Picks]

Einat Orr, PhD

Last updated on October 20, 2025

Table of Contents

What is the best vector database you can choose for your project? What are the leading vector databases on the market today? Dive into our overview of the 17 most popular vector databases on the market to understand your options and pick the best tool for your project.

Key Takeaways

- **Vector databases specialize in high-dimensional search:** Unlike relational or NoSQL systems, vector databases are optimized for storing and querying vector embeddings used in LLM and neural network applications.
- **Use cases vary between libraries and databases:** Vector libraries are best for static data scenarios like academic benchmarks, while full vector databases support dynamic applications such as semantic search and personalized recommendations.
- **Metadata and hybrid search are key differentiators:** Many modern vector databases, including Pinecone, MongoDB, and Qdrant, support metadata filtering and hybrid search, enhancing relevance and control in retrieval tasks.
- **Open-source dominance supports customization:** Most featured solutions (e.g., Milvus, Weaviate, Deep Lake, Faiss) are open source, allowing teams to tailor the database for specific infrastructure or performance needs.
- **Data lineage and versioning are emerging priorities:** Tools like Deep Lake integrate version control and reproducibility features, enabling better tracking, rollback, and experiment management in LLM workflows.

vector database is designed to store and retrieve just one type of data: vector embeddings.

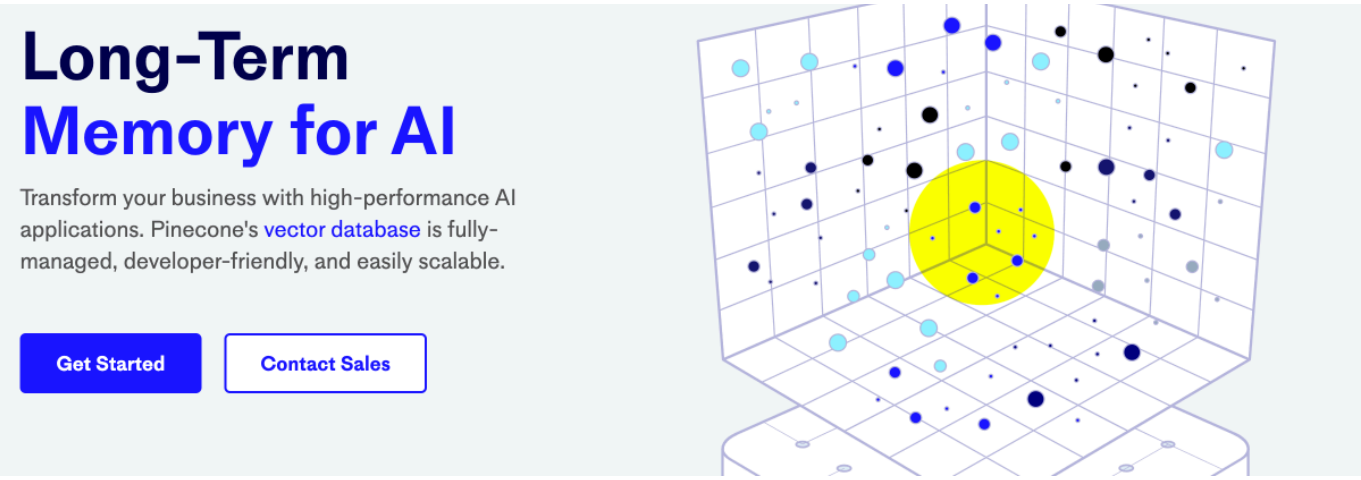
Vector embeddings are the distilled representations of the training data produced as an output from the training stage of the machine learning process. They serve as the filter through which fresh data is processed during inference.

What vector database solutions are available today to help you store and retrieve high-dimensional vectors? Before we move on to the review of the 17 most promising vector databases and libraries, let's clarify the difference between these two technologies.

Vector libraries vs. vector databases

While specialized vector databases are storage systems developed for efficient management of dense vectors, vector libraries are integrated into existing database management systems (DBMS) or search engines to provide similarity search.

Vector libraries are a good choice for static data applications such as academic information retrieval benchmarks. Vector databases are useful for applications that require frequent data changes, such as e-commerce suggestions, image search, and semantic search.



