

LLM Engineering - Master AI and LLMs

Setup instructions for Linux

Welcome, Linux people!

I should reveal that I had ChatGPT make this document based on the Mac instructions, but then I went through and checked and tweaked some sections. If any of these instructions don't work for your distro, please do reach out and let me know - we'll figure it out, then I'll update the instructions for the future.

Setting up a powerful environment to work at the forefront of AI requires some effort, but these instructions should guide you smoothly. If you encounter any issues, don't hesitate to reach out to me. I'm here to ensure you get set up without hassle.

Email: ed@edwarddonner.com

LinkedIn: <https://www.linkedin.com/in/eddonner/>

For this setup, we'll use Anaconda to create a reliable environment for your AI work. Alternatively, I've provided a lighter option if you prefer to avoid Anaconda. Let's get started!

Part 1: Clone the Repo

This gets you a local copy of the code on your machine.

1. **Install Git** if not already installed:

- Open your terminal.
- Run `git --version`. If Git isn't installed, follow the instructions for your distribution:
 - Debian/Ubuntu: `sudo apt update && sudo apt install git`
 - Fedora: `sudo dnf install git`
 - Arch: `sudo pacman -S git`

2. **Navigate to your projects folder:**

If you have a specific folder for projects, navigate to it using the `cd` command.

For example: `cd ~/Projects`

If you don't have a projects folder, you can create one:

```
mkdir ~/Projects
```

```
cd ~/Projects
```

3. **Clone the repository:**

Run the following command in your terminal: `git clone https://github.com/ed-donner/llm_engineering`

This creates a new directory `llm_engineering` within your `Projects` folder and downloads the course code. Use `cd llm_engineering` to enter the directory. This is your “project root directory.”

Part 2: Install Anaconda environment

If this Part 2 gives you any trouble, refer to the alternative Part 2B below.

1. Install Anaconda:

- Download the Linux installer from <https://www.anaconda.com/download>.
- Open a terminal and navigate to the folder containing the downloaded `.sh` file.
- Run the installer: `bash Anaconda3*.sh` and follow the prompts. Note: This requires about 5+ GB of disk space.

2. Set up the environment:

- Open a terminal and navigate to the “project root directory” using: `cd ~/Projects/llm_engineering` (adjust the path as necessary).
- Run `ls` to confirm the presence of subdirectories for each week of the course.
- Create the environment: `conda env create -f environment.yml`

This may take several minutes (even up to an hour for new Anaconda users). If it takes longer or errors occur, proceed to Part 2B.

- Activate the environment: `conda activate llms`.

You should see `(llms)` in your prompt, indicating successful activation.

3. Start Jupyter Lab:

From the `llm_engineering` folder, run: `jupyter lab`.

Jupyter Lab should open in your browser. Close it after confirming it works, and proceed to Part 3.

Part 2B - Alternative to Part 2 if Anaconda gives you trouble

1. Install Python 3.11 (if not already installed):

- Debian/Ubuntu: `sudo apt update && sudo apt install python3.11`
- Fedora: `sudo dnf install python3.11`
- Arch: `sudo pacman -S python`

2. Navigate to the project root directory:

Use `cd ~/Projects/llm_engineering` and verify the folder contents with `ls`.

3. Create a virtual environment:

Run: `python3.11 -m venv llms`

4. Activate the virtual environment:

Use: `source llms/bin/activate`

Your prompt should now display (llms), indicating the environment is active.

5. Install required packages:

Run: `python -m pip install --upgrade pip` followed by `pip install -r requirements.txt`.

If issues occur, try the fallback: `pip install --retries 5 --timeout 15 --no-cache-dir --force-reinstall --verbose -r requirements.txt`

6. Start Jupyter Lab:

From the `llm_engineering` folder, run: `jupyter lab`.

Part 3 - OpenAI key (OPTIONAL but recommended)

Particularly during weeks 1 and 2 of the course, you'll be writing code to call the APIs of Frontier models (models at the forefront of AI).

For week 1, you'll only need OpenAI, and you can add the others if you wish later on.

1. Create an OpenAI account if you don't have one by visiting: <https://platform.openai.com/>
2. OpenAI asks for a minimum credit to use the API. For me in the US, it's \$5. The API calls will spend against this \$5. On this course, we'll only use a small portion of this. I do recommend you make the investment as you'll be able to put it to excellent use. But if you'd prefer not to pay for the API, I give you an alternative in the course using Ollama.

You can add your credit balance to OpenAI at Settings > Billing:

<https://platform.openai.com/settings/organization/billing/overview>

I recommend you disable the automatic recharge!

3. Create your API key

The webpage where you set up your OpenAI key is at <https://platform.openai.com/api-keys> - press the green 'Create new secret key' button and press 'Create secret key'. Keep a record of the API key somewhere private; you won't be able to retrieve it from the OpenAI screens in the future. It should start `sk-proj-`.

In week 2 we will also set up keys for Anthropic and Google, which you can do here when we get there.

- Claude API at <https://console.anthropic.com/> from Anthropic - Gemini API

at <https://ai.google.dev/gemini-api> from Google

Later in the course you'll be using the fabulous HuggingFace platform; an account is available for free at <https://huggingface.co> - you can create an API token from the Avatar menu » Settings » Access Tokens.

And in Week 6/7 you'll be using the terrific Weights & Biases at <https://wandb.ai> to watch over your training batches. Accounts are also free, and you can set up a token in a similar way.

PART 4 - .env file

When you have these keys, please create a new file called `.env` in your project root directory. The filename needs to be exactly the four characters `“.env”` rather than `“my-keys.env”` or `“.env.txt”`. Here's how to do it:

1. Open Terminal (Applications > Utilities > Terminal)
2. Navigate to the “project root directory” using `cd ~/Documents/Projects/llm_engineering` (replace this path with the actual path to the `llm_engineering` directory, your locally cloned version of the repo).
3. Create the `.env` file with

```
nano .env
```

4. Then type your API keys into nano, replacing `xxxxx` with your API key (starting `sk-proj-`).

```
OPENAI_API_KEY=xxxxx
```

If you have other keys, you can add them too, or come back to this in future weeks:

```
GOOGLE_API_KEY=xxxxx
ANTHROPIC_API_KEY=xxxxx
HF_TOKEN=xxxxx
```

5. Save the file:

```
Control + O
Enter (to confirm save the file)
Control + X to exit the editor
```

6. Use this command to list files in your project root directory:

```
ls -a
```

And confirm that the `.env` file is there.

This file won't appear in Jupyter Lab because jupyter hides files starting with a dot. This file is listed in the `.gitignore` file, so it won't get checked in and your keys stay safe.

Part 5 - Showtime!!

1. Open a terminal.
2. Navigate to the “project root directory” using: `cd ~/Projects/llm_engineering`.
3. Activate your environment:
 - If you used Anaconda: `conda activate llms`
 - If you used the alternative: `source llms/bin/activate`

You should see `(llms)` in your prompt. Run: `jupyter lab` to get started.

Enjoy your journey into mastering AI and LLMs!