?? Products

- ⚛ llama.cpp
- ⚛ GAN-MASTER
- ⚛ ComfyUV
- □□ Ramen

?? llama.cpp

Run high-performance GGUF inference with CUDA 13.2 using llama.cpp and the Docker image `sandichhuu/llama-cpp:cu132`.

This setup is optimized for:

- Large GGUF models
- Long context inference
- Multi-modal models with `mmproj`
- NVIDIA GPU acceleration
- Speculative decoding (`draft-mtp`)
- High-throughput inference workloads

Features

- CUDA 13.2 support
- Prebuilt `llama-server`
- GGUF + multimodal support
- Optimized KV cache quantization
- Flash Attention enabled
- Long context support (`262144`)
- NVIDIA Container Toolkit compatible
- Ready for Docker Compose deployment

Requirements

Before starting, make sure you have:

- Docker
- Docker Compose
- NVIDIA Driver installed
- NVIDIA Container Toolkit

Verify GPU access:

```
docker run --rm --gpus all nvidia/cuda:13.2.0-base-ubuntu24.04 nvidia-smi
```

Docker Compose Example

```
services:
  llama-cpp:
    image: sandichhuu/llama-cpp:cu132
    container_name: llama-cpp

    ports:
      - 8080:8080

    environment:
      - GGML_CUDA_GRAPH_OPT=1
      - LD_LIBRARY_PATH=/usr/local/nvidia/lib64:/usr/local/nvidia/lib:/usr/lib/x86_64-linux-
gnu

    privileged: true
    ipc: host

    volumes:
      - ./models:/app/models

    command: >
      -m
      /app/models/unsloth/Qwen3.6-35B-A3B-MTP-GGUF/Qwen3.6-35B-A3B-UD-IQ4_NL.gguf
      --mmproj /app/models/unsloth/Qwen3.6-35B-A3B-MTP-GGUF/mmproj-BF16.gguf
      --port 8080
      --host 0.0.0.0
      -t 12
      -c 262144
      --parallel 1
      -fa on
      --cache-type-k q4_0
      --cache-type-v q4_0
      --n-cpu-moe 35
      --no-context-shift
      -b 4096
      -ub 2048
      --no-mmap
      --direct-io
      --fit off
      --jinja
      --no-cache-prompt
      --cache-ram 0
      --spec-type draft-mtp
      --spec-draft-n-max 2

    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities:
                - gpu

    restart: unless-stopped
```

Folder Structure

```
.
??? docker-compose.yml
??? models
    ??? unsloth
        ??? Qwen3.6-35B-A3B-MTP-GGUF
```

Start the Container

```
docker compose up -d
```

Check logs:

```
docker logs -f llama-cpp
```

OpenAI-Compatible API

The server exposes an OpenAI-compatible API on:

```
http://localhost:8080
```

Example request:

```
curl http://localhost:8080/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "Qwen",
    "messages": [
      {
        "role": "user",
        "content": "Hello"
      }
    ]
  }'
```

Explanation of Important Flags

Flag	Description
`-fa on`	Enables Flash Attention
`-c 262144`	Sets context length to 262k

Flag	Description
`--cache-type-k q4_0`	Quantized KV cache for lower VRAM usage
`--cache-type-v q4_0`	Quantized V cache
`--no-mmap`	Fully loads model into memory
`--direct-io`	Reduces page cache overhead
`--parallel 1`	Single inference stream
`--spec-type draft-mtp`	Enables speculative decoding
`--spec-draft-n-max 2`	Number of speculative draft tokens
`--mmproj`	Loads multimodal projection model
`--jinja`	Enables chat template rendering

Performance Notes

This configuration is tuned for large-scale inference workloads:

- Better throughput on large context windows
- Reduced VRAM usage using quantized KV cache
- Lower latency with CUDA graph optimization
- Improved token generation using speculative decoding
- Optimized for high-end NVIDIA GPUs

Recommended GPUs:

- RTX 4090
- RTX 5090
- A100
- H100
- L40S

Model Sources

You can download GGUF models from:

- https://huggingface.co/models?library=gguf
- https://huggingface.co/unsloth

Docker Image

Docker Hub:

Link DockerHub

Pull manually:

```
docker pull sandichhuu/llama-cpp:cu132
```

?? GAN-MASTER

An optimized AI tool support image restoration, image denoise, upscale, ... using GFP-GAN, GPEN, ...