Pinecone: https://www.pinecone.io/

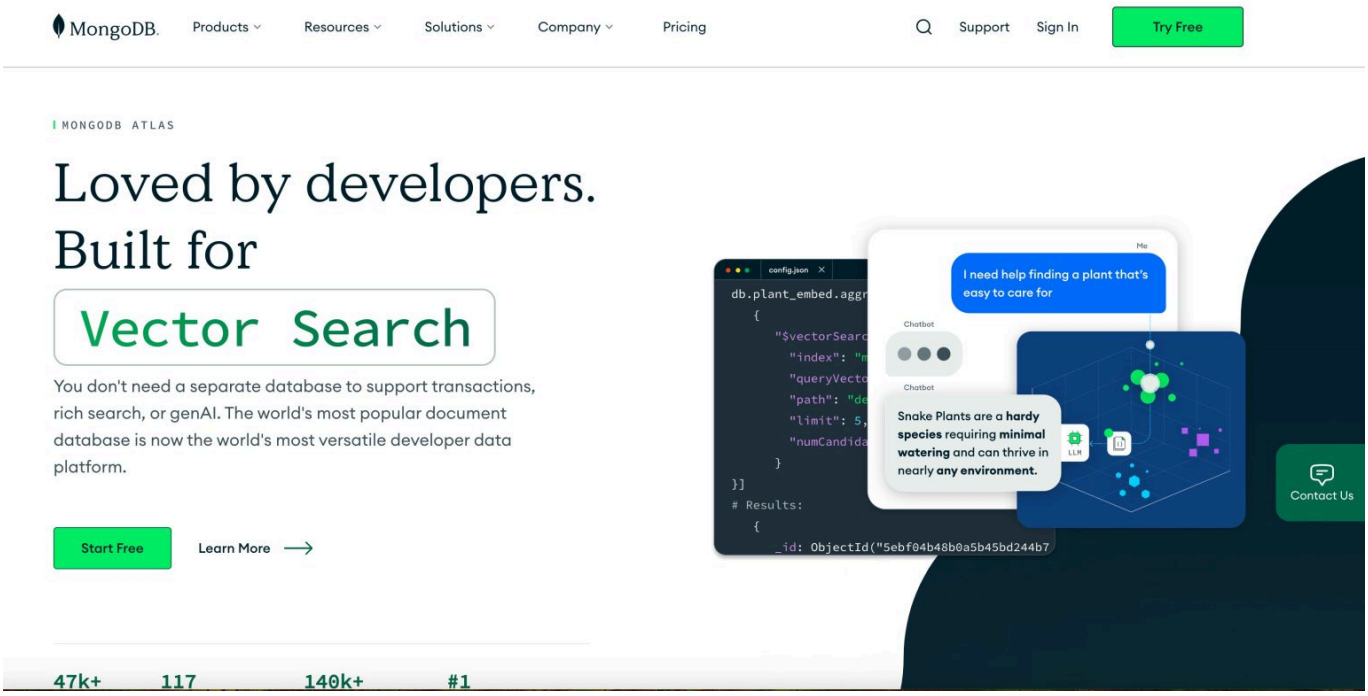
Open source? No

GitHub stars: –

What problem does it solve?

Pinecone is a managed, cloud-native vector database with a straightforward API and no infrastructure requirements. Users can launch, operate, and expand their AI solutions without the need for any infrastructure maintenance, service monitoring, or algorithm troubleshooting.

The solution processes data quickly and lets users use metadata filters and sparse-dense index support for high-quality relevance, guaranteeing speedy and accurate results across a wide range of search needs.



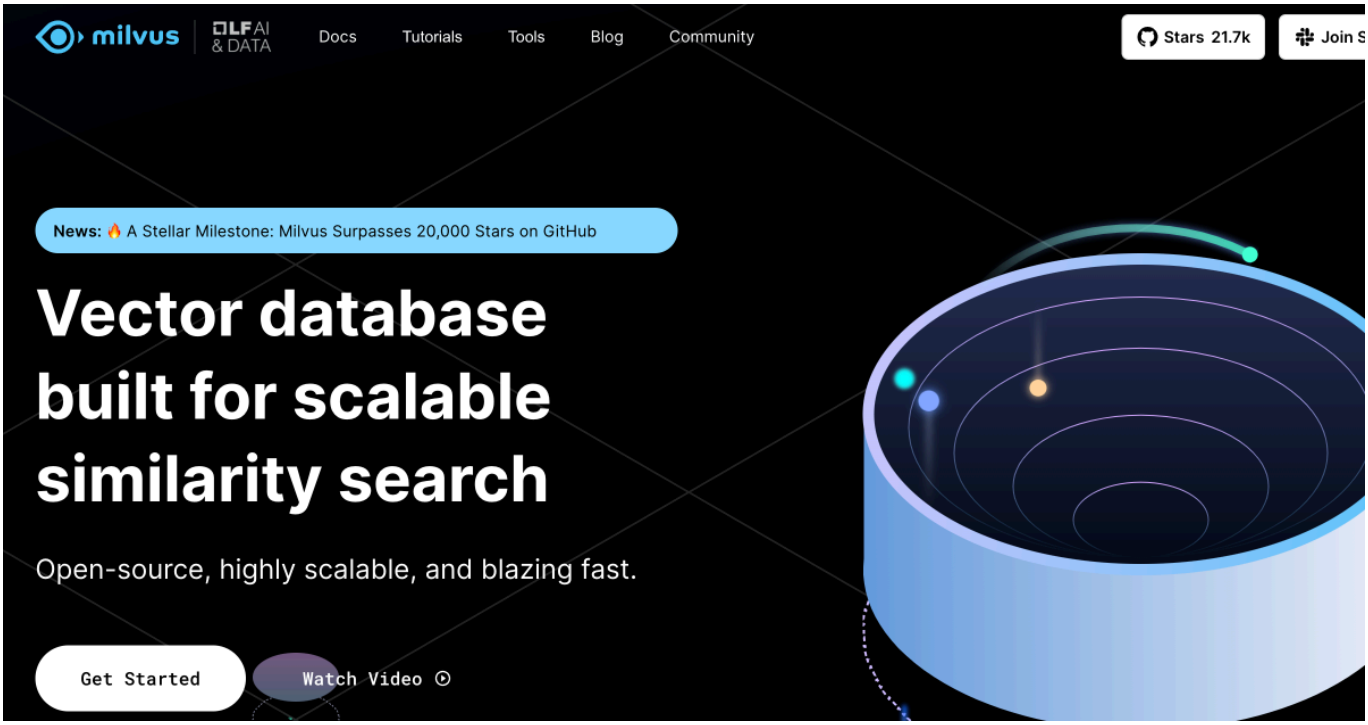
MongoDB: https://www.mongodb.com/

GitHub stars: 25.2k

What problem does it solve?

