Mediac

Media server - Serving Video, Audio, Image, Document.

How to install

You need create 3 files: docker-compose.yaml, settings.ini, .env file.
Put these file into a folder.
Open with cmd then type `docker compose up -d`.

docker-compose.yaml

```
services:
  mediac-postgres:
    image: postgres:16-alpine
    container_name: mediac-postgres
    restart: unless-stopped
    env_file:
      - .env
    volumes:
      - ./data:/var/lib/postgresql/data
    networks:
      - shared-network
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U $${POSTGRES_USER} -d $${POSTGRES_DB}"]
      interval: 5s
      timeout: 5s
      retries: 10

  mediac:
    image: sandichhuu/mediac:nvidia
    container_name: mediac
    restart: unless-stopped
    user: "1000:1000"
    env_file:
      - .env
    ports:
      - "80:80"
    depends_on:
      mediac-postgres:
        condition: service_healthy
    networks:
      - shared-network
    environment:
      - NVIDIA_VISIBLE_DEVICES=all
      - NVIDIA_DRIVER_CAPABILITIES=compute,video,utility
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
```

```
              count: all
              capabilities: [gpu, compute, video, utility]
    volumes:
      - ./storage:/app/storage
      - ./settings.ini:/app/settings.ini:ro
networks:
  shared-network:
    external: true
```

.env

```
# PostgreSQL container
POSTGRES_USER=postgres
POSTGRES_PASSWORD={TYPE_RANDOM_PASSWORD}
POSTGRES_DB=uwu_videoplayer

# Backend database connection
DB_ENDPOINT=mediac-postgres
DB_PORT=5432
DB_USERNAME=postgres
DB_PASSWORD={TYPE_RANDOM_PASSWORD}
DB_NAME=uwu_videoplayer
DB_SSLMODE=disable

# Authentication
JWT_SECRET={RANDOM_32_DIGIT}
SERVER_SECRET={RANDOM_32_DIGIT}

# Initial administrator
ADMIN_USERNAME=admin
ADMIN_PASSWORD=admin

# Backend storage
STORAGE_DIR=/app/storage

# Video/transcoding
VIDEO_ENCODE_ENABLED=true
TEMP_TRANSCODE_TTL=30m
TEMP_TRANSCODE_TARGET=h264

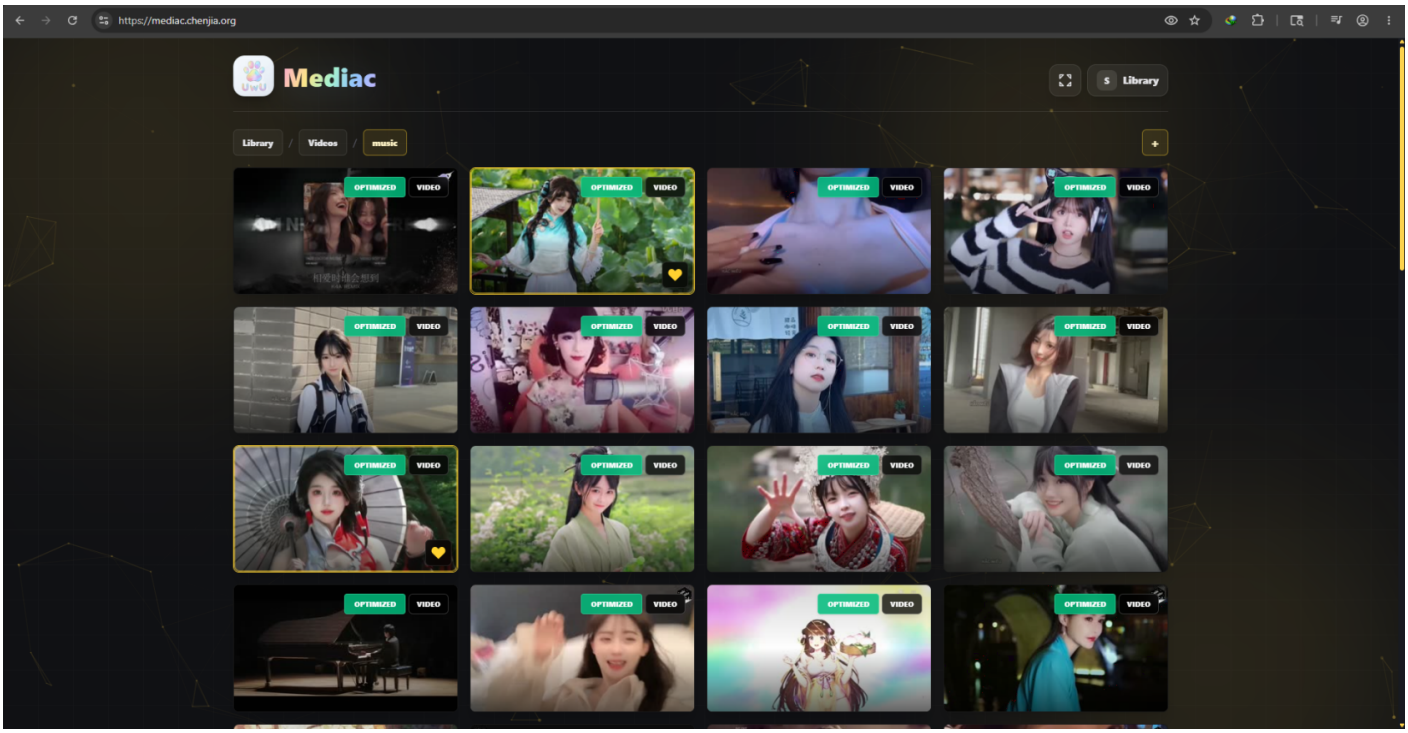
# Streaming
BUFFER_SIZE=2048
TRANSCODE_GOP_SIZE=30
SECURE_STREAM=0
```

