

日々是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年1月24日 火曜日

OpenAI Whisperを使った文字起こしアプリの作成(3) - Flaskを使ったAPIサーバー

PythonのFlaskを使い、Whisperを呼び出して文字起こしをするAPIを実装します。

Ubuntu 20.04のコンピュート・インスタンスの方がWhisperの処理が速いため、Ubuntu上にAPIサーバーを実装します。

最終的にAutonomous Databaseに作成するAPEXアプリケーションから呼び出すため、ホスト名がDNSに登録されIPアドレスの解決ができること、HTTPS化がされていることが必要になります。

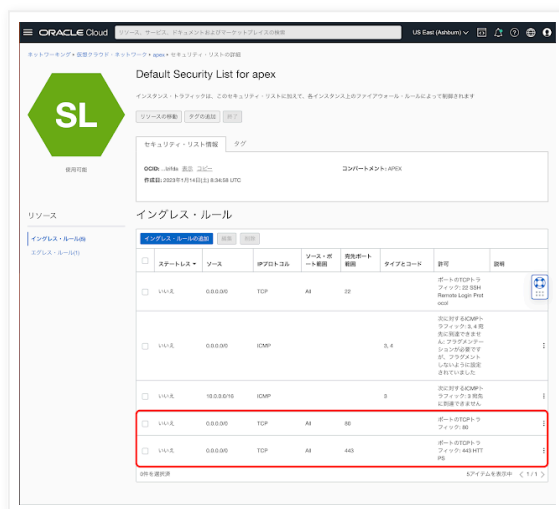
サーバー証明書は、Let's Encryptより発行します。

DNSへのホスト名とIPアドレスの登録は手順に含みません。hostコマンドの実行で、IPアドレスが返ってくるような状態になっていることが前提です。

host ホスト名

```
ubuntu@mywhisper2:~$ host ホスト名
ホスト名 has address ***.***.***.***
ubuntu@mywhisper2:~$
```

あらかじめ、パブリック・ネットワークの**イングレス・ルール**にて、ポート**80** (http) と**443** (https) への通信を許可しておきます。



以下より、APIサーバーの実装手順を記述します。作業のほとんどは、OS関連の設定です。

作業はユーザー**ubuntu**で実施します。

最初に受信した音声ファイルを一時保存するディレクトリ**audio**と、サーバー証明書と秘密キーを保存するディレクトリ**certs**を作成します。

```
cd
mkdir certs audio
```

```
ubuntu@mywhisper2:~$ cd
ubuntu@mywhisper2:~$ mkdir certs audio
ubuntu@mywhisper2:~$
```

構成に使用する**firewalld**と**certbot**をインストールします。

```
sudo apt install firewalld certbot
```

```
ubuntu@mywhisper2:~$ sudo apt install firewalld certbot
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

[中略]

```
Processing triggers for systemd (245.4-4ubuntu3.19) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
ubuntu@mywhisper2:~$
```

Flaskをインストールします。

```
pip install flask
```

```
ubuntu@mywhisper2:~$ pip install flask
Collecting flask
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB)
    |████████████████████████████████████████| 101 kB 13.2 MB/s
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
    |████████████████████████████████████████| 232 kB 76.9 MB/s
Collecting click>=8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    |████████████████████████████████████████| 96 kB 10.0 MB/s
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Requirement already satisfied: importlib-metadata>=3.6.0; python_version < "3.10"
in /usr/local/lib/python3.8/dist-packages (from flask) (5.1.0)
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    |████████████████████████████████████████| 133 kB 80.5 MB/s
Collecting MarkupSafe>=2.1.1
  Downloading MarkupSafe-2.1.2-cp38-cp38-
manylinux_2_17_aarch64.manylinux2014_aarch64.whl (26 kB)
Requirement already satisfied: zipp>=0.5 in /usr/lib/python3/dist-packages (from
importlib-metadata>=3.6.0; python_version < "3.10"->flask) (1.0.0)
Installing collected packages: MarkupSafe, Werkzeug, click, itsdangerous, Jinja2,
flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.2.2 click-8.1.3
flask-2.2.2 itsdangerous-2.1.2
ubuntu@mywhisper2:~$
```

firewalldを構成します。

ポート80と443の通信を許可します。ポート443は8443へ転送するよう設定します。Flaskは一般ユーザーで起動するため、ポート8443で待ち受けるようにします。Certbotがスタンドアロンのサーバーを起動し、Let's Encryptの認証局が発行するHTTPのチャレンジを受信するために、ポート80の通信の許可が必要です。

```
sudo firewall-cmd --add-service=http
sudo firewall-cmd --add-service=https
sudo firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8443
sudo firewall-cmd --runtime-to-permanent
sudo firewall-cmd --reload
sudo firewall-cmd --list-all
```

```
ubuntu@mywhisper2:~$ sudo firewall-cmd --add-service=http
success
ubuntu@mywhisper2:~$ sudo firewall-cmd --add-service=https
success
ubuntu@mywhisper2:~$ sudo firewall-cmd --add-forward-
port=port=443:proto=tcp:toport=8443
success
ubuntu@mywhisper2:~$ sudo firewall-cmd --runtime-to-permanent
success
ubuntu@mywhisper2:~$ sudo firewall-cmd --reload
success
ubuntu@mywhisper2:~$ sudo firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client http https ssh
  ports:
  protocols:
  masquerade: no
  forward-ports: port=443:proto=tcp:toport=8443:toaddr=
  source-ports:
  icmp-blocks:
  rich rules:
```

ubuntu@mywhisper2:~\$

Certbotをスタンドアロンで実行し、サーバー証明書を取得します。