MongoDB Atlas is the most popular managed developer data platform that can handle a large variety of transactional and search workloads. Atlas Vector Search uses a specialized vector index that is automatically synced with the core database and can be configured to run on separate infrastructure, offering the benefits of an integrated database with the independent scaling that is often why users would look to a vector database.

3. Milvus



Milvus: https://milvus.io/

Open source? Yes

GitHub stars: 21.1k

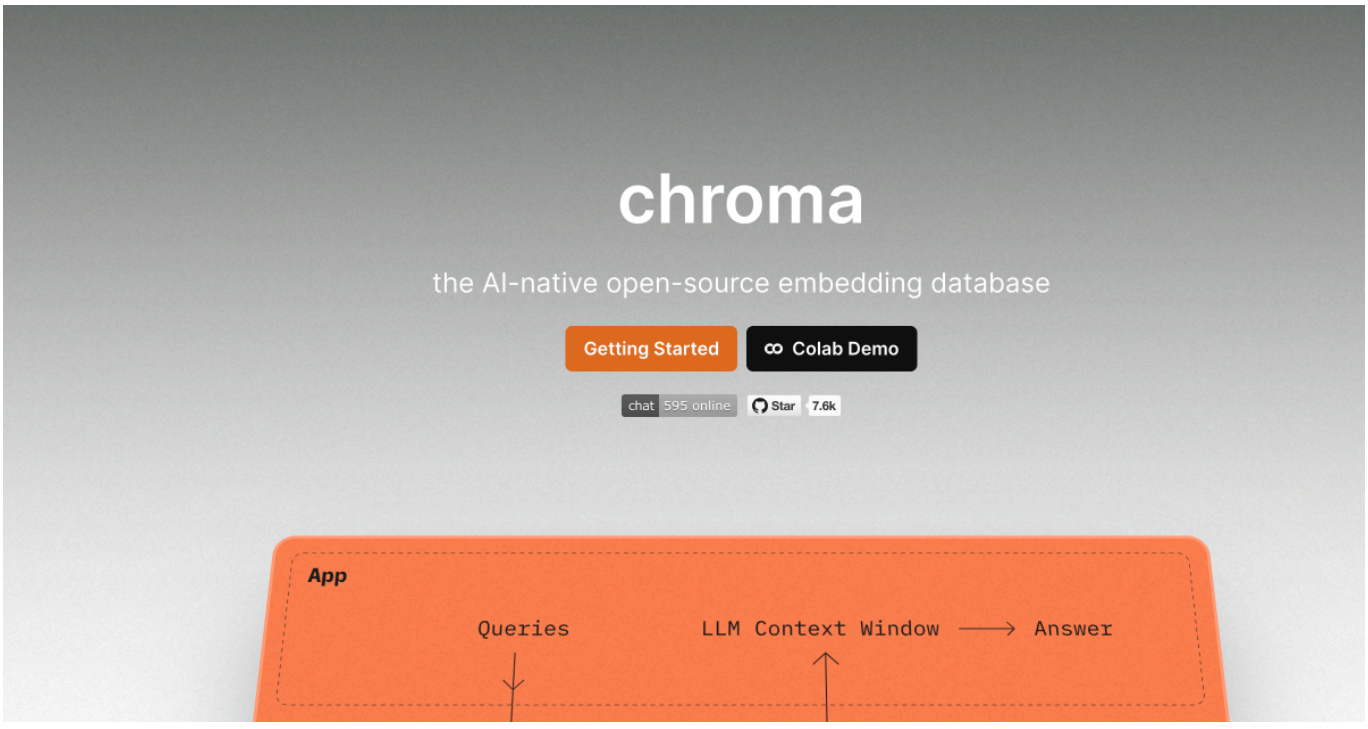
What problem does it solve?

Milvus is an open-source vector database designed to facilitate vector embedding, efficient similarity search, and AI applications. It was published in October 2019 under

Key features

- Searching trillions of vector datasets in milliseconds.
- Unstructured data management is simple.
- Reliable vector database that is always available.
- Highly scalable and adaptable.
- Search hybrid.
- Unified Lambda structure.
- Supported by the community and acknowledged by the industry.

4. Chroma



Chroma: https://www.trychroma.com/

models at the scale of LLMs – as well as avoiding hallucinations.

Many engineers have expressed a desire for "ChatGPT but for data," and Chroma offers this link via embedding-based document retrieval. It also provides 'batteries included' with everything teams need to store, embed, and query data, including strong capabilities like filtering, with more features like intelligent grouping and query relevance on the way.

