Ollama vs VLLM

Ollama: Designed for local deployment of LLMs, emphasizing ease of use and is ideal for developers seeking straightforward integration without extensive setup.

vLLM: A more high-performance tool to run LLMs efficiently, ideal for hosting large scale LLMs on GPUs.

Feature	**Ollama**	**vLLM**
Deployment	Local (CLI & GUI), Docker support	Local & Cloud, CLI-based
Hardware Support	NVIDIA, AMD, Apple Silicon, CPU-only fallback	NVIDIA, AMD, Intel, TPUs, AWS Neuron, IBM PowerPC, Gaudi
Model Management	Easy to switch between models, track different versions.	Compatible with popular HuggingFace models.
Quantization Support	Ollama uses 4-bit quantization	GPTQ, AWQ, INT4, INT8, and FP8
Scalability	Limited horizontal scaling	Efficiently manages and scales LLM deployments across multiple devices or clusters.
Ease of Use	High (CLI commands like pull, run, list, ps, push, rm)	Lower, steeper learning curve comparatively.
Context Limit	2048 Tokens	13k Tokens

Choose Ollama if:

- You need a user-friendly tool for local deployment.
- Your application involves offline research, content creation, or education.
- You prefer a simpler tool with a less steep learning curve.

Choose vLLM if:

- You need high-performance inference for demanding applications.
- Scalability and efficient resource utilization are crucial.
- Your application involves serving multiple users concurrently.
- You are comfortable with a more complex tool that offers greater performance optimization

Llama vs Mixtral comparison

We chose the llama3.2:1b-instruct-q4_0 model for its ease of integration with Ollama as well as its lightweight design.

t

Model	**Ease of Use**	**Ollama Integration Status**	**Hardware Requirement**	**Estimated Cloud Cost (per hour)**	**Maximum output of tokens**
llama3.2:1b-instruct-q4_0	Plug & play	Fully supported	Low (1.3 GB VRAM)	~$0.50–$0.70	131,072
mixtral-8x7b-instruct-q4_0	Requires some tuning	Supported, newer addition	High (28+ GB RAM/VRAM)	$3–$30+	32,768