Ollama Open-Webui

Note: You should have at least 8 GB of RAM available to run the 7B models, 16 GB to run the 13B models, and 32 GB to run the 33B models.

Notes only for now:

https://github.com/open-webui/open-webui
https://docs.openwebui.com/getting-started/env-configuration/#general

Nvidia GPU, linux:

```
mkdir /opt/ollama
mkdir /opt/open-webui
mkdir /opt/openedai-speech/tts-voices
mkdir /opt/openedai-speech/tts-config
mkdir /opt/pipelines
mkdir /opt/docker-ssl-proxy
mkdir /opt/faster-whisper-server
```

docker-ollama.yml

```
name: ollama
services:
  ollama:
    image: ollama/ollama:latest
    container_name: ollama
    volumes:
      - /opt/ollama:/root/.ollama
    ports:
      - 11434:11434
    #runtime: nvidia
    restart: unless-stopped
    deploy:
      resources:
        reservations:
          devices:
          - driver: nvidia
            #device_ids: ['0']
            count: 1
            capabilities: [gpu]
```

docker-openwebui.yml

```
name: open-webui
services:
  open-webui:
    image: ghcr.io/open-webui/open-webui:main
```

```
    container_name: open-webui
    deploy:
      resources:
        reservations:
          devices:
          - driver: nvidia
            device_ids: ['0']
            capabilities: [gpu]
    ports:
      - 3000:8080
    volumes:
      - /opt/open-webui:/app/backend/data
    restart: unless-stopped
    extra_hosts:
      host.docker.internal: host-gateway
    environment:
      - WEBUI_NAME=CustomGPTName
      - TZ=Europe/London
      - RAG_EMBEDDING_MODEL_TRUST_REMOTE_CODE=True # allow
sentencetransformers to execute code like for alibaba-nlp/gte-large-en-
v1.5
```

docker-openedai-speech.yml

```
name: openedai-speech
services:
  openedai-speech:
    image: ghcr.io/matatonic/openedai-speech
    container_name: openedai-speech
    ports:
      - "8060:8000"
    volumes:
      - /opt/openedai-speech/tts-voices:/app/voices
      - /opt/openedai-speech/tts-config:/app/config
    restart: unless-stopped
    deploy:
      resources:
        reservations:
          devices:
          - driver: nvidia
            device_ids: ['0']
            capabilities: [gpu]
    environment:
      - TZ=Europe/London
      - TTS_HOME=voices
      - HF_HOME=voices
      # - PRELOAD_MODEL=xtts
      # - PRELOAD_MODEL=xtts_v2.0.2
      # - PRELOAD_MODEL=parler-tts/parler_tts_mini_v0.1
```

In open-webui under settings → audio

1. set to openai
2. API Base URL: http://host.docker.internal:8060/v1
3. API Key: sk-111111111 (note: this is a dummy API key, no key required)
4. Under TTS Voice within the same audio settings menu in the admin panel, you can set the TTS Model to use from the following choices below that openedai-speech supports. The voices of these models are optimized for the English language.
   tts-1 or tts-1-hd: alloy, echo, echo-alt, fable, onyx, nova, and shimmer (tts-1-hd is configurable; uses OpenAI samples by default)

docker-pipelines.yml

```
name: pipelines
services:
  pipelines:
    image: ghcr.io/open-webui/pipelines:main
    container_name: pipelines
    volumes:
      - /opt/pipelines:/app/pipelines
    ports:
      - 9099:9099
    restart: always
    extra_hosts:
      host.docker.internal: host-gateway
    deploy:
      resources:
        reservations:
          devices:
          - driver: nvidia
            device_ids: ['0']
            capabilities: [gpu]
```

https://zohaib.me/extending-openwebui-using-pipelines/

Under settings→connections set:

1. OPENAI API host: http://host.docker.internal:9099
2. OPENAI API Key: 0p3n-w3bu!

```
git clone https://github.com/fedirz/faster-whisper-server
```

docker-faster-whisper-server.yml

```
name: faster-whisper-server
services:
  faster-whisper-server-cuda:
    image: fedirz/faster-whisper-server:latest-cuda
    build:
      dockerfile: faster-whisper-server/Dockerfile.cuda
```

```
      context: ./faster-whisper-server
      platforms:
        - linux/amd64
    volumes:
      - /opt/faster-whisper-server/:/root/.cache/huggingface
    restart: unless-stopped
    ports:
      - 8010:8000
    develop:
      watch:
        - path: faster_whisper_server
          action: rebuild
    deploy:
      resources:
        reservations:
          devices:
            - capabilities: ["gpu"]
```

go to settings → audio, set

1. OPENAI API host: http://host.docker.internal:8010/v1
2. OPENAI API Key: sk-something
3. model: whisper-1

NOTE: speech to text requires https connection to open-webui as browsers do not have access to microphone on http connection!

```
mkdir /opt/docker-ssl-proxy/
cd /opt/docker-ssl-proxy/
openssl req -subj '/CN=hostname.example.com' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

/opt/docker-ssl-proxy/proxy_ssl.conf

