

Here's a comprehensive, step-by-step guide for installing and using Miniforge on your M1 MacBook Pro, which is an excellent solution for managing Python and AI/ML environments:

1. Download Miniforge:

- Open your web browser and go to: <https://github.com/conda-forge/miniforge/releases>
- Find the latest release for “Miniforge3-MacOSX-arm64.sh”
- Click on the file name to download it

2. Install Miniforge:

- Open Terminal (you can find it in Applications > Utilities > Terminal)
- Navigate to your Downloads folder:

```
cd ~/Downloads
```

- Make the installer executable:

```
chmod +x Miniforge3-MacOSX-arm64.sh
```

- Run the installer:

```
./Miniforge3-MacOSX-arm64.sh
```

- Follow the prompts. When asked, type “yes” to agree to the license terms
- When asked if you want to initialize Miniforge3, type “yes”

- Close and reopen Terminal to apply changes

- Disable auto-activation of the base environment:

```
conda config --set auto_activate_base false
```

4. Create a new environment:

- Create an environment named “aiml” with Python 3.10:

```
conda create -n aiml python=3.10
```

- Activate the new environment:

```
conda activate aiml
```

5. Install common AI/ML packages:

```
conda install numpy pandas scikit-learn matplotlib  
conda install -c pytorch pytorch torchvision torchaudio
```

6. Verify the installation:

```
python -c "import torch; print(torch.backends.mps.is_available())"
```

This should print “True” if PyTorch is correctly installed and can use the M1 GPU.

7. Create a central project directory:

```
mkdir ~/AIMLProjects
cd ~/AIMLProjects
```

8. For each new project:

- Create a project folder:

```
mkdir MyNewProject
cd MyNewProject
```
- Create a symbolic link to the shared environment:

```
ln -s ~/miniforge3/envs/aiml ./env
```
- Activate the environment:

```
conda activate ./env
```

9. Install project-specific packages (if needed):

```
conda install package_name
```

10. Create a requirements file:

```
conda list --export > requirements.txt
```

11. Deactivate the environment when done: `conda deactivate`

By following these steps, you’ll have a centralized AI/ML environment that you can easily link to individual projects, saving disk space while maintaining flexibility. Remember to activate the environment each time you work on a project, and update your requirements.txt file when you install new packages.

My Mistake :- To activate this environment, use:

```
micromamba activate /Users/rajesh/miniforge3
```

Or to execute a single command in this environment, use:

```
micromamba run -p /Users/rajesh/miniforge3 mycommand
```

installation finished. Do you wish to update your shell profile to automatically initialize conda? This will activate conda on startup and change the command prompt when activated. If you’d prefer that conda’s base environment not be activated on startup, run the following command when conda is activated:

```
conda config --set auto_activate_base false
```

You can undo this by running `conda init --reverse $SHELL?` [yes|no] [no] »>
no

You have chosen to not have conda modify your shell scripts at all. To activate conda's base environment in your current shell session:

```
eval "$(/Users/rajesh/miniforge3/bin/conda shell.YOUR_SHELL_NAME hook)"
```

To install conda's shell functions for easier access, first activate, then:

```
conda init
```

Thank you for installing Miniforge3!

**** Final Commands to run after installation **:**

```
eval "$(/Users/rajesh/miniforge3/bin/conda shell.zsh hook)"
conda init
conda activate ./env
```