> ⚠ **Important Note**
> □□ I only have RTX 30 series (SM86). So newer series untested.
> □□ If you facing any issue, please send the error log to:
>
> - □□ LinkedIn
> - □□ Facebook
> - □□ Whatsapp

□□ **Compatible GPU**	
□ **Newer**	Runnable, but slower.
□ **SM86**	RTX 30 Series.
□ **Unsupported**	RTX 20 Series and older, AMD or Intel GPU.

? Quick Start (Docker Compose)

Install **Docker**, **Newest Nvidia Driver & CUDA toolkit**
Next, copy **docker-compose.yml** content below and put to empty folder.
Then, open **Terminal/CMD/Windows PowerShell** and navigate to directory contain `docker-compose.yml` file.
Finally, run this command `docker compose up -d` to start the container. Go to `http://localhost:7860` and enjoy.

> ## Requirements
>
> ⬇ Docker
> ⬇ nvidia-driver
> ⬇ cuda-toolkit

> ## Models (download then copy into models folder
>
> ⬇ RetinaFace-R50
>
> ⬇ ParseNet-latest
>
> ⬇ realesrnet_x4
>
> ⬇ GPEN-BFR-512
>
> ⬇ Colorization
>
> ⬇ RealESRGAN_x2plus
>
> ⬇ GFPGANv1.4

docker-compose.yml

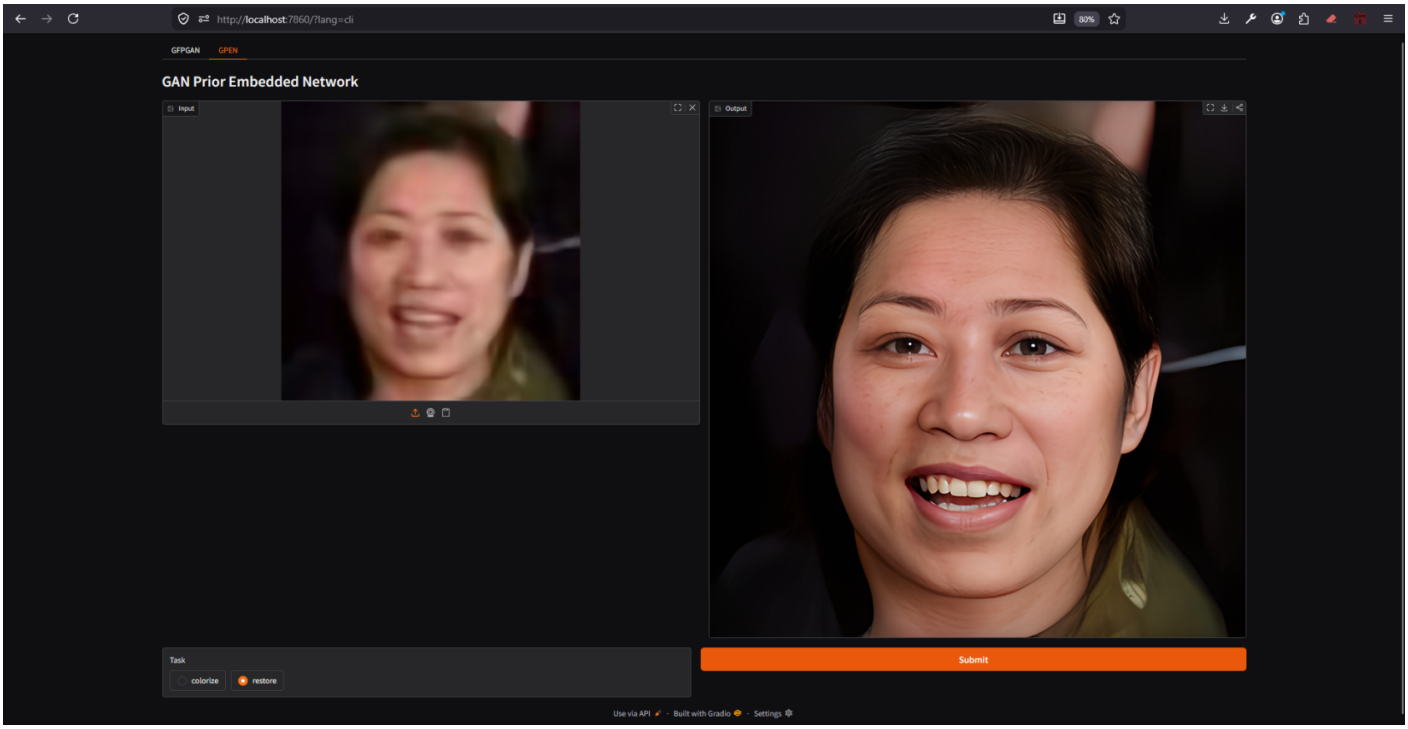
```
services:
  gan-master:
    image: sandichhuu/gan-master:sm86cu130
    container_name: gan-master
    ipc: host
    port:
      - 7860:7860
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]
    volumes:
      - ./models:/workspace/app/models
      - ./outputs:/workspace/app/outputs
```

