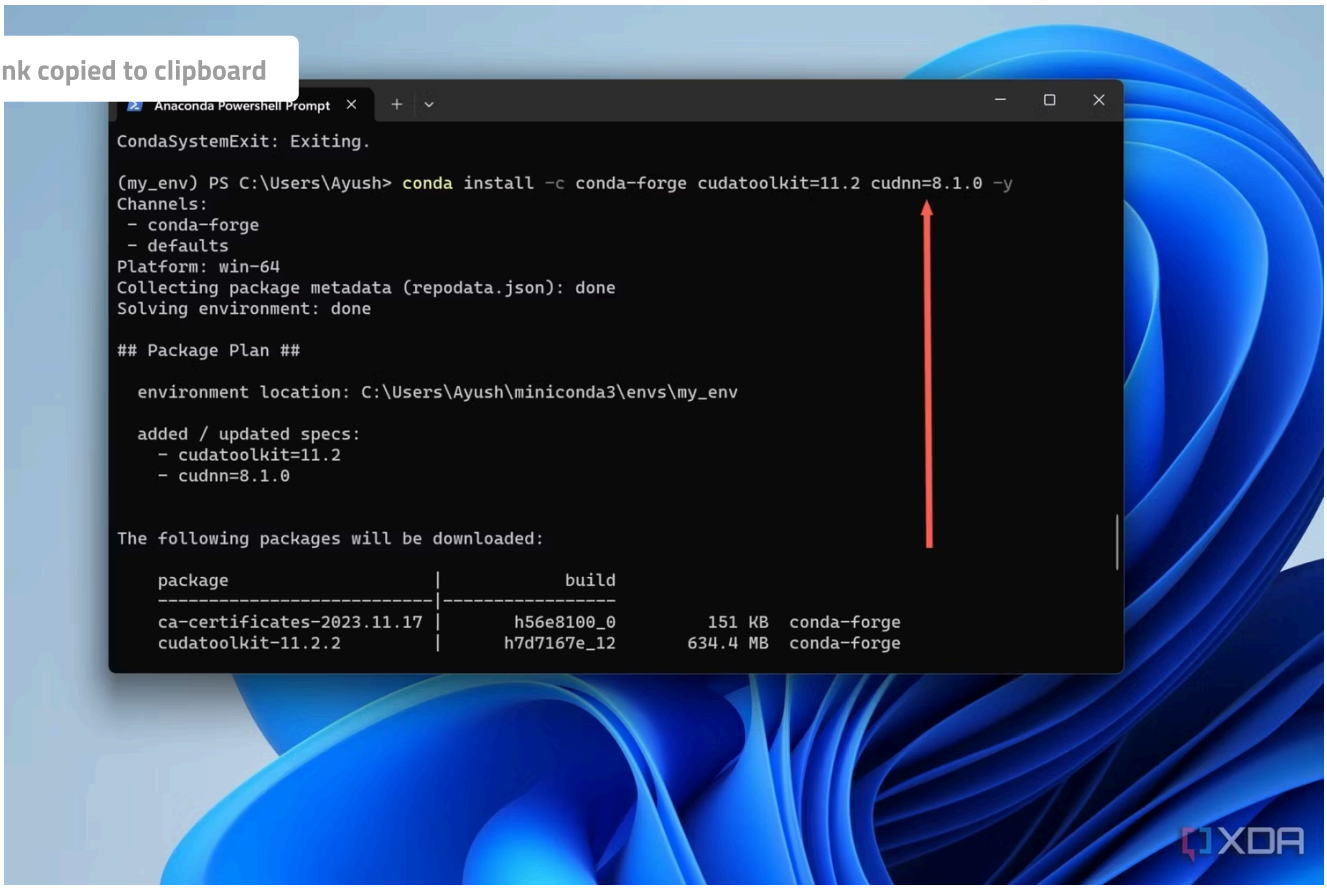
```
conda activate my_env
```



4. Run this command to install the cuDNN library and CUDA drivers:

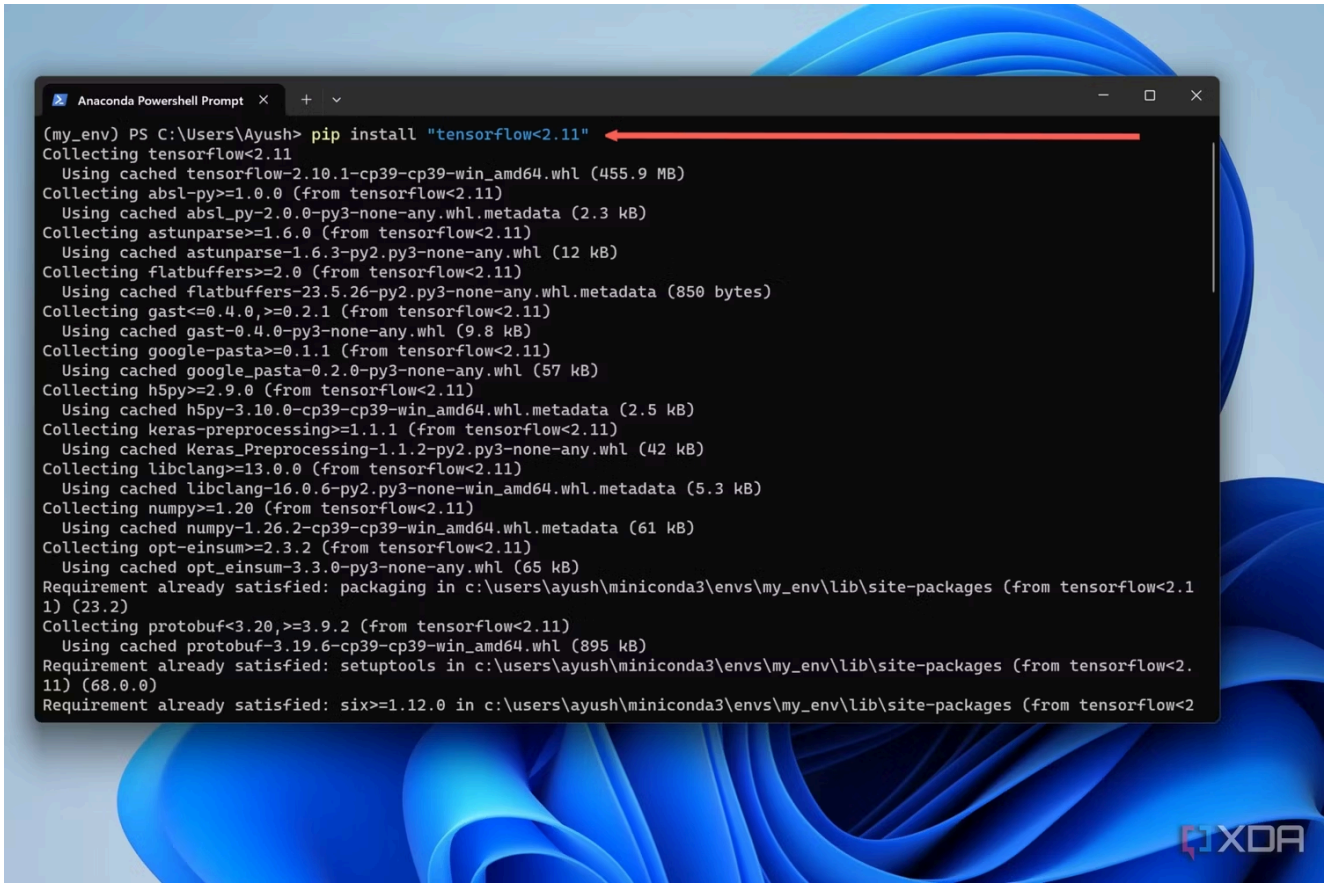
```
conda install -c conda-forge cudatoolkit=11.2 cudnn=8.1.0 -y
```