Key features

- Feature-rich: queries, filtering, density estimates, and many other features.
- LangChain (Python and JavScript), LlamaIndex, and more will be added shortly.
- The same API that runs in your Python notebook scales to your cluster for development, testing, and production.

5. Weaviate

Weaviate: https://github.com/weaviate/weaviate

Open source? Yes

GitHub stars: 6.7k

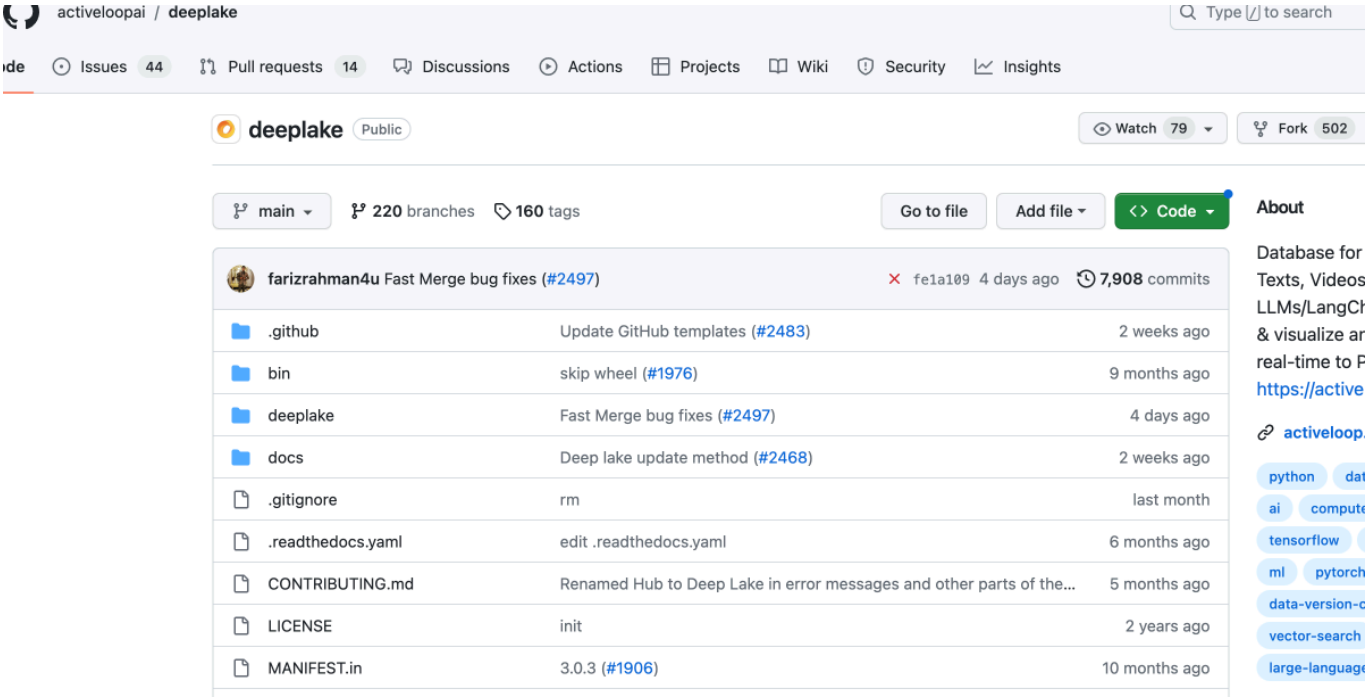
What problem does it solve?

Weaviate is a cloud-native, open-source vector database that is resilient, scalable, and quick. The tool can convert text, photos, and other data into a searchable vector database using cutting-edge machine learning models and algorithms.

It can perform a 10-NN neighbor search in single-digit milliseconds over millions of items. Engineers can use it to vectorize their data during the import process or submit their own vectors, ultimately creating systems for question-and-answer extraction, summarization, and categorization.

Weaviate modules enable the use of prominent services and model hubs like OpenAI, Cohere, or HuggingFace, as well as the use of local and bespoke models. Weaviate is designed with scale, replication, and security in mind.