? Setup Structure

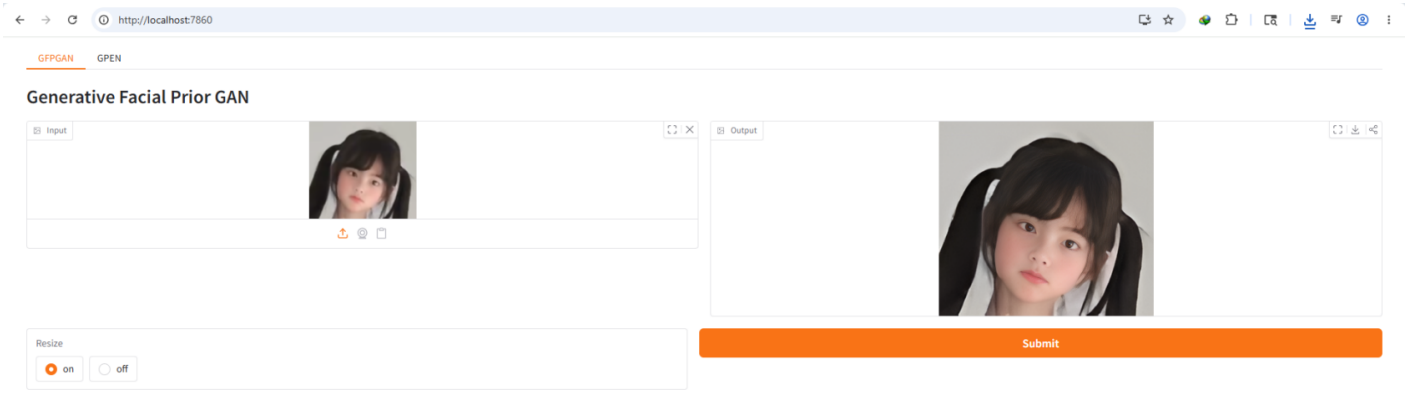
```
comfyuv/
??? docker-compose.yml
??? models/
??? output/
```

? Screenshot

GPEN (Restore Mode)



GFPGAN (Upscale Mode)



?? ComfyUV

This guide explains how to deploy the Docker image:

[sandichhuu/comfyuv on Docker Hub]

An optimized, production-ready ComfyUI Docker image built with `uv` for ultra-fast Python package management. It comes pre-configured with modern performance optimizations like Triton, SageAttention.