Link copied to clipboard



5. Install the TensorFlow library by running the following command:

```
pip install "tensorflow<2.11"
```



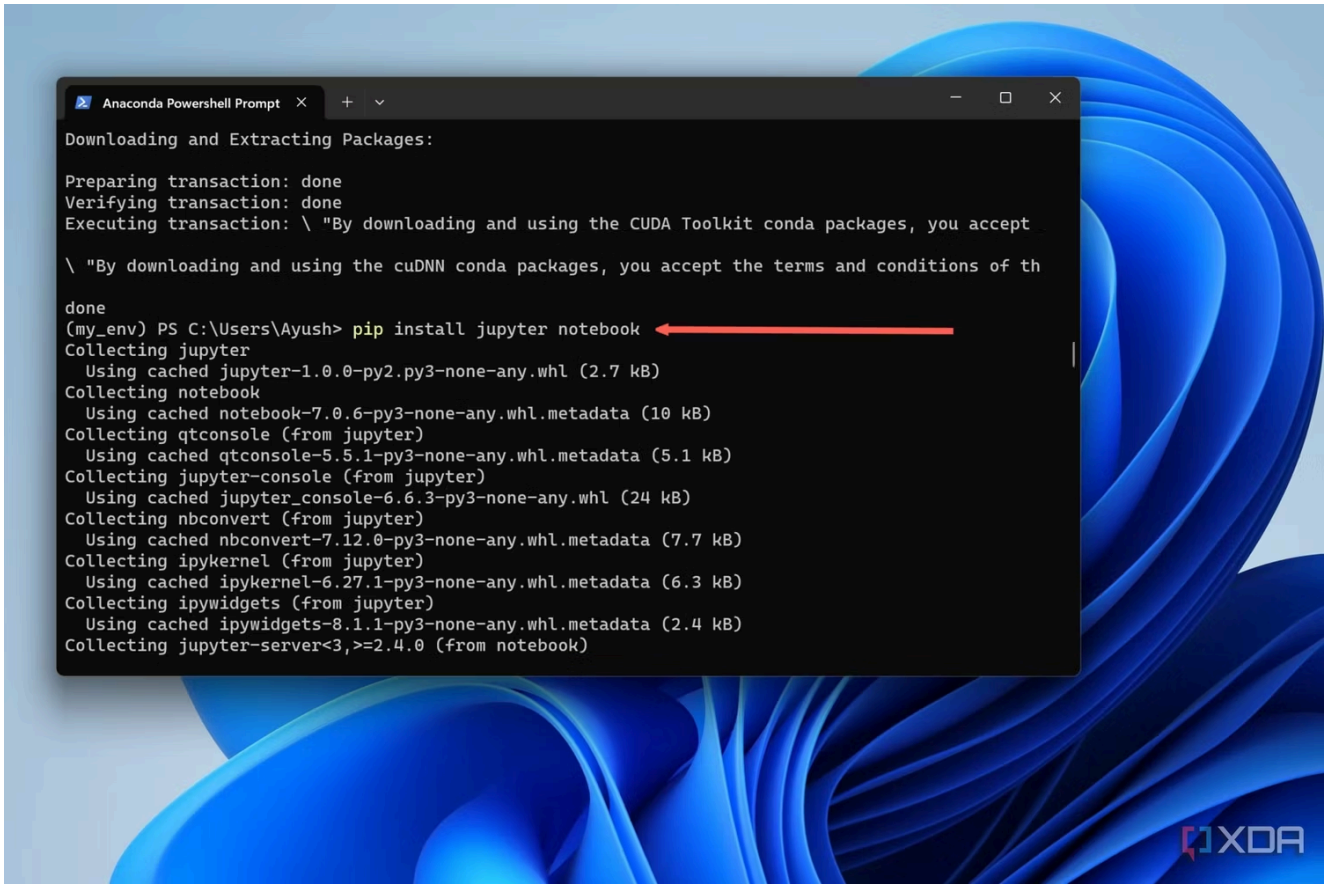
Installing Jupyter Notebook

Link copied to clipboard

Finally, you can set up a local Jupyter Notebook server containing all your project files.