Key features



Deep Lake: https://github.com/activeloopai/deeplake

Open source? Yes

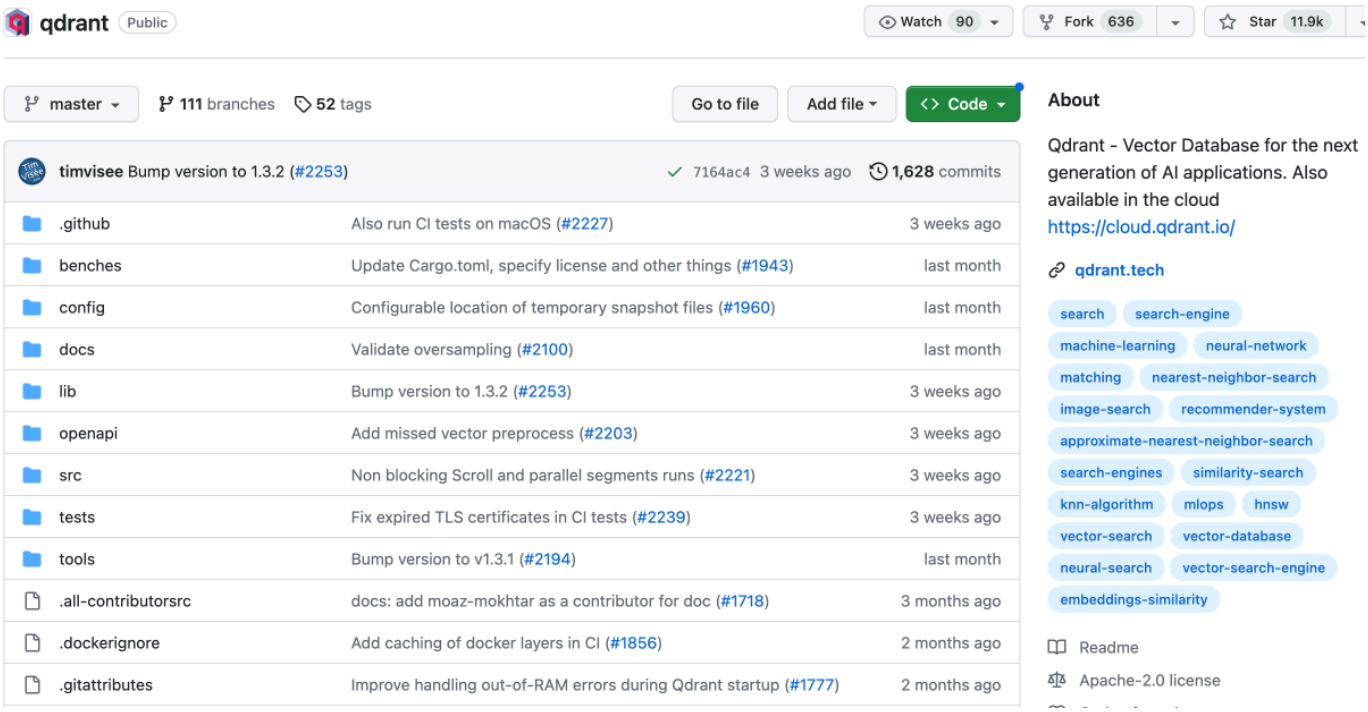
GitHub stars: 6.4k

What problem does it solve?

Deep Lake is an AI database powered by a proprietary storage format designed specifically for deep-learning and LLM-based applications that leverage natural language processing. It helps engineers deploy enterprise-grade LLM-based products faster via vector storage and an array of features.

- Storage for all data types (embeddings, audio, text, videos, images, pdfs, annotations, and so on).
- Querying and vector search,
- Data streaming during training models at scale.
- Data versioning and lineage for workloads.
- Integrations with tools like LangChain, LlamaIndex, Weights & Biases, and many more.

7. Qdrant



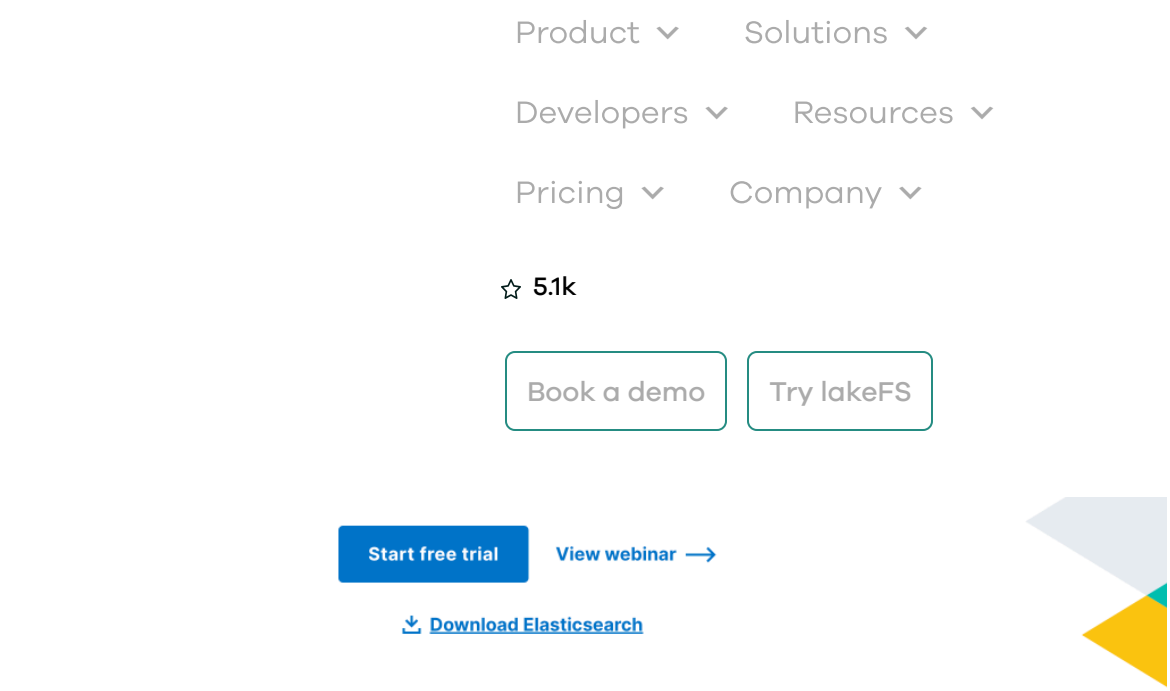
Qdrant: https://github.com/qdrant/qdrant

The tool was designed to provide extensive filtering support. Qdrant's versatility makes it a good pick for neural network or semantic-based matching, faceted search, and other applications.

