Deployment method of Face recognition application

1 Load a trained face recognition model and y label

In [4]:
```
import pickle

def load_model():
    global trained_model, y
    #load trained model
    model_path = 'face_model_file'
    trained_model =  pickle.load(open(model_path, 'rb'))
    y_path = 'y_list.txt'
    with open(y_path, 'r') as filehandle:
        y = [line.rstrip() for line in filehandle.readlines()]

load_model()
```

2 Define a 'predict' endpoint

In [ ]:
```
@app.route('/predict', methods=['POST'])
def get_prediction():
    # Works only for a single sample
    if request.method == 'POST':
        data = request.get_json()  # Get data posted as a json
        data = np.array(data)[np.newaxis, :]  # converts shape from
        prediction = trained_model.predict(data)  # runs globally lo
    return str(prediction[0])
```

3 Use app.py to wrap the inference logic in a flask server to serve the model as a REST webservice

In [ ]:

```
# Serve model as a flask application

import pickle
import numpy as np
from flask import Flask, request

model = None
app = Flask(__name__)


def load_model():
    global trained_model, y
    #load trained model
    model_path = 'face_model_file'
    trained_model =  pickle.load(open(model_path, 'rb'))
    y_path = 'y_list.txt'
    with open(y_path, 'r') as filehandle:
        y = [line.rstrip() for line in filehandle.readlines()]

@app.route('/')
def home_endpoint():
    return 'Hello World!'


@app.route('/predict', methods=['POST'])
def get_prediction():
    # Works only for a single sample
    if request.method == 'POST':
        data = request.get_json()  # Get data posted as a json
        data = np.array(data)[np.newaxis, :]  # converts shape from
        prediction = trained_model.predict(data)  # runs globally lo
    return str(prediction[0])


if __name__ == '__main__':
    load_model()  # load model at the beginning once only
    app.run(host='0.0.0.0', port=5000)
```

4 Setup docker

#Install Docker on Ubuntu 18.04

sudo apt install docker.io
#Start and Automate Docker. The Docker service needs to be setup to run at startup
 sudo systemctl start docker

$ sudo systemctl enable docker

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.

5 Create a docker image. This image contains executing environment of app.py

$ docker build -t app-facerecog .

In [ ]:

```
6 Execute docker image<br>
docker run -p 80:80 app-facerecog .
```