

Vast ai login

Regions: ANY 1X 2X 4X 8X 9X+ On-Demand 4 GPUs Planet Earth Auto

ID	Host	Location	GPU Type	TFLOPS	VRAM	Max CUDA	Availability	Cost
m:28301	host:149988	Virginia, US	1x Titan V	15.7	12 GB VRAM 553.6 GB/s	Max CUDA: 12.4	100%	\$0
m:26485	host:149988	Virginia, US	1x Titan V	15.7	12 GB VRAM 555.1 GB/s	Max CUDA: 12.4	100%	\$0
m:27389	host:155348	South Korea, KR	1x Titan Xp	11.7	12 GB VRAM 410.4 GB/s	Max CUDA: 12.8	100%, 11d	\$0

Search for GPUs

- NVIDIA
- RTX
- Data Center
- GTX
- Quadro
- Titan
- AMD

37 templates shown

ARM SSH Jupyter

**Recommended**

NVIDIA CUDA 

Image: <https://hub.docker.com/r/vastai/bas...> 

   

## Verify when up

The screenshot shows the Vast.ai Instances page. On the left is a sidebar with navigation links: Search, Templates, Instances (selected), Storage, Serverless (NEW), Account, Billing, Earnings, Members, and Audit Logs. The main area is titled "Instances (1)". It displays a single instance named "1x Titan V". The instance details include:

Instance ID:	30216047	Max CUDA:	12.4	DLPf	Network	CPU	Disk	Motherboard
Host:	149988	15.7 TFLOPS	-	250 ports	27.9 Mbps ↑	AMD EPYC 7K62 48-Cor...	Samsung SSD 970 EVO ...	EPYCD8
Machine ID:	28301	VRAM 0/12.0 GB	553.5 DLP/S/hr	593.5 Mbps ↓	1798.0 MB/s	24.0/96 CPU	0/32.0 GB	PCIE 3.0/16x
Vol:	No Volumes	553.5 GB/s	-	-	-	0/32.2 GB	0.0/32.0 GB	11.9 GB/s

Status: not running

Actions: Creating..., Stop, Delete, Power, Refresh, Log, Share, Details.

## Verify cuda installed.

The screenshot shows the Vast.ai Applications page. On the left is a sidebar with navigation links: Applications (Manage Your Services), Tunnels (Open New Ports), Instance Logs (Debugging), and Tools & Help (Resources & Guides). The main area is titled "Applications". It contains four application cards:

- Instance Portal**: Currently Active. Advanced Connection Options. Launch Application button.
- Jupyter**: Launch Application button. Advanced Connection Options.
- Jupyter Terminal**: Launch Application button. Advanced Connection Options.
- Syncthing**: Launch Application button. Advanced Connection Options.
- Tensorboard**: Launch Application button. Advanced Connection Options.

```

Activated conda/uv virtual environment at /venv/main
(main) root@C.30216047:/workspaces nvidia-smi
Mon Jan 19 05:35:15 2026
+-----+
| NVIDIA-SMI 550.127.05      Driver Version: 550.127.05     CUDA Version: 12.4 |
+-----+
| GPU  Name Persistence-M | Bus-Id Disp.A  Volatile Uncorr. ECC | | | | | |
| Fan  Temp  Perf  Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|          |          |          |          |          |          |          |
| 0  NVIDIA TITAN V      On   00000000:41:00.0 Off  1MiB / 12288MiB | 0%       N/A |
| 0%   25C    P8   23W / 250W |          |          |          |          |          |
+-----+
Processes:
+-----+
| GPU  GI  CI      PID  Type  Process name          GPU Memory |
| ID   ID              ID           Usage          |
+-----+
| No running processes found |
+-----+
(main) root@C.30216047:/workspaces$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Thu_Mar_28_02:18:24_PDT_2024
Cuda compilation tools, release 12.4, V12.4.131
Build cuda_12.4.r12.4/compiler.34097967_0
(main) root@C.30216047:/workspaces$
```

```
python -m pip install uvicorn fastapi
```

```
(main) root@C.30216047:/workspace$ python -m uvicorn server:app --host 0.0.0.0 --port 8000
Traceback (most recent call last):
  File "/venv/main/lib/python3.10/runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "/venv/main/lib/python3.10/runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "/venv/main/lib/python3.10/site-packages/uvicorn/__main__.py", line 4, in <module>
    uvicorn.main()
  File "/venv/main/lib/python3.10/site-packages/click/core.py", line 1485, in __call__
    return self.main(*args, **kwargs)
  File "/venv/main/lib/python3.10/site-packages/click/core.py", line 1406, in main
    rv = self.invoke(ctx)
  File "/venv/main/lib/python3.10/site-packages/click/core.py", line 1269, in invoke
```

```
return ctx.invoke(self.callback, **ctx.params)
File "/venv/main/lib/python3.10/site-packages/click/core.py", line 824, in invoke
    return callback(*args, **kwargs)
File "/venv/main/lib/python3.10/site-packages/uvicorn/main.py", line 424, in main
    run(
File "/venv/main/lib/python3.10/site-packages/uvicorn/main.py", line 594, in run
    server.run()
File "/venv/main/lib/python3.10/site-packages/uvicorn/server.py", line 67, in run
    return asyncio_run(self.serve(sockets=sockets), loop_factory=self.config.get_loop_factory())
File "/venv/main/lib/python3.10/site-packages/uvicorn/_compat.py", line 60, in asyncio_run
    return loop.run_until_complete(main)
File "/venv/main/lib/python3.10/asyncio/base_events.py", line 649, in run_until_complete
    return future.result()
File "/venv/main/lib/python3.10/site-packages/uvicorn/server.py", line 71, in serve
    await self._serve(sockets)
File "/venv/main/lib/python3.10/site-packages/uvicorn/server.py", line 78, in _serve
    config.load()
File "/venv/main/lib/python3.10/site-packages/uvicorn/config.py", line 439, in load
    self.loaded_app = import_from_string(self.app)
File "/venv/main/lib/python3.10/site-packages/uvicorn/importer.py", line 22, in
import_from_string
    raise exc from None
File "/venv/main/lib/python3.10/site-packages/uvicorn/importer.py", line 19, in
import_from_string
    module = importlib.import_module(module_str)
File "/venv/main/lib/python3.10/importlib/_init_.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
File "<frozen importlib._bootstrap>", line 1050, in _gcd_import
File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
File "<frozen importlib._bootstrap_external>", line 883, in exec_module
File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
File "/workspace/server.py", line 7, in <module>
    import torch
ModuleNotFoundError: No module named 'torch'
```

```
pip install torch --index-url https://download.pytorch.org/whl/cu118
python -m pip install transformers pillow safetensors huggingface_hub accelerate
pip install matplotlib tiktoken
pip install einops transformers_stream_generator
python -m pip uninstall -y transformers_stream_generator
```

```
python -m pip uninstall -y transformers
python -m pip install "transformers==4.37.2"
python -m pip install transformers_stream_generator
python -m pip install -U "optimum[gptq]"
python -m pip install -U auto-gptq
```

```
python -m pip uninstall -y transformers peft optimum auto-gptq  
transformers_stream_generator
```

```
python -m pip install \  
    "transformers==4.37.2" \  
    "peft==0.7.1" \  
    "optimum==1.17.1" \  
    "auto-gptq==0.7.1" \  
    "transformers_stream_generator" \  
    "accelerate" "safetensors" "pillow" "huggingface_hub"
```

```
python - <<'PY'  
import transformers  
import importlib.metadata as md  
print("transformers", transformers.__version__)  
print("peft", md.version("peft"))  
print("optimum", md.version("optimum"))  
print("auto-gptq", md.version("auto-gptq"))  
PY
```

```
1.0.0. Downloads always resume when possible. If you want to force a new download, use `force_download=True`.  
warnings.warn()  
model.safetensors.index.json: 126kB [00:00, 97.1MB/s] | 1.98G/1.98G [00:28<00:00, 70.2MB/s]  
model-00001-of-00005.safetensors: 100% | 1.98G/1.98G [00:27<00:00, 71.6MB/s]  
model-00002-of-00005.safetensors: 100% | 2.00G/2.00G [00:28<00:00, 70.5MB/s]  
model-00003-of-00005.safetensors: 100% | 1.99G/1.99G [00:29<00:00, 67.8MB/s]  
model-00004-of-00005.safetensors: 100% | 1.78G/1.78G [00:25<00:00, 70.2MB/s]  
model-00005-of-00005.safetensors: 100% | 1.78G/1.78G [00:25<00:00, 70.2MB/s]  
Downloading shards: 100% | 5/5 [02:19<00:00, 27.89s/it] | 5/5 [00:03<00:00, 1.48it/s]  
Loading checkpoint shards: 100% | 5/5 [00:03<00:00, 1.48it/s] | 221/221 [00:00<00:00, 2.78MB/s]  
generation_config.json: 100% |  
INFO: Started server process [2702]  
INFO: Waiting for application startup.  
INFO: Application startup complete.  
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

Get a new terminal. Have to use Jupyter server not UI



File View Settings Help

New ▾

- Console
- Notebook
- Terminal
- Text File
- Markdown File
- Python File

Open...

New Console for Activity

Save ⌘ S

Save As... ⌘ ⌘ S

Save All

Reload from Disk

Revert to Checkpoint...

Download

Save and Export Notebook As ▾

Trust Notebook

Close and Shut Down Notebook ⌘ ⌘ Q

Log Out

Shut Down