Key features

- JSON payloads can be connected with vectors, allowing for payload-based storage and filtering.
- Supports a wide range of data types and query criteria, such as text matching, numerical ranges, geo-locations, and others.
- The query planner makes use of cached payload information to improve query execution.
- Write-Ahead during power outages, with the update log recording all operations, allowing for easy reconstruction of the most recent database state.
- Qdrant functions independently of external databases or orchestration controllers, which simplifies configuration.

8. Elasticsearch



Elasticsearch: https://www.elastic.co/elasticsearch/

Open source? Yes

GitHub stars: 64.4k

What problem does it solve?

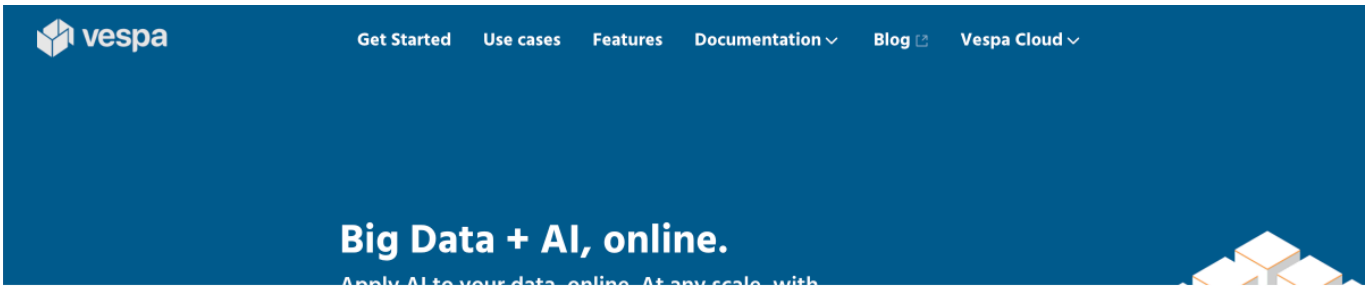
Elasticsearch is an open-source, distributed, and RESTful analytics engine that can handle textual, numerical, geographic, structured, and unstructured data. Based on Apache Lucene, it was initially published in 2010 by Elasticsearch N.V. (now Elastic). Elasticsearch is part of the Elastic Stack, a suite of free and open tools for data intake, enrichment, storage, analysis, and visualization.

Elasticsearch can handle a wide range of use cases – it centrally stores your data for lightning fast search, finetuned relevance, and sophisticated analytics that scale easily. It expands horizontally to accommodate billions of events per second while automatically controlling how indexes and queries are dispersed throughout the cluster for slick operations.

Key features

constant peace of mind.

9. Vespa



Vespa: https://vespa.ai/

Open source? Yes

GitHub stars: 4.5k

What problem does it solve?

- Writes are acknowledged back to the client, issuing them in a few milliseconds when they are durable and visible in queries.
- While servicing requests, writes can be delivered at a continuous rate of thousands to tens of thousands per node per second.
- Data is copied with redundancy that may be configured.
- Queries can include any combination of structured filters, free text search operators, and vector search operators, as well as enormous tensors and vectors.
- Matches to a query can be grouped and aggregated based on a query definition.
- All of the matches are included, even if they are running on several machines at the same time.

10. Vald

Vald offers automated vector indexing and index backup, as well as horizontal scaling, allowing it to search across billions of feature vector data. It's simple to use and extremely configurable – for example, the highly configurable Ingress/Egress filter you can customize to work with the gRPC interface.

Key features

- Vald offers automatic backups through Object Storage or Persistent Volume, allowing for disaster recovery.
- It distributes vector indexes to numerous agents, each of which retains a unique index.
- The tool replicates indexes by storing each index in many agents. When a Vald agent goes down, automatically rebalance the duplicate.
- Highly adaptable – you may choose the number of vector dimensions, replicas, and so forth.
- Python, Golang, Java, Node.js, and more programming languages are supported.

11. ScaNN

ScaNN: https://github.com/google-research/google-research/tree/master/scann

Open source? Yes

GitHub stars: –

What problem does it solve?

ScaNN (Scalable Nearest Neighbors) is a method for efficiently searching for vector similarity at scale. Google's ScaNN proposes a brand-new compression method that significantly increases accuracy. This allows it to outperform other vector similarity search libraries by a factor of two, according to ann-benchmarks.com.

It includes search space trimming and quantization for Maximum Inner Product Search, as well as additional distance functions like Euclidean distance. The implementation is intended for x86 processors that support AVX2.

12. Pgvector

Pgvector: https://github.com/pgvector/pgvector

Open source? Yes

GitHub stars: 4.5k

What problem does it solve?

pgvector is a PostgreSQL extension for searching for vector similarity. You can also use it to keep embeddings as well. Ultimately, pgvector helps you store all of the application data in one place.

Its users get to benefit from ACID compliance, point-in-time recovery, JOINs, and all of the other fantastic features we love PostgreSQL for.