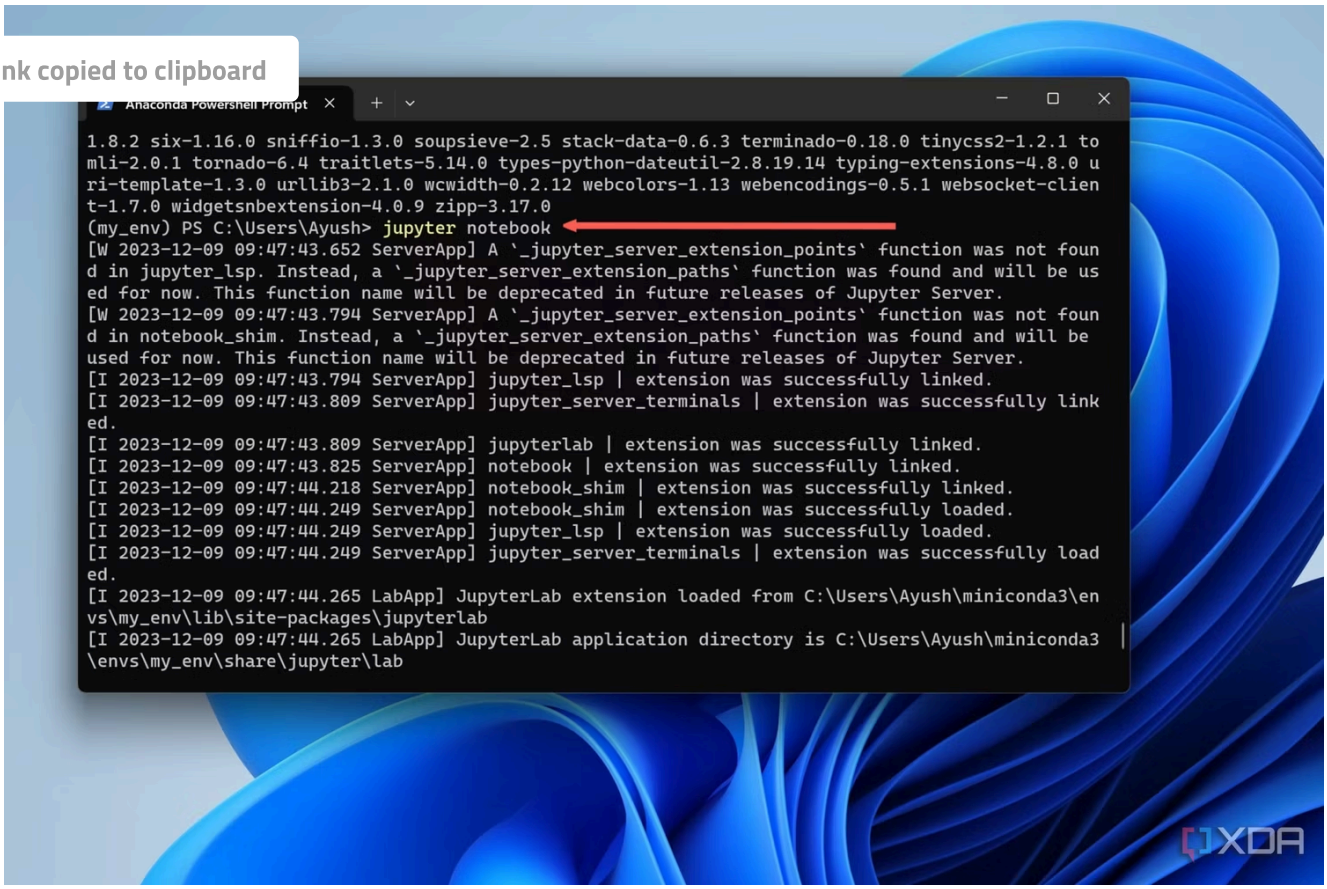
1. Run this code inside the **Anaconda Powershell Prompt**:

```
pip install jupyter notebook -y
```



