readme.md for @host.today/ht-docker-ai

Production-ready Docker images for state-of-the-art AI Vision-Language Models. Run powerful multimodal AI locally with GPU acceleration—**no cloud API keys required**.

> □□ **Three VLMs, one registry.** From high-performance document OCR to GPT-4o-level vision understanding—pick the right tool for your task.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

□□ What's Included

Model	Parameters	Best For	API	Port	VRAM
MiniCPM-V 4.5	8B	General vision understanding, multi-image analysis	Ollama-compatible	11434	~9GB
Nanonets-OCR2-3B	~3B	Document OCR with semantic markdown, LaTeX, flowcharts	OpenAI-compatible	8000	~12-16GB
Qwen3-VL-30B	30B (A3B)	Advanced visual agents, code generation from images	Ollama-compatible	11434	~20GB

□□ Quick Reference: All Available Images

```
code.foss.global/host.today/ht-docker-ai:<tag>
```

Tag	Model	Runtime	Port	VRAM
`minicpm45v` / `latest`	MiniCPM-V 4.5	Ollama	11434	~9GB
`nanonets-ocr`	Nanonets-OCR2-3B	vLLM	8000	~12-16GB
`qwen3vl`	Qwen3-VL-30B-A3B	Ollama	11434	~20GB

□□ MiniCPM-V 4.5

A GPT-4o level multimodal LLM from OpenBMB—handles image understanding, OCR, multi-image analysis, and visual reasoning across **30+ languages**.

□ Key Features

- □□ **Multilingual:** 30+ languages supported
- □□ **Multi-image:** Analyze multiple images in one request
- □□ **Versatile:** Charts, documents, photos, diagrams
- ⚡ **Efficient:** Runs on consumer GPUs (9GB VRAM)

Quick Start

```
docker run -d \
  --name minicpm \
  --gpus all \
  -p 11434:11434 \
  -v ollama-data:/root/.ollama \
  code.foss.global/host.today/ht-docker-ai:minicpm45v
```

> □□ **Pro tip:** Mount the volume to persist downloaded models (~5GB). Without it, models re-download on every container start.

API Examples

List models:

```
curl http://localhost:11434/api/tags
```

Analyze an image:

```
curl http://localhost:11434/api/generate -d '{
  "model": "minicpm-v",
  "prompt": "What do you see in this image?",
  "images": ["<base64-encoded-image>"]
}'
```

Chat with vision:

```
curl http://localhost:11434/api/chat -d '{
  "model": "minicpm-v",
  "messages": [{
    "role": "user",
    "content": "Describe this image in detail",
    "images": ["<base64-encoded-image>"]
  }]
}'
```

Hardware Requirements

Mode	VRAM Required
int4 quantized	~9GB
Full precision (bf16)	~18GB

□□ Nanonets-OCR2-3B

The **latest Nanonets document OCR model** (October 2025 release)—based on Qwen2.5-VL-3B, fine-tuned specifically for document extraction with significant improvements over the original OCR-s.

□ Key Features

- □□ **Semantic output:** Tables → HTML, equations → LaTeX, flowcharts → structured markup
- □□ **Multilingual:** Inherits Qwen's broad language support
- □□ **30K context:** Handle large, multi-page documents
- □□ **OpenAI-compatible:** Drop-in replacement for existing pipelines
- □□ **Improved accuracy:** Better semantic tagging and LaTeX equation extraction vs. OCR-s

Quick Start

```
docker run -d \
  --name nanonets \
  --gpus all \
  -p 8000:8000 \
  -v hf-cache:/root/.cache/huggingface \
  code.foss.global/host.today/ht-docker-ai:nanonets-ocr
```

API Usage

```
curl http://localhost:8000/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "nanonets/Nanonets-OCR2-3B",
    "messages": [{
      "role": "user",
      "content": [
        {"type": "image_url", "image_url": {"url": "<base64>"}},
        {"type": "text", "text": "Extract the text from the above document as if you were reading it naturally. Return the tables in html format. Return the equations in LaTeX representation."}
      ]
    }],
```

```
    "temperature": 0.0,
    "max_tokens": 4096
  }'
```

Output Format

Nanonets-OCR2-3B returns markdown with semantic tags:

Element	Output Format
Tables	`<table>...</table>` (HTML)
Equations	`$...$` (LaTeX)
Images	`<img>description</img>`
Watermarks	`<watermark>OFFICIAL COPY</watermark>`
Page numbers	`<page_number>14</page_number>`
Flowcharts	Structured markup

Hardware Requirements

Config	VRAM
30K context (default)	~12-16GB
Speed	~3-8 seconds per page

□□ Qwen3-VL-30B-A3B

The **most powerful** Qwen vision model—30B parameters with 3B active (MoE architecture). Handles complex visual reasoning, code generation from screenshots, and visual agent capabilities.