> ⚠ **Important Note**
> □□ I only have RTX 20, 30 series (SM75 & SM86). So newer series untested.
> □□ If you facing any issue, please send the error log to:
>
> □□ LinkedIn
>
> □□ Facebook
>
> □□ Whatsapp

□□ **Compatible GPU**	
□ **Newer**	Runnable, but slower.
□ **SM120**	RTX 50 Series.
□ **SM89**	RTX 40 Series.
□ **SM86**	RTX 30 Series.
□ **SM75**	RTX 20 Series.
□ **Unsupported**	GTX 10 Series and older, AMD or Intel GPU.

> ⚠ Unsupport **xformers**.
> Because cu132 have no pre-build for this package & have no reason to use the slowest method.
> □ You can use sage-attention or triton instead.

> ⚠ No longer support **flash-attn** (because the compile time is super slow).
> You can do it your self with this command:
>
> ```
> MAX_JOBS=10 NVCC_THREADS=1 uv pip install flash-attn --no-build-isolation
> ```

? Why use this Docker image for ComfyUI?

Nodes Compatibility	Python 3.13.13, widely compatible with extensions.
Blazing Fast Speed	Uses `uv` and **CUDA 13.2** for ultra-fast compilation and execution.
Precompiled Nodes	Comes with precompiled ecosystem, saving your setup and compile time.
Strict Privacy	Completely stripped of telemetry. Your data and workflows remain private.
Cloud Optimized	Ready for public deployment. Use caddy or nginx for authentication.

? PRE-INSTALLED Custom Nodes

▫ calcuis/gguf	(calcuis/gguf)
▫ ComfyUI-Crystools-MonitorOnly	(BobRandomNumber)
▫ ComfyUI-VideoHelperSuite	(Kosinkadink)
▫ rgthree-comfy	(rgthree)
▫ ComfyUI-Lora-Manager	(willmiao)
▫ Nvidia_RTX_Nodes_ComfyUI	(Comfy-Org)
▫ ComfyUI-KJNodes	(kijai)

? Quick Start (Docker Compose)

Install **Docker**, **Nvidia Driver** (refer Studio version), **CUDA Toolkit** (version 13.2).
Next, copy **docker-compose.yml** content below and put to empty folder.
Then, open **Terminal/CMD/Windows PowerShell** and navigate to directory contain `docker-compose.yml` file.
Finally, run this command `docker compose up -d` to start the container. Go to `http://localhost:8188` and enjoy.

> ## Requirements
>
> ⬇ Docker
>
> ⬇ nvidia-driver
>
> ⬇ cuda-toolkit (CUDA 13.2)

docker-compose.yml

```
services:
  comfyuv:
    image: sandichhuu/comfyuv:sm86cu132
    container_name: comfyuv
    ipc: host
    port:
      - 8188:8188
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]
    volumes:
      - ./input:/comfy/input
      - ./output:/comfy/output
      - ./workflows:/comfy/user/default/workflows
      - ./models:/comfy/models
      - ./workflows:/comfy/user/default/workflows

      - comfyuv:/comfy
      - uv_cache:/root/.cache/uv
    command: >
      --enable-triton-backend
#     --use-flash-attention unsupport becase it take long times to compile.
#     --lowvram
      --use-sage-attention
      --disable-pinned-memory
      --fast fp8_matrix_mult autotune
      --front-end-version Comfy-Org/ComfyUI_frontend@latest

volumes:
  comfyuv:
  uv_cache:
```

? Setup Structure

```
comfyuv/
??? docker-compose.yml
??? input/
??? models/
??? workflows/
??? output/
```

? Why have no all-in-one image ?