2. Open the Jupyter Notebook server by typing:

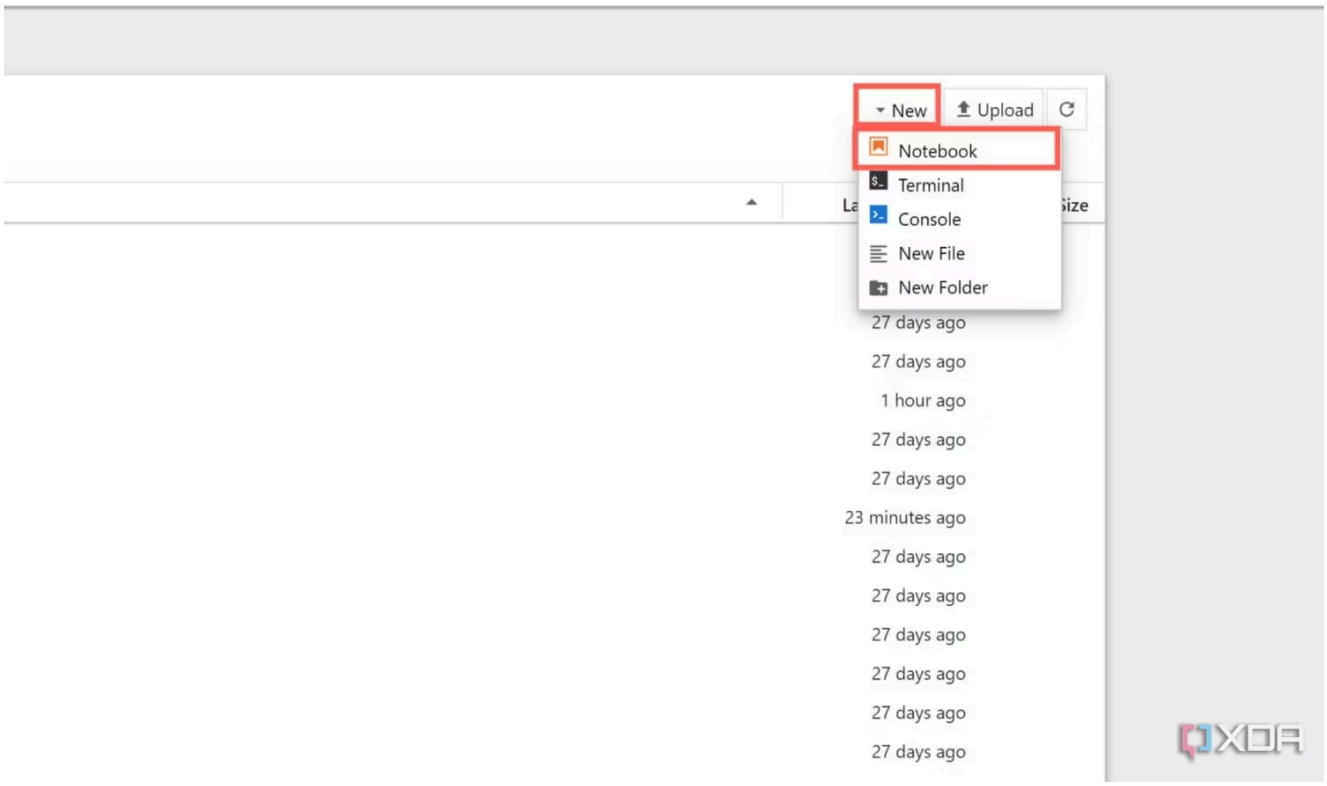
```
jupyter notebook
```



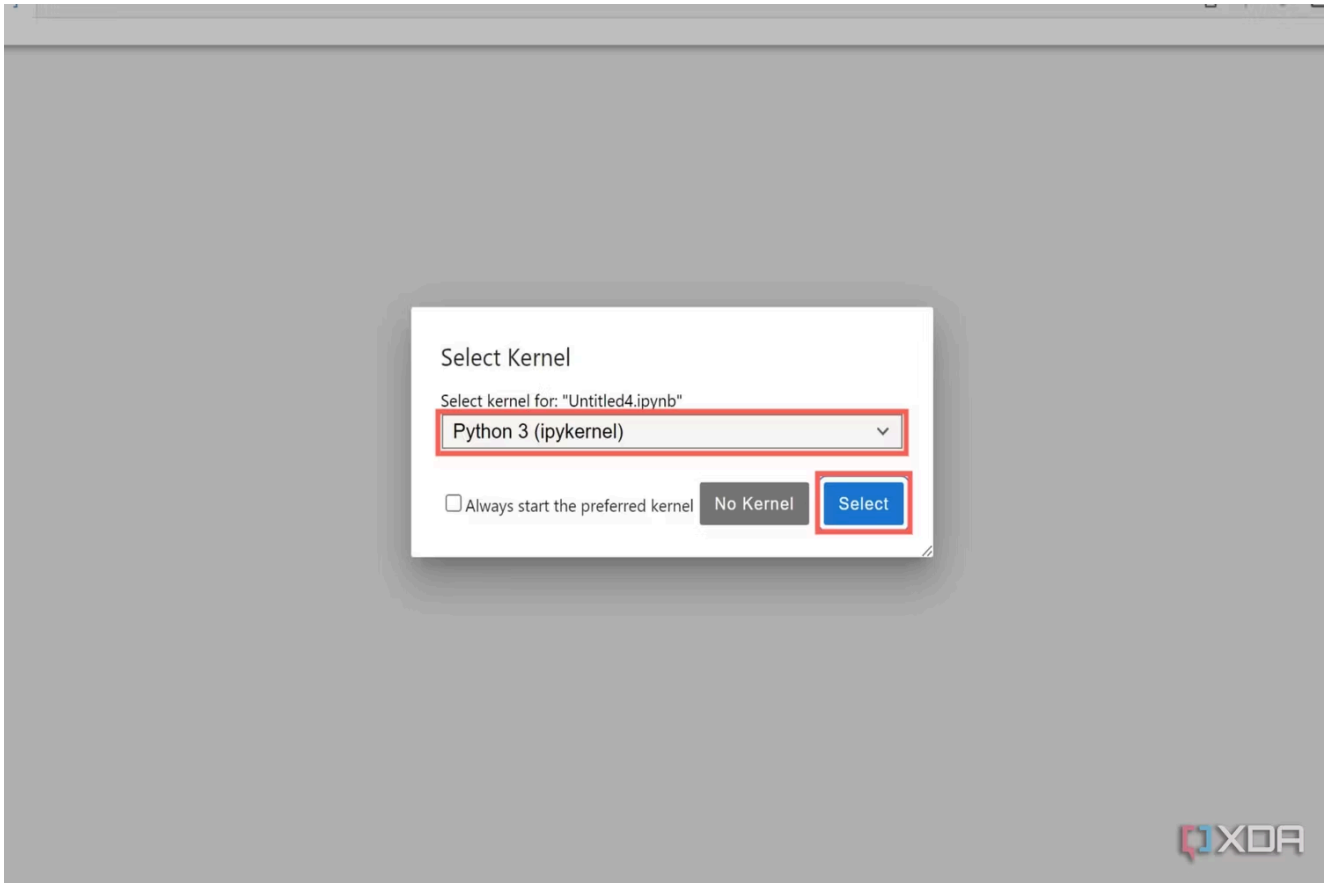
You can check if the Miniconda coding environment works with the GPU. To do so,

1. Click on the **New** button and choose **Notebook**.

Link copied to clipboard



2. Select **Python 3 (ipykernel)** as the kernel.



3. Copy these lines of code inside the newly created Notebook:

Link copied to clipboard

```
low as tf
```

```
gpus = tf.config.list_physical_devices("GPU")
```