□ Key Features

- □□ **256K context** (expandable to 1M tokens!)
- □□ **Visual agent capabilities** — can plan and execute multi-step tasks
- □□ **Code generation from images** — screenshot → working code
- □□ **State-of-the-art** visual reasoning

Quick Start

```
docker run -d \
  --name qwen3vl \
  --gpus all \
  -p 11434:11434 \
  -v ollama-data:/root/.ollama \
  code.foss.global/host.today/ht-docker-ai:qwen3vl
```

Then pull the model (one-time, ~20GB):

```
docker exec qwen3vl ollama pull qwen3-vl:30b-a3b
```

API Usage

```
curl http://localhost:11434/api/chat -d '{
  "model": "qwen3-vl:30b-a3b",
  "messages": [{
    "role": "user",
    "content": "Analyze this screenshot and write the code to recreate this UI",
    "images": ["<base64-encoded-image>"]
  }]
}'
```

Hardware Requirements

Requirement	Value
VRAM	~20GB (Q4_K_M quantization)
Context	256K tokens default

□□ Docker Compose

Run multiple VLMs together for maximum flexibility:

```yaml
services:
  # General vision tasks
  minicpm:
    image: code.foss.global/host.today/ht-docker-ai:minicpm45v
    ports:
      - "11434:11434"
    volumes:
      - ollama-data:/root/.ollama
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]
    restart: unless-stopped

  # Document OCR with semantic output
  nanonets:
    image: code.foss.global/host.today/ht-docker-ai:nanonets-ocr
    ports:
      - "8000:8000"
    volumes:
      - hf-cache:/root/.cache/huggingface
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]
    restart: unless-stopped

volumes:
  ollama-data:
  hf-cache:
```

⚙ Environment Variables

MiniCPM-V 4.5 & Qwen3-VL (Ollama-based)

Variable	Default	Description
`MODEL_NAME`	`minicpm-v`	Ollama model to pull on startup
`OLLAMA_HOST`	`0.0.0.0`	API bind address
`OLLAMA_ORIGINS`	`*`	Allowed CORS origins

Nanonets-OCR (vLLM-based)

Variable	Default	Description
`MODEL_NAME`	`nanonets/Nanonets-OCR2-3B`	HuggingFace model ID
`HOST`	`0.0.0.0`	API bind address
`PORT`	`8000`	API port
`MAX_MODEL_LEN`	`30000`	Maximum sequence length
`GPU_MEMORY_UTILIZATION`	`0.9`	GPU memory usage (0-1)

□□ Architecture Notes

Dual-VLM Consensus Strategy

For production document extraction, consider using multiple models together:

1. **Pass 1:** MiniCPM-V visual extraction (images → JSON)
2. **Pass 2:** Nanonets-OCR semantic extraction (images → markdown → JSON)
3. **Consensus:** If results match → Done (fast path)
4. **Pass 3+:** Additional visual passes if needed

This dual-VLM approach catches extraction errors that single models miss.

Why Multi-Model Works

- **Different architectures:** Independent models cross-validate each other
- **Specialized strengths:** Nanonets-OCR2-3B excels at document structure; MiniCPM-V handles general vision
- **Native processing:** All VLMs see original images—no intermediate structure loss

Model Selection Guide

Task	Recommended Model
General image understanding	MiniCPM-V 4.5
Document OCR with structure preservation	Nanonets-OCR2-3B
Complex visual reasoning / code generation	Qwen3-VL-30B
Multi-image analysis	MiniCPM-V 4.5
Visual agent tasks	Qwen3-VL-30B
Large documents (30K+ tokens)	Nanonets-OCR2-3B

□□ Building from Source

```
# Clone the repository
git clone https://code.foss.global/host.today/ht-docker-ai.git
cd ht-docker-ai

# Build all images
./build-images.sh

# Run tests
pnpm test
```

□□ Troubleshooting

Model download hangs

```
docker logs -f <container-name>
```

Model downloads can take several minutes (~5GB for MiniCPM-V, ~20GB for Qwen3-VL).

Out of memory

- **GPU:** Use a lighter model variant or upgrade VRAM
- **CPU:** Increase container memory: `--memory=16g`

API not responding

1. Check container health: `docker ps`
2. Review logs: `docker logs <container>`
3. Verify port: `curl localhost:11434/api/tags` or `curl localhost:8000/health`

Enable NVIDIA GPU support on host

```
# Install NVIDIA Container Toolkit
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-
toolkit.list | \
  sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg]
https://#g' | \
  sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

GPU Memory Contention (Multi-Model)

When running multiple VLMs on a single GPU:

- vLLM and Ollama both need significant GPU memory
- **Single GPU:** Run services sequentially (stop one before starting another)
- **Multi-GPU:** Assign each service to a different GPU via `CUDA_VISIBLE_DEVICES`

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the LICENSE file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #4
Created 2026-03-28 11:09:32 UTC by foss.global Team
Updated 2026-03-28 12:15:46 UTC by foss.global Team