# End-to-End Deployment of Mistral-7B with vLLM and FastAPI

## 1. Overview

This guide describes how to deploy the Mistral-7B model using Hugging Face, serve it using vLLM, and connect it to a FastAPI application for prompt-based inference via HTTP API. The vLLM server hosts the LLM, while FastAPI acts as a proxy API for frontend or client services.

## 2. Prerequisites

- A virtual machine (VM) with GPU (e.g., GCP with A100 or T4).  
- Python 3.10+ installed.  
- Docker (optional).  
- Ports 8000 (vLLM) and 9000 (FastAPI) opened in firewall settings.  
- Basic knowledge of terminal, Python, and FastAPI.

## 3. Installing and Running vLLM with Mistral-7B

1. SSH into your VM.  
  
2. Create a virtual environment and activate it:  
 python3 -m venv vllm\_env  
 source vllm\_env/bin/activate  
  
3. Install vLLM and related packages:  
 pip install vllm transformers accelerate

4. Install Hugging Face CLI:  
 Login into hugging face with the help of token ID  
 hugging face-cli repo clone TheBloke/Mistral-7B-Instruct-v0.2-AWQ  
  
5. Launch vLLM server with the Mistral model:  
 python3 -m vllm.entrypoints.openai.api\_server \  
 --model TheBloke/Mistral-7B-Instruct-v0.2-AWQ \  
 --quantization awq \  
 --max-model-len 4096 \  
 --gpu-memory-utilization 0.85 \  
 --host 0.0.0.0 \  
 --port 8000  
  
6. Ensure you get a response from:  
 curl http://<VM\_IP>:8000/v1/completions

## 4. FastAPI Server Setup

1. Create a file `main.py` with the following content:

from fastapi import FastAPI  
from pydantic import BaseModel  
import httpx  
  
app = FastAPI()  
  
VLLM\_URL = "http://<VLLM\_VM\_IP>:8000/v1/completions"  
  
class PromptRequest(BaseModel):  
 prompt: str  
 max\_tokens: int = 1246  
  
@app.post("/generate")  
async def generate\_text(request\_data: PromptRequest):  
 payload = {  
 "model": "TheBloke/Mistral-7B-Instruct-v0.2-AWQ",  
 "prompt": request\_data.prompt,  
 "max\_tokens": request\_data.max\_tokens  
 }  
 async with httpx.AsyncClient(timeout=30.0) as client:  
 response = await client.post(VLLM\_URL, json=payload)  
 if response.status\_code != 200:  
 return {"error": response.text}  
 return response.json()

2. Run FastAPI with Uvicorn:  
 uvicorn main:app --host 0.0.0.0 --port 9000

## 5. How to Use the FastAPI Endpoint

You can send a request to the FastAPI server like this:

curl -X POST http://<FASTAPI\_VM\_IP>:9000/generate \  
-H "Content-Type: application/json" \  
-d '{"prompt": "What is Quantum computing?", "max\_tokens": 100}'

**6. Output**

{  
    "id": "cmpl-52129c04aab947f99ae1a9d6ffddf0fd",  
    "object": "text\_completion",  
    "created": 1754472621,  
    "model": "TheBloke/Mistral-7B-Instruct-v0.2-AWQ",  
    "choices": [  
        {  
            "index": 0,  
            "text": "?\n\nQuantum computing is a new type of computing that uses quantum bits,  
or qubits, instead of classical bits to represent and process information. A classical bit can  
only be in one of two states - 0 or 1. A qubit, however, can be in a superposition of 0 and 1  
states, meaning it can represent multiple states at once. This property, along with another quantum phenomenon called entanglement, allows quantum computers to perform certain calculations much faster",  
            "logprobs": null,  
            "finish\_reason": "length",  
            "stop\_reason": null,  
            "prompt\_logprobs": null  
        }  
    ],  
    "service\_tier": null,  
    "system\_fingerprint": null,  
    "usage": {  
        "prompt\_tokens": 5,  
        "total\_tokens": 105,  
        "completion\_tokens": 100,  
        "prompt\_tokens\_details": null  
    },  
    "kv\_transfer\_params": null  
}

## 7. Notes and Tips

- Ensure vLLM server is running before starting FastAPI.  
- Use external IPs when testing from outside the VM.  
- Open necessary ports in the VM firewall settings.  
- Increase `max\_tokens` and timeout for longer responses.