

Installation Guide (Windows)

-This is intended to cover all the steps required to get the dependencies of the environment without using Docker images.

!! disclaimer: you need to have at least 3GB of space in your disk to get all the PyTorch related packages!!

NerfStudio doesn't not support MacOS operation system - at least not with some serious digging and container work around solutions. So let's switch to good (or bad) old windows.

1. For installing NerfStudio you would need to install `git` first. Navigate to your command prompt and type `winget -version`. In case you get an answer "'winget' is not recognized as the name of a cmdlet" here [Winget Installation](#) you can get it. You would likely get the version as an answer though since winget is available for most -if not all- versions of Windows 10 and on Windows 11. If everything goes to hell follow those instructions on [how to get windows to recognise winget](#). Stay in your command prompt and type `winget install --id Git.Git -e --source winget`. This will get you git.
2. [Download Microsoft Visual Studio](#) (note Visual Studio Code is something different) and install it. Check the box "Desktop Development with C++".
3. We need conda for creating environments in VS studio. [Here](#) you can find the `Miniconda` installer links 😊 After the installation is finished you will have to work from the Anaconda Prompt. Just type Anaconda Prompt in the search tab and you will see it on top.
4. You should now create a folder for your project in a directory of your preference and navigate to it from the Anaconda Command Prompt by typing `cd path/to/your/folder`. We need to be inside the folder of the project to create an environment for the workspace. Before you create it, you should check the version of your Python by typing `python --version` or `python3 --version`. Make sure you have anaconda with `python>=3.8` since

earlier versions are NOT supported by PyTorch . Note that if you have previous versions of anaconda the python version supported there might be different from the one you have installed locally. To check that, go to cmd or Powershell, write `python --version` if this is a newest version you should return to anaconda prompt and update it. For this you need `conda update python` or if this is not resolving the issue update anaconda with `conda update conda` so it can support different versions of python. Last step type `conda install=pythonversion` with pythonversion being the one installed in your system. You could also create an environment with the version of Python you prefer but make sure to choose the interpreter accordingly when you use VS Code 😊 . Be careful with Python installations, having different versions installed locally may cause conflicts. In case python is not being recognized and before thinking it is not installed navigate to the environmental variables and check whether the executable it is included on your system's path. Here is [a tutorial on how to install Python](#) and what to do to ensure it is in your path. Use python version < 3.12 since the latest open3d package from PyPi is only supported by 3.8 to 3.11 versions. Now you are ready to create and activate the environment by typing the following three commands replacing 3.8 with your own python version.

- a. `conda create --name nerfstudio -y python=3.8`
- b. `conda activate nerfstudio`
- c. `python -m pip install --upgrade pip`

5. Now for installing PyTorch with Cuda) you HAVE TO find a windows computer with NVIDIA GPU (good luck 🙄). Some say PyTorch AMD can run on top of the Radeon Open Compute Stack (ROCm) but this lacks the documentation and it is probably cause issues with NeRF compatibility, it is also not available as an option on PyTorch official documentation. Now, go to NVIDIA control panel, system information and check the driver version and make sure it is supported by CUDA. For CUDA 11.8 that is the recommended version in our case, make sure you have NVIDIA driver version 452.39 or higher. If not, [update](#)

NVIDIA drivers by manually selecting your videocard type. Best alternative is to get GeForce experience installed so you can get check for updates at anytime. You can try to follow the official documentation but I wouldn't suggest it since they have some errors with CUDA dependencies. In my case using driver version 536.23 I used CUDA Toolkit 11.8. This is the best option to get cuda to work with pytorch. First navigate to your Program Files under Local Disk. Check if there is a folder NVIDIA GPU Computing Tools. If not, you have two options. Either install the toolkit from here or you type `coda install -c "nvidia/label/cuda-11.8.0" cuda-toolkit`. The second option did not work in my case, probably because the repository was not accessible or included in the package files. Either way you should add(C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin) to your path under user variables. Then go ahead and install the available version of PyTorch for CUDA cu121 using pip3 -> `pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121`. Lastly, install the `torch bindings pip install ninja git+https://github.com/NVlabs/tiny-cuda-nn/#subdirectory=bindings/torch`.

6. Install manually COLMAP and ffmpeg. Add to your path both the bin and lib folders for COLMAP.
7. Now for the last step, install nerfstudio using pip following those steps:
 - a. `git clone https://github.com/nerfstudio-project/nerfstudio.git`
 - b. `cd nerfstudio`
 - c. `pip install --upgrade pip setuptools`
 - d. `pip install -e .`