```
sudo certbot certonly --standalone
```

```
ubuntu@mywhisper2:~$ sudo certbot certonly --standalone
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning:
urllib3 (1.26.13) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported version")
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): 申請者のメール・アドレス
```

```
- - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
```

agree in order to register with the ACME server at
<https://acme-v02.api.letsencrypt.org/directory>

(A)gree/(C)ancel: **A**

Would you be willing to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web, EFF news, campaigns, and ways to support digital freedom.

(Y)es/(N)o: **N**

Please enter in your domain name(s) (comma and/or space separated) (Enter 'c' to cancel): **サーバー証明書を発行するホスト名**

Obtaining a new certificate

Performing the following challenges:

http-01 challenge for **ホスト名**

Waiting for verification...

Cleaning up challenges

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:

/etc/letsencrypt/live/ホスト名/fullchain.pem

Your key file has been saved at:

/etc/letsencrypt/live/ホスト名/privkey.pem

Your cert will expire on 2023-04-23. To obtain a new or tweaked version of this certificate in the future, simply run certbot again. To non-interactively renew *all* of your certificates, run "certbot renew"

- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>

Donating to EFF: <https://eff.org/donate-le>

ubuntu@mywhisper2:~\$

サーバー証明書のファイルが**/etc/letsencrypt/live/ホスト名/fullchain.pem**、秘密キーのファイルが**/etc/letsencrypt/live/ホスト名/privkey.pem**として作成されます。

これらを**\$HOME/certs**以下にコピーし、ユーザーubuntuより読み取れるように設定します。

```
sudo cp /etc/letsencrypt/live/ホスト名/privkey.pem certs
```

```
sudo cp /etc/letsencrypt/live/ホスト名/fullchain.pem certs
```

```
sudo chown ubuntu certs/*
```

```
sudo chmod 400 certs/*
```

```
ubuntu@mywhisper2:~$ sudo cp /etc/letsencrypt/live/ホスト名/privkey.pem certs
```

```
ubuntu@mywhisper2:~$ sudo cp /etc/letsencrypt/live/ホスト名/fullchain.pem certs
```

```
ubuntu@mywhisper2:~$ ls certs
```

```
fullchain.pem  privkey.pem
```

```
ubuntu@mywhisper2:~$ sudo chown ubuntu certs/*
```

```
ubuntu@mywhisper2:~$ sudo chmod 400 certs/*
```

ファイル**whisper-server.py**を作成します。内容は以下になります。タイムアウトの発生を避けるため、モデルとして**small**を選択しています。

```
import os
from flask import Flask, request
```

```

import uuid
import whisper
model = whisper.load_model("small")

app = Flask(__name__)
@app.route('/transcribe', methods=['POST'])
def transcribe():
    if request.method == 'POST':
        if 'file' not in request.files:
            return 'No file part'

        tempfile = os.path.join('./audio', str(uuid.uuid1()))
        file = request.files['file']
        file.save(tempfile)
        result = model.transcribe(tempfile)
        os.remove(tempfile)

    return result

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8443, ssl_context=('./certs/fullchain.pem', './certs/privkey.p

```

whisper-server.py hosted with ❤ by GitHub

[view raw](#)

作成した**whisper-server.py**を実行します。

python whisper-server.py

```

ubuntu@mywhisper2:~$ python whisper-server.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning:
urllib3 (1.26.13) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
* Serving Flask app 'whisper-server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on https://127.0.0.1:8443
* Running on https://10.0.0.131:8443
Press CTRL+C to quit
* Restarting with stat
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning:
urllib3 (1.26.13) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
* Debugger is active!
* Debugger PIN: 197-750-656

```

以上で、APIの呼び出しによる文字起こしができるようになりました。

curlコマンドと音声ファイルを使って、APIを呼び出してみます。curlを実行するディレクトリに音声ファイルとしてtest.m4aが作成済みとします。

curl -X POST -F 'file=@test.m4a' https://ホスト名/transcribe

```
% curl -X POST -F 'file=@test.m4a' https://ホスト名/transcribe
{
  "language": "ja",
  "segments": [
    {
      "avg_logprob": -0.39915904131802643,
      "compression_ratio": 1.065217391304348,
      "end": 10.6,
      "id": 0,
      "no_speech_prob": 0.02103608287870884,
      "seek": 0,
      "start": 0.0,
      "temperature": 0.0,
      "text":
"\u3053\u3093\u306b\u3061\u306f\u521d\u3081\u3066Whipper\u3092\u30a4\u30f3\u30b9\u30c8\u30fc\u30eb\u3057\u3066\u307f\u307e\u3057\u305f\u3053\u308c\u3067\u8a66\u3057\u3066\u307f\u307e\u3059",
      "tokens": [
        50364,
        38088,
        28727,
        38975,
        2471,
        15124,
        5998,
        8040,
        4824,
        40498,
        35307,
        8822,
        11362,
        12072,
        33732,
        2474,
        22099,
        8822,
        11362,
        5368,
        50894
      ]
    }
  ],
  "text":
"\u3053\u3093\u306b\u3061\u306f\u521d\u3081\u3066Whipper\u3092\u30a4\u30f3\u30b9\u30c8\u30fc\u30eb\u3057\u3066\u307f\u307e\u3057\u305f\u3053\u308c\u3067\u8a66\u3057\u3066\u307f\u307e\u3059"
}
%
```

次の記事では、APEXアプリケーションより文字起こしをするAPIを呼び出してみます。

続く

Yuji N. 時刻: 11:19

共有

[ウェブ バージョンを表示](#)

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

Powered by [Blogger](#).