All GPU support int4, int8, fp16, fp32.
But each GPU series has different physics structure due to different kernels and different attention optimization.

For example RTX4090 support fp8 native, RTX5090 support fp8 & nvfp4 native.
RTX40xx fastest with FlashAttention3, RTX50xx with FlashAttention4.
If I compile everything on one image, the file size come super heavy.
And the compile time is slow as hell.

References

- ComfyUI Official Docs
- Docker Hub Image
- Docker Hub Documentation

? Ramen

More easy UI for ComfyUI. Auto release VRAM / RAM, ensure stability and UX when generate Image / Video ...

> ⚠ **Important Note**
> □□ I only have RTX 20, 30 series (SM75 & SM86). So newer series untested.
> □□ If you facing any issue, please send the error log to:
>
> - □□ LinkedIn
> - □□ Facebook
> - □□ Whatsapp

□□ **Compatible GPU**	
□ **Newer**	Runnable, but slower.
□ **SM120**	RTX 50 Series.
□ **SM89**	RTX 40 Series.
□ **SM86**	RTX 30 Series.
□ **Unsupported**	RTX 20 Series and older, AMD or Intel GPU.

> ⚠ Unsupport **xformers**.
> Because cu132 have no pre-build for this package & have no reason to use the slowest method.
> □ You can use sage-attention or triton instead.

> ⚠ No longer support **flash-attn** (because the compile time is super slow).
> □ You can do it your self with this command:
>
> ```
> MAX_JOBS=10 NVCC_THREADS=1 uv pip install flash-attn --no-build-isolation
> ```

? Why use this Docker image ?

Nodes Compatibility	Python 3.13.13, widely compatible with extensions.
Blazing Fast Speed	Uses `uv` and **CUDA 13.2** for ultra-fast compilation and execution.
Precompiled Nodes	Comes with precompiled ecosystem, saving your setup and compile time.
Strict Privacy	Completely stripped of telemetry. Your data and workflows remain private.
Cloud Optimized	Ready for public deployment. Use caddy or nginx for authentication.

? Quick Start (Docker Compose)

Install **Docker**, **Nvidia Driver** (refer Studio version), **CUDA Toolkit** (version 13.2).
Next, copy **docker-compose.yml** content below and put to empty folder.
Then, open **Terminal/CMD/Windows PowerShell** and navigate to directory contain `docker-compose.yml` file.
Finally, run this command `docker compose up -d` to start the container. Go to `http://localhost:7860` and enjoy.

> “
>
> ## Requirements
>
> ⬇ Docker
>
> ⬇ nvidia-driver
>
> ⬇ cuda-toolkit (CUDA 13.2)

Models

> ⚠ **Important Note**
> □□ The path must be correct □□

>

> - LTX2.3 from Kijai:
>   **models/diffusion_models/ltx-2.3-22b-distilled_transformer_only_fp8_scaled.safetensors**: download
>   **models/text_encoders/ltx-2.3_text_projection_bf16.safetensors**: download
>   **model/vae/LTX23_audio_vae_bf16.safetensors**: download
>   **model/vae/LTX23_video_vae_bf16.safetensors**: download
>   **model/text_encoders/gemma_3_12B_it_fp4_mixed.safetensors**: download

> - Z-Image-Turbo from ComfyUI:
>   **models/diffusion_models/ZImageTurbo/z_image_turbo_bf16.safetensors**: download
>   **models/text_encoders/qwen_3_4b.safetensors"**: download
>   **models/vae/ae.safetensors**: download