Key features

- Exact and approximate nearest neighbor search
- L2 distance, inner product, and cosine distance
- any language with a PostgreSQL client

13. Faiss

Faiss: https://github.com/facebookresearch/faiss

Open source? Yes

GitHub stars: 23k

What problem does it solve?

Developed by Facebook AI Research, Faiss is an open-source library for fast, dense vector similarity search and grouping. It includes methods for searching sets of vectors of any size, up to those that may not fit in RAM. It also comes with code for evaluation and parameter adjustment.

Faiss is based on an index type that maintains a set of vectors and offers a function for searching through them using L2 and/or dot product vector comparison. Some index types, such as precise search, are simple baselines.

Key features

- Returns not just the nearest neighbor but also the second nearest, third nearest, and k-th nearest neighbor.
- You can search several vectors at once rather than just one (batch processing).

ClickHouse: https://clickhouse.com/

Open source? Yes

GitHub stars: 31.8k

What problem does it solve?

ClickHouse is an open-source column-oriented DBMS for online analytical processing that enables users to produce analytical reports in real time by running SQL queries. The actual column-oriented DBMS design is at the heart of ClickHouse's uniqueness. This distinct design provides compact storage with no unnecessary data accompanying the values, which significantly improves processing performance.

- It uses multicore and multiserver setups to accelerate massive queries, which is a rare feature in columnar DBMSs.
- With robust SQL support, ClickHouse excels at processing a wide range of queries.
- ClickHouse's continuous data addition and quick indexing meet real-time demands.
- Its low latency provides quick query processing, which is critical for online activities.

15. OpenSearch

OpenSearch: https://opensearch.org/

Open source? Yes

You can bring in your models, vectors, and information to enable vector, lexical, and hybrid search and analytics, with built-in performance and scalability.

Key features

- As a vector database, OpenSearch may be used for a variety of purposes, such as search, personalization, data quality, and vector database engine.
- Among its search use cases, you can find multimodal search, semantic search, visual search, and gen AI agents.
- You can create product and user embeddings using collaborative filtering techniques and fuel your recommendation engine with OpenSearch.
- To aid data quality operations, OpenSearch users can use similarity search to automate pattern matching and duplication in data.
- The solution lets you create a platform with an integrated, Apache 2.0-licensed vector database that offers a dependable and scalable solution for embeddings and power vector search.

16. Apache Cassandra

Open source? Yes

GitHub stars: 8.3k

What problem does it solve?

Cassandra is a distributed, wide-column store, NoSQL database management system that is free and open-source. It was designed to handle massive volumes of data across many commodity servers while maintaining high availability with no single point of failure.

Cassandra will soon be equipped with vector search, which demonstrates the Cassandra community's dedication to delivering dependable innovations quickly. Cassandra's popularity is growing among AI developers and businesses dealing with big data volumes as provides them with the capabilities to build complex, data-driven applications.

Key features

- Cassandra will have a new data type to facilitate the storage of high dimensional vectors. This will allow for the manipulation and storage of Float32 embeddings, which are extensively used in AI applications.
- The tool will also provide a new storage-attached index (SAI) dubbed "VectorMemtableIndex" to support approximate nearest neighbor (ANN) search capabilities.

Source: https://kdb.ai/

Open source? No

GitHub stars: –

What problem does it solve?

KDB.AI is a knowledge-based vector database and search engine that enables developers to create scalable, dependable, and real-time apps by offering enhanced

- Its native support for Python and RESTful APIs allows developers to perform common operations such as data ingestion, search, and analytics in their preferred applications and languages.

Best Vector Databases: Comparison

Database	Open Source	Supported index types
Pinecone	No	–
MongoDB	No	HNSW
Milvus	Yes	Multiple index types: FLAT, IVF_FLAT, IVF_PQ, IVF_SQ8, HNSW, HNSW_SQ, HNSW_PQ, HNSW_PRQ, and SCANN
Chroma	Yes	HNSW
Weaviate	Yes	HNSW
Deep Lake	Yes	Inverted and BM25
Qdrant	Yes	HNSW
Elasticsearch	No	HNSW (32, 8, or 4 bit) FLAT (32, 8, or 4 bit)
Vespa	Yes	HNSW

ClickHouse	Yes	HNSW
OpenSearch	Yes	HNSW
Apache Cassandra	Yes	HNSW
KDB.AI	No	Multiple index types: Flat, qFlat, IVF, IVFPQ, HNSW and qHnsw

How to Choose the Best Vector Database for Your Project

Choosing the right vector database can significantly impact your application, but it's not always easy. There are numerous things to consider, ranging from the database's performance and scalability to its interoperability with your current systems. When picking a vector database for your project, consider the following factors:

- **Search accuracy:** The database should return accurate search results. This is especially relevant for applications that need high precision.
- **Scalability:** As your data expands, the database should be able to keep up without sacrificing performance.
- **Performance:** Evaluate the database's speed and efficiency. This includes the speed at which data is stored, retrieved, and searched.

offer troubleshooting and optimization suggestions.

Expert Tip: Decouple Your Vector Store from Data Reality with Commit-Pinned Snapshots

Idan Novogroder
Software Engineer

Idan has an extensive background in software and DevOps engineering. He is passionate about tackling real-life coding and system design challenges. As a key contributor, Idan played a significant role in launching, maintaining, and shaping lakeFS Cloud, which is a fully-managed solution offered by lakeFS. In his free time, Idan enjoys playing basketball, hiking in beautiful nature reserves, and scuba diving in coral reefs.

