

Steps to Install PyTorch or TensorFlow on NVIDIA A100 (Cloud VM) & macOS (Apple Silicon M1-M4)

◆ Installing PyTorch or TensorFlow on NVIDIA A100 (Cloud VM)

Prerequisites:

- ✓ Ensure **CUDA & cuDNN** are installed.
- ✓ Verify **NVIDIA drivers** using `nvidia-smi`.

✅ Step 1: Install CUDA & cuDNN

• Update package lists

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

• Install NVIDIA CUDA Toolkit (latest version for A100)

```
sudo apt install -y cuda-toolkit-12-2
```

• Install cuDNN (deep learning acceleration)

```
sudo apt install -y libcudnn8 libcudnn8-dev
```

• Set environment variables

```
echo 'export PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH' >>
~/.bashrc
source ~/.bashrc
```

✅ Step 2: Install PyTorch with GPU Support

• Install PyTorch with CUDA support

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

- **Verify PyTorch installation**

```
import torch
print(torch.cuda.is_available()) # Should return True
print(torch.cuda.get_device_name(0)) # Should return NVIDIA A100
```

Step 3: Install TensorFlow with GPU Support

- **Install TensorFlow GPU version**

```
pip install tensorflow
```

- **Verify TensorFlow installation**

```
import tensorflow as tf
print(tf.config.list_physical_devices('GPU')) # Should return ['GPU:0']
```

Installing PyTorch or TensorFlow on macOS (Apple Silicon M1-M4)

Prerequisites:

- ✓ Install **Homebrew** (brew).
- ✓ Use **Miniforge (Recommended)** instead of Anaconda.
- ✓ Install **Metal Performance Shaders (MPS) backend** for GPU acceleration.

Step 1: Install Dependencies

- **Install Miniforge (Recommended)**

```
curl -L -O https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-MacOSX-arm64.sh
bash Miniforge3-MacOSX-arm64.sh
```

- **Create a Conda environment**

```
conda create --name tf-metal python=3.10 -y
conda activate tf-metal
```

- **Install dependencies**

```
brew install cmake libomp
```

- ✓ **Step 2: Install PyTorch with MPS (Apple Silicon)**

- **Install PyTorch with Metal (MPS) backend**

```
pip install torch torchvision torchaudio
```

- **Verify PyTorch installation**

```
import torch
print(torch.backends.mps.is_available()) # Should return True for Apple
Silicon GPU
```

- ✓ **Step 3: Install TensorFlow with Metal Acceleration**

- **Install TensorFlow and Metal Plugin**

```
conda install -c apple tensorflow-deps
pip install tensorflow-macos
pip install tensorflow-metal # Enables GPU acceleration
```

- **Verify TensorFlow installation**




```
import tensorflow as tf
print(tf.config.list_physical_devices('GPU')) # Should return ['GPU:0']
```

Summary Table

Task	Command (A100 Cloud VM)	Command (Mac M1-M4)
Update System	<code>sudo apt update && sudo apt upgrade -y</code>	<code>brew update</code>
Install CUDA & cuDNN	<code>sudo apt install -y cuda-toolkit-12-2 libcudnn8</code>	✗ Not needed (Apple MPS)
Install PyTorch	<code>pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121</code>	<code>pip install torch torchvision torchaudio</code>
Install TensorFlow w	<code>pip install tensorflow</code>	<code>pip install tensorflow-macos tensorflow-metal</code>

Task	Command (A100 Cloud VM)	Command (Mac M1-M4)
Check PyTorch GPU	<code>print(torch.cuda.is_available())</code>	<code>print(torch.backends.mps.is_available())</code>
Check TensorFlow w GPU	<code>print(tf.config.list_physical_devices('GPU'))</code>	<code>print(tf.config.list_physical_devices('GPU'))</code>

Next Steps

-  If using PyTorch, enable **MPS for Mac** or **CUDA for A100**.
-  If using TensorFlow, **install Metal on Mac** or **check TensorRT support on A100**.
-  Optimize inference with **TensorRT for A100** or **CoreML for macOS**.