> - Flux2-Klein-9B
>   **models/diffusion_models/Flux.2 Klein 9B/flux-2-klein-9b-fp8.safetensors**: download
>   **models/text_encoders/qwen_3_8b_fp8mixed.safetensors**: download
>   **models/vae/full_encoder_small_decoder.safetensors**: download
>   **models/loras/Flux.2 Klein 9B-base/KLEIN-Unchained-V2.safetensors**: download

docker-compose.yml

```
services:
  ramen:
    image: sandichhuu/ramen:sm86-rev1
    container_name: ramen
    ports:
      - 7860:7860
#    environment:
#      - GRADIO_ROOT_PATH=https://ramen.example.com
    volumes:
```

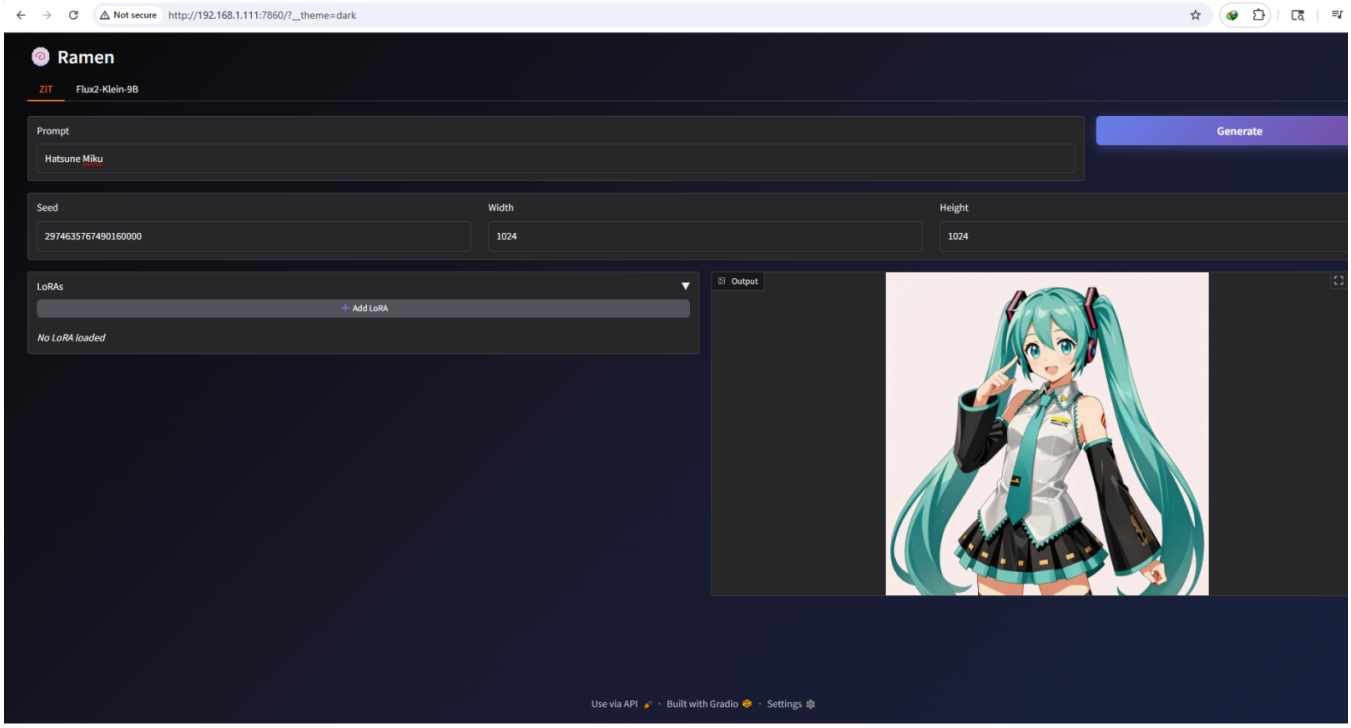
```
      - ./models:/comfy/models
      - comfy:/comfy
      - ramen:/ramen
    ipc: host
    restart: unless-stopped
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities:
                - gpu
volumes:
  comfy: null
  ramen: null
```