```
server {
  listen 80;
  server_name _;
  return 301 https://$host$request_uri;
}
server {
  listen 443 ssl;
  ssl_certificate /etc/nginx/conf.d/cert.pem;
  ssl_certificate_key /etc/nginx/conf.d/key.pem;
  location / {
     proxy_pass http://host.docker.internal:3000;
  }
}
```

docker-ssl-proxy.yml

```
name: nginx-proxy
services:
  nginx-proxy:
    image: nginx
    container_name: nginx-proxy
    ports:
      - 80:80
      - 443:443
    volumes:
      - /opt/docker-ssl-proxy:/etc/nginx/conf.d
    restart: unless-stopped
    extra_hosts:
      host.docker.internal: host-gateway
    environment:
      - TZ=Europe/London
```

To pull an ollama image, better to use ollama directly as the webinterface doesn't handle stalls well:

```
docker exec -ti ollama ollama pull imagename:tag
```

To update all previously pulled ollama models, use this bash script:

update-ollama-models.sh

```
#!/bin/bash

docker exec -ti ollama ollama list | tail -n +2 | awk '{print $1}' |
while read -r model; do
  echo "Updating model: $model..."
  docker exec -t ollama ollama pull $model
  echo "--"
done
echo "All models updated."
```

AMD GPU on Windows:

docker-ollama.yml

```
name: ollama
services:
  ollama:
    image: ollama/ollama:rocm
    container_name: ollama
    volumes:
      - /p/Docker_Volumes/ollama:/root/.ollama
    ports:
      - 11434:11434
    deploy:
```

```
      resources:
        reservations:
          devices:
            - capabilities: [gpu]
```

docker-openwebui.yml

```
name: webui
services:
  open-webui:
    image: ghcr.io/open-webui/open-webui:main
    container_name: open-webui
    deploy:
      resources:
        reservations:
          devices:
          - capabilities: [gpu]
    ports:
      - 3000:8080
    volumes:
      - /p/Docker_Volumes/openwebui:/app/backend/data
    restart: always
    extra_hosts:
      host.docker.internal: host-gateway
    environment:
      - WEBUI_AUTH=false
```

Create the respective docker volumes folder:

```
# p/Docker_Volumes = P:\Docker_Volumes
mkdir P:\Docker_Volumes
```

docker install - choose the WSL2 backend # cmd line

```
docker compose -f docker-openwebui.yml up -d
docker compose -f docker-ollama.yml up -d
```

to update all ollama models on windows, use this powershell command - adjust for the hostname/ip ollama is running on:

```
(Invoke-RestMethod http://localhost:11434/api/tags).Models.Name.ForEach{
ollama pull $_ }

#or if in docker
(Invoke-RestMethod http://localhost:11434/api/tags).Models.Name.ForEach{
docker exex -t ollama ollama pull $_ }
```

Curl OpenAI API test

```
curl http://localhost:11434/v1/chat/completions \
    -H "Content-Type: application/json" \
    -d '{
        "model": "llama3",
        "messages": [
            {
                "role": "system",
                "content": "You are a helpful assistant."
            },
            {
                "role": "user",
                "content": "Hello!"
            }
        ]
    }'
{"id":"chatcmpl-957","object":"chat.completion","created":1722601457,"model"
:"llama3","system_fingerprint":"fp_ollama","choices":[{"index":0,"message":{
"role":"assistant","content":"Hi there! It's great to meet you! I'm here to
help with any questions or tasks you might have. What brings you to this
virtual space today? Are you looking for recommendations, seeking answers to
a specific question, or maybe looking for some inspiration? Let me know, and
I'll do my best to assist
you."},"finish_reason":"stop"}],"usage":{"prompt_tokens":23,"completion_toke
ns":68,"total_tokens":91}}
```