# Comparison of Vector Databases

This document provides a comparison of popular vector databases, including FAISS, Pinecone, Weaviate, Qdrant, Chroma, and Milvus. The table highlights their type, scalability, latency, filtering support, ease of use, deployment options, cost, and best-fit use cases.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature / Criteria | FAISS | Pinecone | Weaviate | Qdrant | Chroma | Milvus |  |
| Type / Model | Open‑source library (C++/Python, GPU/CPU) | Fully managed SaaS / cloud service | Open‑source DB with GraphQL/REST APIs; has managed & self‑hosted options | Open‑source with SaaS, flexible deployment | Open‑source, lighter weight, geared toward prototyping / LLM / RAG workflows | Open‑source designed for large‑scale/distributed workloads; enterprise features |  |
| Scalability | Millions; billions possible with complexity | Excellent, auto-scaling, large-scale ready | Good, supports replication and scaling | Very good, strong performance at scale | Moderate, suited for smaller‑medium workloads | Strong, enterprise-scale, distributed deployments |  |
| Latency / Query Speed | Very low with GPU acceleration; tuning required | Low latency, optimized managed service | Good latency; depends on cluster setup | Good latency with filtering and resources | Decent at small‑medium scale | Competitive with distributed deployments |  |
| Filtering / Metadata | Limited; needs external systems | Built-in metadata & hybrid search support | Strong schema, metadata, hybrid queries | Very good, payload metadata + hybrid | Basic metadata support | Good support with filtering + indexing |  |
| Ease of Use | Manual setup, index choice, persistence | High, simple APIs, no ops burden | Good, GraphQL/REST APIs, ML integrations | Good SDKs/APIs, flexible deployment | Very easy, quick to prototype | More configuration, cluster management |  |
| Deployment Options | Self‑host only (you manage infra) | SaaS / managed only | Both self‑hosted and managed | Both self‑hosted and cloud options | Primarily self‑hosted; emerging cloud | Both self‑hosted and managed offerings |  |
| Cost | Open-source; infra & ops costs | Premium pricing; can be expensive at scale | Moderate; depends on hosting | Flexible; self‑host or SaaS | Low for small/medium; scaling costs rise | Enterprise-level costs; economies of scale |  |
| Best Fit | Fine control, research, GPU/ANN experimentation | Production ML apps, low latency, enterprise scale | Schema, knowledge graphs, hybrid queries | Flexibility, self‑host/cloud mix, metadata heavy | Prototypes, POCs, small RAG apps | Very large scale, enterprise distributed needs |  |