```
Verify image at : rm -f example.jpg
curl -L -o example.jpg "https://upload.wikimedia.org/wikipedia/commons/2/27/
Dog_Looking_Into_The_Sunset.jpg"
file example.jpg
ls -lh example.jpg
```

Makes sure it exists and is 3m not 98 or 100 bytes



```
— client.py —  
import base64  
import requests  
  
with open("example.jpg", "rb") as f:  
    image_b64 = base64.b64encode(f.read()).decode("utf-8")  
  
payload = {  
    "prompt": "Describe the image.",  
    "image_b64": image_b64,  
    "max_new_tokens": 128,  
    "temperature": 0.2  
}  
  
r = requests.post(  
    "http://127.0.0.1:8000/chat",  
    json=payload,  
    timeout=300  
)  
  
print("status:", r.status_code)  
print(r.text)
```

```
(main) root@C.30216047:/workspace$ vim client.py  
(main) root@C.30216047:/workspace$ python client.py  
status: 200  
{"text":"A dog wearing a red collar, watching the sunset."}  
(main) root@C.30216047:
```

—server.py—

```
(main) root@C.30216047:/workspace$ cat server.py
import base64
import io
import os
import tempfile
from typing import Optional

import torch
from fastapi import FastAPI
from pydantic import BaseModel
from PIL import Image
from transformers import AutoTokenizer, AutoModelForCausalLM

# -----
# Model load (Qwen-VL-Chat-Int4)
# -----
MODEL_ID = os.environ.get("MODEL_ID", "Qwen/Qwen-VL-Chat-Int4")

device = "cuda" if torch.cuda.is_available() else "cpu"

tokenizer = AutoTokenizer.from_pretrained(
    MODEL_ID,
    trust_remote_code=True,
)

model = AutoModelForCausalLM.from_pretrained(
    MODEL_ID,
    device_map="auto",      # let HF place on GPU if available
    trust_remote_code=True,
).eval()

# -----
# FastAPI schema
# -----
class ChatReq(BaseModel):
    prompt: str
    image_b64: str
    max_new_tokens: int = 64
    temperature: float = 0.2 # kept for API compatibility; ignored when do_sample=False

app = FastAPI()

def decode_image(image_b64: str) -> Image.Image:
    raw = base64.b64decode(image_b64)
    img = Image.open(io.BytesIO(raw)).convert("RGB")
    return img
```

```
@app.get("/health")
def health():
    dev = torch.cuda.get_device_name(0) if torch.cuda.is_available() else "cpu"
    return {"ok": True, "cuda": torch.cuda.is_available(), "device": dev}

@app.post("/chat")
@torch.inference_mode()
def chat(req: ChatReq):
    # 1) decode base64 image
    img = decode_image(req.image_b64)

    # 2) resize to keep vision compute reasonable
    img.thumbnail((768, 768))

    # 3) Qwen-VL expects an image path in tokenizer.from_list_format
    # with tempfile.NamedTemporaryFile(suffix=".jpg", delete=True) as f:
    #     img.save(f.name, format="JPEG", quality=90)

    query = tokenizer.from_list_format([
        {"image": f.name},
        {"text": req.prompt},
    ])

    # 4) IMPORTANT: use query=query and do_sample=False for stability
    text, _ = model.chat(
        tokenizer,
        query=query,
        history=None,
        do_sample=False,
        max_new_tokens=req.max_new_tokens,
    )

    return {"text": text}

(main) root@C.30216047:/workspace$
```

Credit: \$24.83



.07 cost to debug.