settings.ini

```
FFMPEG_LIST_ENCODERS = ffmpeg -hide_banner -encoders
FFMPEG_LIST_DECODERS = ffmpeg -hide_banner -decoders
FFMPEG_PROBE_NVENC = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v h264_nvenc -f null -
FFMPEG_PROBE_AV1_NVENC = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v av1_nvenc -f null -
FFMPEG_PROBE_AV1_QSV = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v av1_qsv -f null -
FFMPEG_PROBE_QSV = ffmpeg -hide_banner -loglevel error -f lavfi -i
```

```
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v h264_qsv -f null -
FFMPEG_PROBE_AMF = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v h264_amf -f null -
FFMPEG_PROBE_AV1_AMF = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v av1_amf -f null -
FFMPEG_PROBE_VAAPI = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -vf format=nv12,hwupload -c:v
vp9_vaapi -f null -
FFMPEG_PROBE_VP9_QSV = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v vp9_qsv -f null -
FFMPEG_PROBE_H265_NVENC = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v hevc_nvenc -f null -
FFMPEG_PROBE_H265_QSV = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v hevc_qsv -f null -
FFMPEG_PROBE_H265_AMF = ffmpeg -hide_banner -loglevel error -f lavfi -i
testsrc2=size=320x240:rate=1:duration=1 -frames:v 1 -an -c:v hevc_amf -f null -
TRANSCODE_H264_QSV = ffmpeg -init_hw_device qsv=hw -filter_hw_device hw -hwaccel qsv -
hwaccel_output_format qsv -i {INPUT} -vf "hwupload=derive_device=qsv" -c:v h264_qsv -preset 1
-b:v {BITRATE} -g {GOP_SIZE} -forced_idr 1 -c:a aac -movflags +faststart {OUTPUT}
TRANSCODE_H264_CUDA = ffmpeg -hwaccel cuda -hwaccel_output_format cuda -i {INPUT} -c:v
h264_nvenc -preset p7 -b:v {BITRATE} -g {GOP_SIZE} -forced-idr 1 -no-scenecut 1 -c:a aac -
movflags +faststart {OUTPUT}
TRANSCODE_H264_AMF = ffmpeg -i {INPUT} -c:v h264_amf -quality quality -b:v {BITRATE} -g
{GOP_SIZE} -sc_threshold 0 -c:a aac -movflags +faststart {OUTPUT}
TRANSCODE_H264_CPU = ffmpeg -i {INPUT} -c:v libx264 -preset slow -b:v {BITRATE} -g {GOP_SIZE}
-keyint_min {GOP_SIZE} -sc_threshold 0 -c:a aac -movflags +faststart {OUTPUT}
TRANSCODE_H265_QSV = ffmpeg -i {INPUT} -c:v hevc_qsv -preset slow -b:v {BITRATE} -g {GOP_SIZE}
-forced_idr 1 -c:a aac -tag:v hvc1 -movflags +faststart {OUTPUT}
TRANSCODE_H265_CUDA = ffmpeg -hwaccel cuda -hwaccel_output_format cuda -i {INPUT} -c:v
hevc_nvenc -preset p7 -b:v {BITRATE} -g {GOP_SIZE} -forced-idr 1 -no-scenecut 1 -c:a aac -
tag:v hvc1 -movflags +faststart {OUTPUT}
TRANSCODE_H265_AMF = ffmpeg -i {INPUT} -c:v hevc_amf -quality quality -b:v {BITRATE} -g
{GOP_SIZE} -sc_threshold 0 -c:a aac -tag:v hvc1 -movflags +faststart {OUTPUT}
TRANSCODE_H265_CPU = ffmpeg -i {INPUT} -c:v libx265 -preset slow -b:v {BITRATE} -g {GOP_SIZE}
-sc_threshold 0 -c:a aac -tag:v hvc1 -movflags +faststart {OUTPUT}
TRANSCODE_VP9_QSV = ffmpeg -i {INPUT} -c:v vp9_qsv -b:v {BITRATE} -maxrate {BITRATE} -g
{GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
TRANSCODE_VP9_VAAPI = ffmpeg -i {INPUT} -vf "format=nv12,hwupload" -c:v vp9_vaapi -b:v
{BITRATE} -maxrate {BITRATE} -g {GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -
f webm {OUTPUT}
TRANSCODE_VP9_CPU = ffmpeg -i {INPUT} -c:v libvpx-vp9 -row-mt 1 -b:v {BITRATE} -maxrate
{BITRATE} -g {GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
TRANSCODE_AV1_QSV = ffmpeg -i {INPUT} -c:v av1_qsv -preset slow -b:v {BITRATE} -maxrate
{BITRATE} -g {GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
TRANSCODE_AV1_CUDA = ffmpeg -hwaccel cuda -hwaccel_output_format cuda -i {INPUT} -c:v
av1_nvenc -preset p7 -b:v {BITRATE} -maxrate {BITRATE} -g {GOP_SIZE} -forced-idr 1 -no-
scenecut 1 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
TRANSCODE_AV1_AMF = ffmpeg -i {INPUT} -c:v av1_amf -quality quality -b:v {BITRATE} -maxrate
{BITRATE} -g {GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
TRANSCODE_AV1_CPU = ffmpeg -i {INPUT} -c:v libaom-av1 -cpu-used 4 -b:v {BITRATE} -maxrate
{BITRATE} -g {GOP_SIZE} -sc_threshold 0 -c:a libopus -b:a {AUDIO_BITRATE} -f webm {OUTPUT}
```

Screenshot



Revision #3
Created 2026-09-09 02:30:30 UTC by sandichhuu
Updated 2026-09-09 02:41:00 UTC by sandichhuu