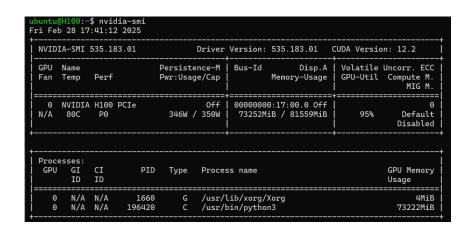


MT [UNIVA-Bllossom/DeepSeek-llama3.1-Bllossom-8B (DOCKER)]

:GPU H100 :CUDA 12.2 : torch: 2.5.1





sudo docker run --runtime nvidia --gpus all --name llama_container -v ~/.cache/huggingface:/root/.cache/huggingface --env "HUC

FastAPI 기반 서버

보통 /vt/chat/completions 을 사용한다면, OpenAI 호환 API 서버를 띄운 것 같은데, FastAPI 기반이면 /generate 가 없을 수도 있어.

1. TGI (Text Generation Inference)

Hugging Face의 text-generation-inference 를 사용했다면 보통 /generate 엔드포인트가 있음.

2. vLLM 서버

vLLM으로 띄웠다면 /generate 가 아니라 /v1/completions 또는 /v1/chat/completions 을 사용해야 함.

도커 컨테이너 실행 sudo docker start llama_container # 도커 컨테이너 종료 sudo docker stop llama_container

도커 컨테이너 상태 확인 sudo docker ps

```
import pymysql
import requests
import re
# LLM FastAPI 서버 URL
LLM_SERVER_URL = "http://localhost:8003/v1/chat/completions"
SIMPLEST = """
적합한 프롬프트
# Corrected function (Gemma 모델 특성)
def call_llm_server(prompt):
  try:
    # OpenAI-style request payload
       "model": "UNIVA-Bllossom/DeepSeek-Ilama3.1-Bllossom-8B (DOCKER)",
       "messages": [
         {
           "role": "user",
           "content": prompt
        }
      ]
    response = requests.post(LLM_SERVER_URL, json=data)
    if response.status_code == 200:
      result_text = response.json().get("choices", [{}])[0].get("message", {}).get("content", "").strip()
      print(result_text)
      return clean_llm_response(result_text)
    else:
      print(f"LLM 서버 오류: {response.status_code}, {response.text}")
      return "0"
  except Exception as e:
    print(f"LLM 서버 요청 실패: {e}")
    return "0"
```