- Tag and promote successful builds
- Roll back instantly by serving the previous commit/tag.

Tactical Insight: Treat index builds like release artifacts; only promote versions that pass relevance and SLO gates. Version every build, and let automation control which tag is considered production.

Tech & Workflow Context:
Use Spark or Ray to generate embeddings, dbt for upstream features, and Airflow or Dagster to orchestrate the branch → build → bench → tag → merge cycle.

Exporters publish index files (HNSW/IVF/PQ/ScaNN) or backend apply manifests, along with a resolver mapping {commit → vector backend URI}for Milvus, pgvector, Weaviate, or Pinecone.

Engineering Impact or Tradeoff:

- **Pros:** reproducibility, backend flexibility, and one-click rollback via versioned commits.
- **Cons:** additional storage for embeddings and index artifacts – mitigated by lakeFS zero-copy branching and retention/GC policies for untagged builds.

Conclusion

As real-world data increasingly takes the form of complex, unstructured content like images, videos, and natural language, traditional databases often fall short. Vector databases fill this gap, offering a powerful solution for managing and retrieving vector embeddings that fuel modern AI applications.

Frequently Asked Questions

What is the best open-source database for vector search?

Why do you need a vector database?

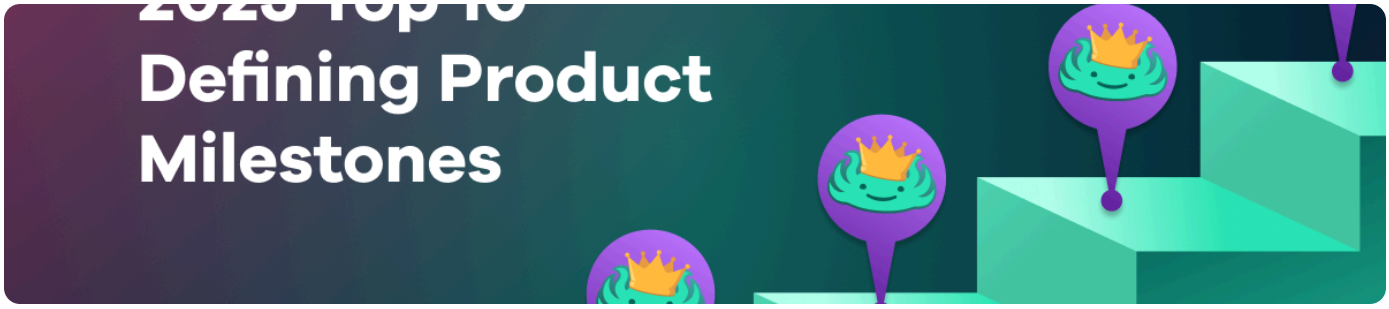
What is vector data best used for?

Is there a free vector database?

Einat Orr, PhD
Author

Einat Orr is the CEO and Co-founder of lakeFS, a scalable data version control platform that delivers a Git-like experience to object-storage based data lakes. She received her PhD. in Mathematics from Tel Aviv University, in the field of optimization in graph theory. Einat previously led several engineering organizations, most recently as CTO at SimilarWeb.

Related articles

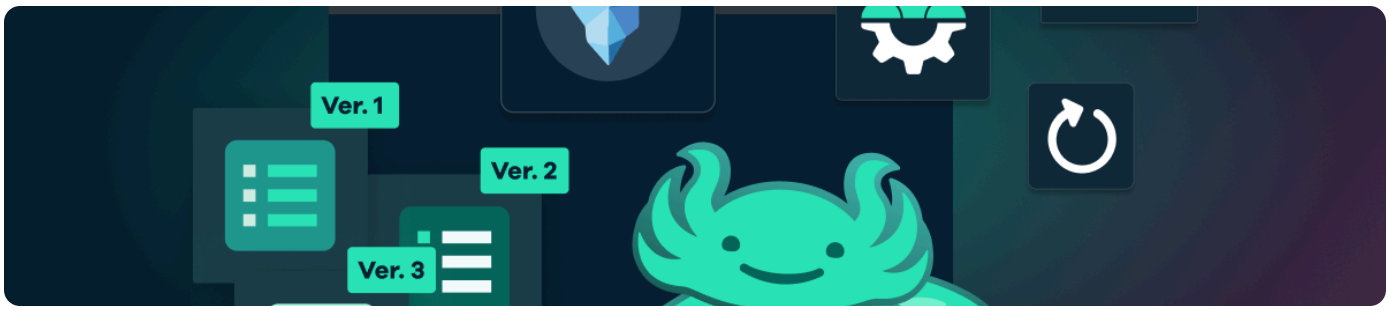


BEST PRACTICES DATA ENGINEERING MACHINE LEARNING

lakeFS Top 10 Defining Product Milestones in 2025

Oz Katz January 14, 2026

Einat Orr, PhD January 4, 2026



DATA ENGINEERING MACHINE LEARNING

What is Iceberg Versioning and How It Improves Data Reliability

Itai Gilo November 10, 2025