

# GET STARTED

Select references and run the command to install PyTorch locally, or get started quickly with one of the supported cloud platforms.

	Start Locally	PyTorch 2.x	Start via Cloud Partners	Previous PyTorch Versions	Execute Torch
--	---------------	-------------	--------------------------	---------------------------	---------------

## START LOCALLY

Select your references and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. You can also **install previous versions of PyTorch**. Note that Lightning is only available for C++.

**NOTE:** Latest PyTorch requires Python 3.9 or later.

Language	Python			C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.4	CUDA 12.6	ROCm 6.2.4	CP
Run this Command :	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118				

### Shortcuts

#### Prerequisites

Supported Linux Distributions

Python

Package Manager

#### Installation

Anaconda

Pip

#### Verification

Building from source

Prerequisites

## Installing on Linux

PyTorch can be installed and used on various Linux distributions. Depending on your system and compute requirements, your experience with PyTorch on Linux may vary in terms of processing time. It is recommended, but not required, that your Linux system has an NVIDIA or AMD GPU in order to harness the full power of PyTorch's CUDA support or ROCm support.

## PREREQUISITES

### Supported Linux Distributions

PyTorch is supported on Linux distributions that use `glibc >= v2.17`, which include the following:

- Arch Linux, minimum version 2012-07-15
- CentOS, minimum version 7.3-1611
- Debian, minimum version 8.0
- Fedora, minimum version 24
- Mint, minimum version 14
- OpenSUSE, minimum version 42.1
- PCLinuxOS, minimum version 2014.7
- Slackware, minimum version 14.2
- Ubuntu, minimum version 13.04

The install instructions here will generally apply to all supported Linux distributions. An example difference is that your distribution may support `yum` instead of `apt`. The specific examples shown were run on an Ubuntu 18.04 machine.

## Python

Python 3.9-3.12 is generally installed by default on any of our supported Linux distributions, which meets our recommendation.

*Tip: By default, you will have to use the command `python3` to run Python. If you want to use just the command `python`, instead of `python3`, you can symlink `python` to the `python3` binary.*

However, if you want to install another version, there are multiple ways:

- APT
- Python website

If you decide to use APT, you can run the following command to install it:

```
sudo apt install python
```

*If you use Anaconda to install PyTorch, it will install a sandboxed version of Python that will be used for running PyTorch applications.*

## Package Manager

To install the PyTorch binaries, you will need to use one of two supported package managers: **Anaconda** or **Pip**. Anaconda is the recommended package manager as it will provide you all of the PyTorch dependencies in one, sandboxed, installation, including Python.

### Anaconda

To install Anaconda, you will use the **conda** -line installer. Right-click on the 64-bit installer link, select **Copy Link Location**, and then use the following commands:

```
# The version of Anaconda may be different depending on when you are installing'
curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sh Miniconda3-latest-Linux-x86_64.sh
# and follow the prompts. The defaults are generally good.'
```

You may have to open a new terminal or re-source your `~/.bashrc` to get access to the `conda` command.

## pip

`python3`

While Python 3.x is installed by default on Linux, `pip` is not installed by default.

```
sudo apt install python3-pip
```

*Tip: If you want to use just the command `pip`, instead of `pip3`, you can symlink `pip` to the `pip3` binary.*

## INSTALLATION

### Anaconda

#### With CUDA/ROCm

To install PyTorch via Anaconda, and do not have a **CUDA-enabled** or **ROCm-enabled** system or do not require CUDA/ROCm (i.e. GPU support), in the above selector, choose OS: Linux, Package: Conda, Language: Python and Compute Platform: CP. Then, run the command that is presented to you.

#### With CUDA

To install PyTorch via Anaconda, and you have a **CUDA-enabled** system, in the above selector, choose OS: Linux, Package: Conda and the CUDA version suite to your machine. Often, the latest CUDA version is better. Then, run the command that is presented to you.

#### With ROCm

PyTorch via Anaconda is not supported on ROCm currently. Please use `Pip` instead.

## pip

#### With CUDA

To install PyTorch via `Pip`, and do not have a **CUDA-enabled** or **ROCm-enabled** system or do not require CUDA/ROCm (i.e. GPU support), in the above selector, choose OS: Linux, Package: `Pip`, Language: Python and Compute Platform: CP. Then, run the command that is presented to you.

#### With CUDA

To install PyTorch via `Pip`, and you have a **CUDA-enabled** system, in the above selector, choose OS: Linux, Package: `Pip`, Language: Python and the CUDA version suite to your machine. Often, the latest CUDA version is better. Then, run the command that is presented to you.

#### With ROCm

To install PyTorch via `Pip`, and you have a **ROCm-enabled** system, in the above selector, choose OS: Linux, Package: `Pip`, Language: Python and the ROCm version suite. Then, run the command that is presented to you.

## VERIFICATION

To ensure that PyTorch was installed correctly, we can verify the installation by running sample PyTorch code. Here we will construct a randomly initialized `tensor`.

```
import torch
x = torch.randn(5, 3)
print(x)
```

The output should be something similar to:

```
tensor([[ 0.3380,  0.3845,  0.3217],
        [ 0.8337,  0.9050,  0.2650],
        [ 0.2979,  0.7141,  0.9069],
        [ 0.1449,  0.1132,  0.1375],
        [ 0.4675,  0.3947,  0.1426]])
```

Additionally, to check if your GPU driver and CUDA/ROCm is enabled and accessible by PyTorch, run the following commands to return whether or not the GPU driver is enabled (the ROCm utility of PyTorch uses the same semantics as the Python API level [link](#), so the below commands should also work for ROCm):

```
import torch
torch.cuda.is_available()
```

## BUILDING FROM SOURCE

For the majority of PyTorch users, installing from a pre-built binary via a package manager will provide the best experience. However, there are times when you may want to install the latest development PyTorch code, whether for testing or actual development on the PyTorch core. To install the latest PyTorch code, you will need to [build PyTorch from source](#).

## Prerequisites

- Install **Anaconda** or **Pip**
- If you need to build PyTorch with GPU support a. for NVIDIA GPUs, install **CUDA**, if your machine has a **CUDA-enabled** GPU. b. for AMD GPUs, install **ROCm**, if your machine has a **ROCm-enabled** GPU
- Follow the steps described here: <https://github.com/pytorch/pytorch#from-source>

You can verify the installation as described above.

## Stay in touch for updates, event info, and the latest news

First Name*	Last Name*	Email*	Select Country*	SUBMIT
-------------	------------	--------	-----------------	--------

By submitting this form, I consent to receive marketing emails from the LF and its projects regarding their events, training, research, developments, and related announcements. I understand that I can unsubscribe at any time using the links in the footers of the emails I receive. [Privacy Policy](#).