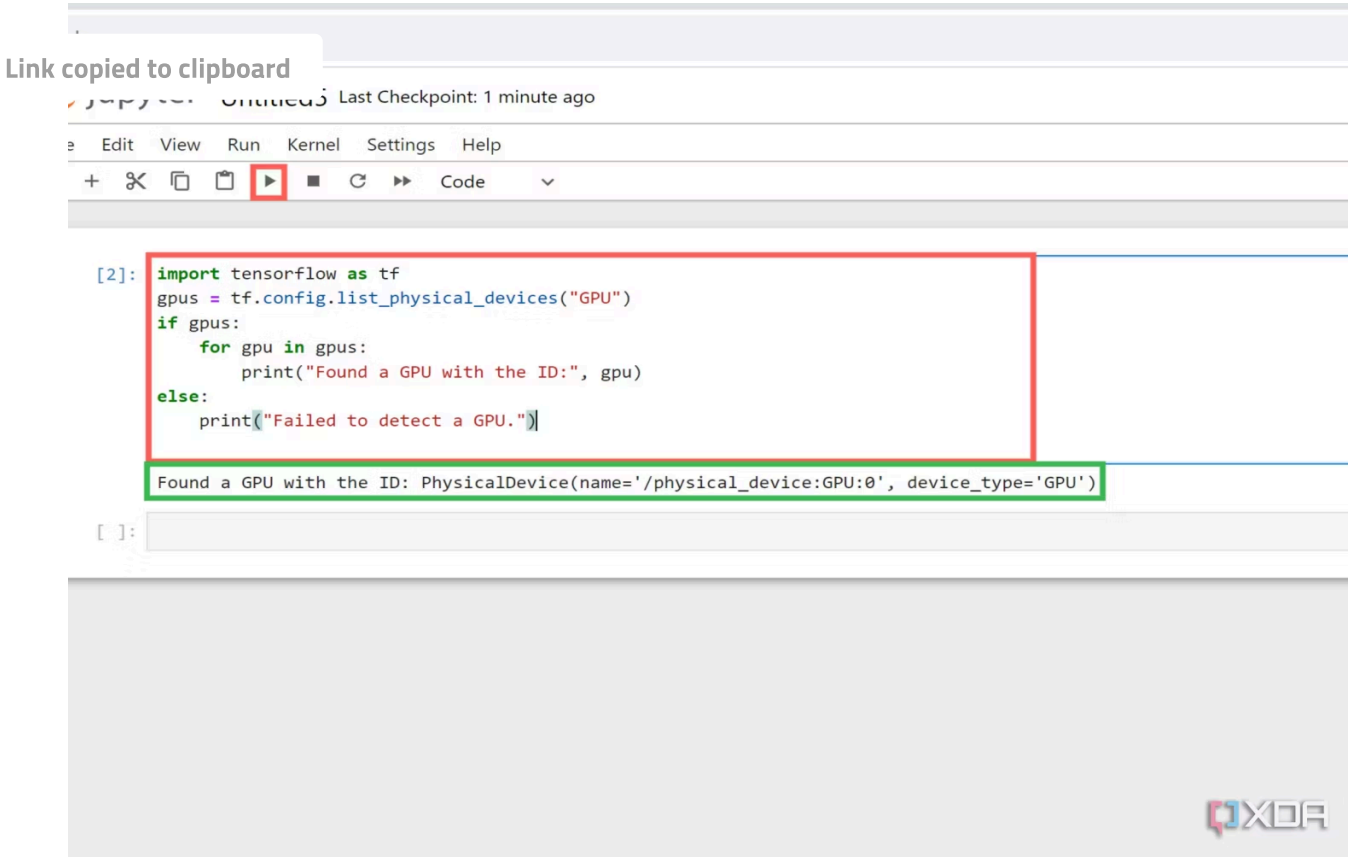
```
if gpus:
```

```
for gpu in gpus:
```

```
print("Found a GPU with the name:", gpu)
```

```
else:
```