? Setup Structure

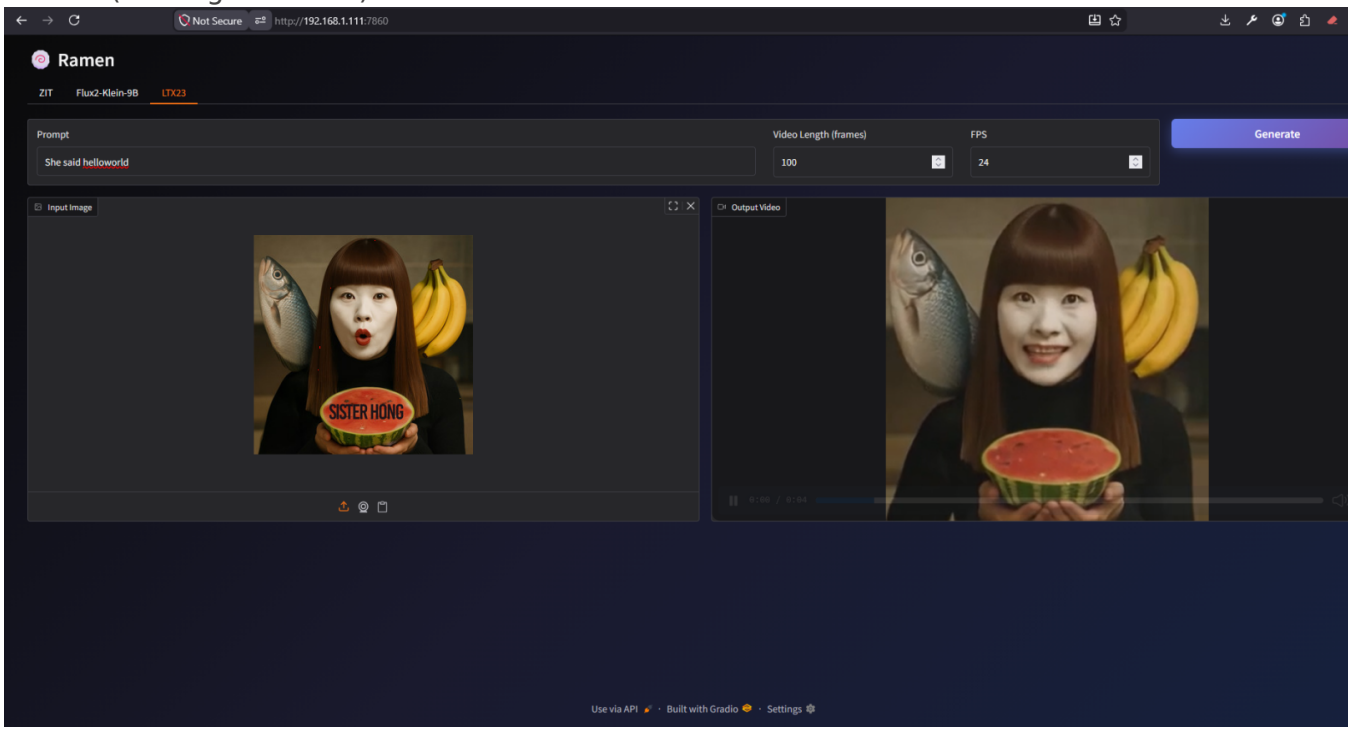
```
comfyuv/
??? docker-compose.yml
??? model/
```

? Screenshots

- Z-Image-Turbo (generate image)



- LTX2.3 (video generation)



? Why have no all-in-one image ?

All GPU support int4, int8, fp16, fp32.
But each GPU series has different physics structure due to different kernels and different attention optimization.

For example RTX4090 support fp8 native, RTX5090 support fp8 & nvfp4 native.
RTX40xx fastest with FlashAttention3, RTX50xx with FlashAttention4.
If I compile everything on one image, the file size come super heavy.
And the compile time is slow as hell.