Speech-to-Text Transcription with Deepgram

Marc Anthony Atanante

Introduction

In this activity, we will develop a Speech-to-Text transcription service using FastAPI and Deepgram (Credits to Tonya Sims for the documentation).

FastAPI is a new, innovative Python web framework gaining popularity because of its modern features like support for concurrency and asynchronous code and Deepgram is an AI speech platform that offers real-time audio transcription.

Step 1: Cloning the repository

The first step is to clone the following repository to your local machine. Make sure that you review and understand each file.

There are four main files required to create the Q&A API:

1. **main.py** - Main app file. This defines the FastAPI and Deepgram logic.
2. **requirements.txt** - List of dependencies to ensure a smooth run.
3. **templates/index.html** - HTML file for the home page.
4. **.env** - This environment file holds the Deepgram API key. This will be used in the main.py file for the Deepgram client.

Step 2: Install Dependencies

Make sure that your environment has the needed packages and versions. Use the command **pip install -r requirements.txt** to do this.

Note that I did not create a virtual environment for this task and I just used my local machine.

Step 3: Running the project

Make sure that you are in the directory of **main.py** file and run the project using the command **uvicorn main:app --reload**. A successful run will look like this:

```
-transcription-fastapi$ uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['/mnt/c/Users/user/D
/Deepgram Transcription/live-transcription-fastapi']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [63] using watchgod
INFO:     Started server process [65]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Step 4: Demonstration

Open a browser and go to your localhost. The page should display a “Connected” status and should look like this:

Next, allow access to your microphone and start speaking. A transcript of your audio will appear in the browser.

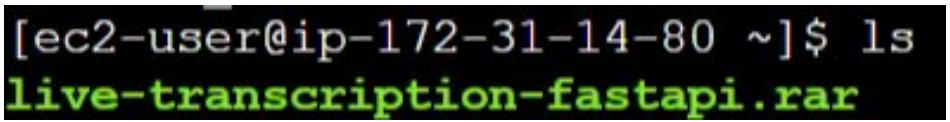
Step 4: Demonstration

The following is a 30 second excerpt from JRE MMA Show #137.

Step 5: Hosting on AWS EC2

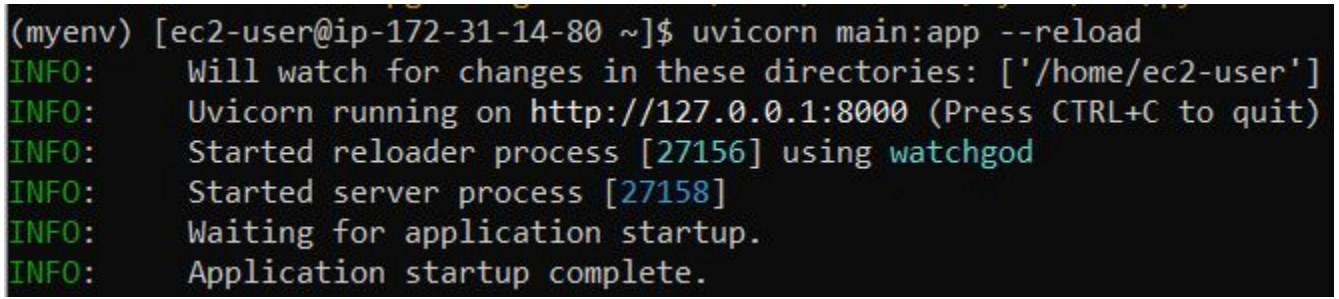
Connect to your EC2 instance and follow the steps below:

1. You might encounter different errors when trying to copy a file from your local machine to your EC2 instance. Personally, this trick worked using Windows Powershell:
    - icacls.exe key.pem /reset
    - icacls.exe key.pem /grant:r "$($env:username):(r)"
    - icacls.exe key.pem /inheritance:r
2. After that, send the zipped project folder to your EC2 instance using the command **sudo scp -i “your keys.pem” ./live-transcription-fastapi.rar ec2-user@”your public DNS”:/home/ec2-user**. After copying, your EC2 instance should contain the file:



Step 5: Hosting on AWS EC2

3. Unzip the .rar file using the following commands:
    - wget https://www.rarlab.com/rar/rarlinux-x64-621.tar.gz
    - tar -zxvf rarlinux-*.tar.gz
    - Cd rar -> ./unrar -> cp unrar /usr/local/bin -> unrar e live-transcription-fastapi.rar
    - sudo yum install python3-devel python3-pip
    - python3 -m venv myenv
    - source myenv/bin/activate
    - pip install -r requirements.txt
    - uvicorn main:app --reload
    - If you wish to exit the python environment, type deactivate.



Next Steps

I tried to install a desktop environment and browser so that I can demonstrate the task inside my EC2 using a VNC server. However, I encountered problems with the installation since I am using Amazon Linux 2023. But assuming smooth installations of all dependencies, then you should be able to run the page on your localhost.