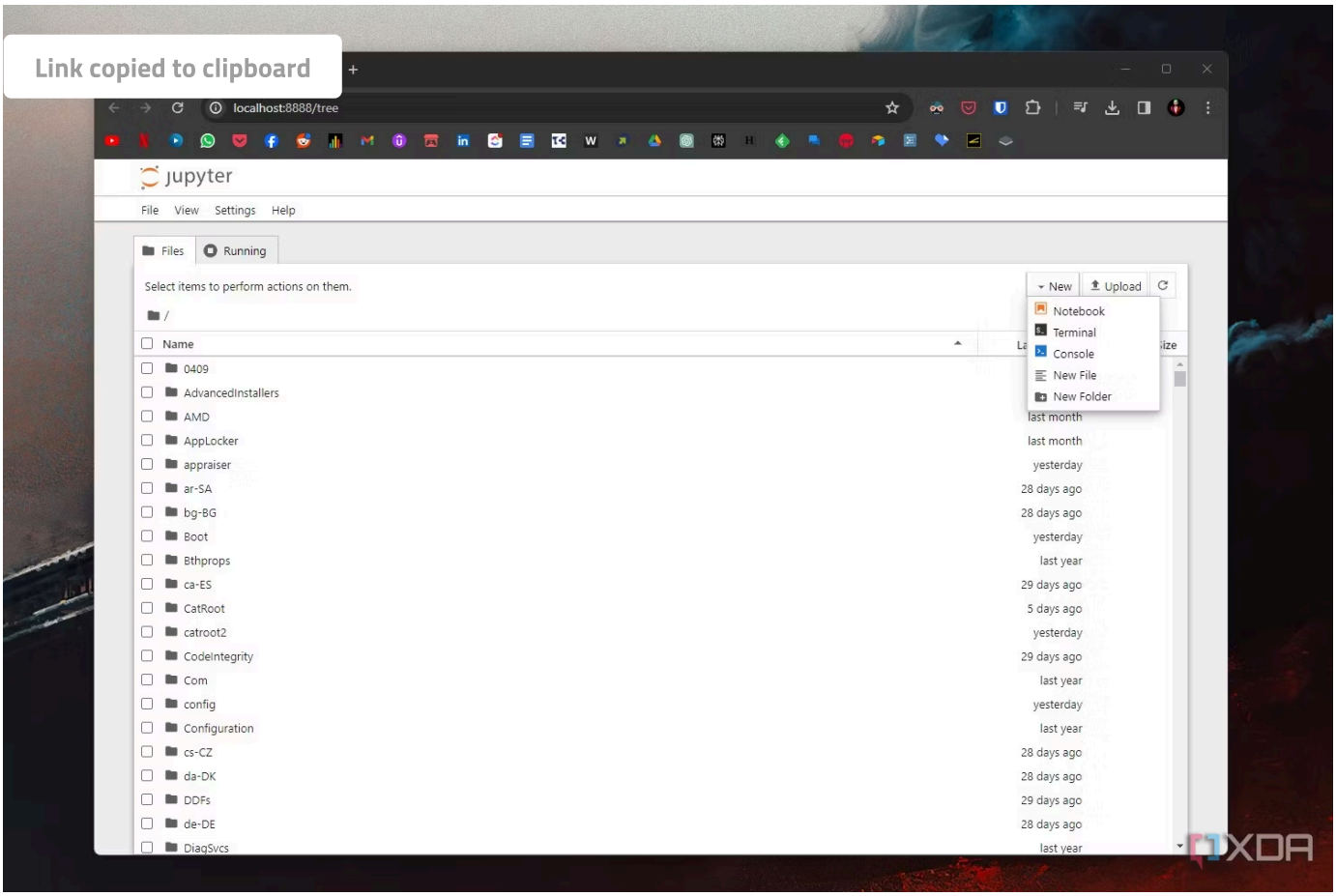
```
print("Failed to detect a GPU.")
```



4. Press the **Run** button.

If Jupyter Notebook displays a graphics card as the output, it means the process was successful!

Running Jupyter Notebook on a GPU



Once you've verified that the graphics card works with Jupyter Notebook, you can use the *import tensorflow* code snippet to leverage your GPU in all your machine-learning projects. In case Jupyter Notebook is unable to detect your graphics card, you can retry the same procedure in another Conda environment. Be sure to install the same versions of the CUDA drivers and the cuDNN and TensorFlow libraries as I've used in this tutorial to avoid running into compatibility issues.

If your projects take eons to compile, your graphics card could be lacking in horsepower. Upgrading to a better GPU is an easy fix that will give your PC the much-needed boost to run complex AI and deep learning algorithms.

RELATED

Best GPUs for deep learning in 2024

Readers like you help support XDA Developers. When you make a purchase using links on our site, we may earn an affiliate commission. ***Read More****.*

Link copied to clipboard

)aily Newsletter! 80,000+ People Already Do

Email Address

SUBSCRIBE

By subscribing, you agree to our Privacy Policy and may receive occasional deal communications; you can unsubscribe anytime.

Related Topics

GPU GPU CODING PYTHON

About The Author

Ayush Pande

(192 Articles Published)

Ayush Pande is a PC hardware and gaming journalist. When he's not writing articles, you can find him tinkering with computers, playing long RPGs, or strumming a guitar.

Recommended Articles

NVIDIA

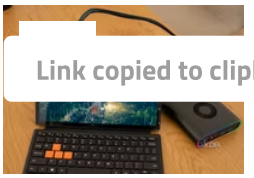
Nvidia GeForce RTX 5000 series: All of the rumors so far

NVIDIA

Light your money on fire instead of buying these 7 graphics cards

GPU

What is ATX 3.0 and why you need to know before your next PC upgrade



GPU

Link copied to clipboard an OCuLink external GPU — is it actually better than USB4?

DELL

Dell XPS 14 (2024) review: It's just so much fun

AMD

6 reasons AMD is better than Intel for the average gamer

XDA Developers is part of the **Valnet Publishing Group**

Link copied to clipboard

Copyright © 2024 Valnet Inc.