- You want plug-and-play solutions without backend complexity.

---

6. 🔧 Best Practices

- Use **approximate indexing** (like IVF + PQ or HNSW) for speed and memory efficiency.
- **Benchmark** speed vs. recall trade-offs on your dataset.
- Store serialized indexes on disk and implement update pipelines.
- In production, wrap FAISS with databases or services that handle metadata and persistence (e.g., Milvus uses FAISS under the hood).

🔚 TL;DR

- **FAISS** is a **lightning-fast**, **tunable**, **GPU-ready search library**—great if you can manage infrastructure and don't need DB features.
- Want features like **scaling**, **filters**, **persistence**, or a managed experience? Look at vector databases like **Milvus**, **Qdrant**, **Pinecone**, or **pgvector**.

Let me know if you'd like guidance on choosing an index type or integrating FAISS into your stack!

Skip to content

Skip to content

Skip to content