

Top 10 open source vector databases

A vector database manages, indexes, and queries high-dimensional vector data, often used in machine learning, data mining, and advanced analytical applications.

What is a vector database?



What is a vector database?

A vector database is a data storage system used to manage, index, and query high-dimensional vector data. Vectors, in this context, represent data points in multi-dimensional space, often used in machine learning, data mining, and other advanced analytical applications.

Vector databases are useful in tasks involving the computation of distances or similarities between vectors, such as recommendation systems, image and video recognition, and natural language processing.

Unlike traditional databases that manage structured rows and columns, vector databases handle more complex, high-dimensional data representations, which are essential for applications requiring efficient similarity searches and pattern recognition. Some general purpose databases, such as PostgreSQL and Cassandra, now support both traditional data formats and vector data.



This is part of a series of articles about [vector databases](#)

Create your database with Instaclustr

See how easy it is to set up your first PostgreSQL or Cassandra clusters



Key features of open source vector databases

Open-source vector databases typically include the following features:

Efficient indexing: Indexing mechanisms such as Approximate Nearest Neighbor (ANN) searches reduce the time required to find similar vector representations, useful for applications involving real-time data analysis.

Similarity search: This feature finds vectors that are close to a given query vector in high-dimensional space, based on measures like Euclidean distance and cosine similarity. It is essential for applications like recommendation engines, where the system needs to



Scalability: As organizations collect more high-dimensional data, the database must efficiently manage this increase without compromising performance. Open-source solutions often offer distributed architectures that help in scaling out, ensuring consistent response times even as data volumes expand.

Integration with machine learning libraries: Open-source vector databases often work with popular machine learning frameworks, allowing for simple deployment of machine learning models directly on the database. This enables the direct application of learned models to the stored data for real-time analysis and predictions.

Community and support: An open-source community can provide assistance through forums, documentation, or contributions to the codebase. These databases often benefit from active communities that help in troubleshooting, feature enhancements, and providing comprehensive usage guides.

Related content: Read our guide to [vector database use cases](#)

Tips from the expert



Ritam Das
Solution Architect

Ritam Das is a trusted advisor with a proven track record in translating complex business problems into practical technology solutions, specializing in cloud computing and big data analytics.

In my experience, here are tips that can help you better utilize open-source vector databases:

1. **Monitor memory usage:** Ensure your vector indexes fit within available memory. If you use PostgreSQL with the pgvector extension you can ensure this by setting the appropriate maintenance_work_mem. Vector data can grow large, and exceeding available memory during indexing can drastically increase build times.
2. **Understand your indexing algorithms:** Use specialized vector indexes like HNSW (Hierarchical Navigable Small Worlds) or IVFFlat (Inverted File with Flat



structure adapts to dataset evolution because it is based on graphs, while IVFFlat is better for memory efficiency and lower build times.

3. **Incorporate vector quantization:** Utilize scalar quantization to reduce 4-byte floats to 2-byte floats, and binary quantization to reduce the dimensions to a single bit. This dramatically cuts storage costs, especially for large datasets with high-dimensional vectors.
4. **Monitor vector database performance:** Implement monitoring and logging tools to track the performance of your vector database, particularly during high-load periods. This can help in identifying bottlenecks and optimizing query strategies in real-time.

Open source vector databases

Here are some of the most popular open source vector databases.

1. OpenSearch



[OpenSearch as a vector database](#) brings together the power of traditional search, analytics, and vector search in one complete package. OpenSearch's vector database capabilities can accelerate artificial intelligence (AI) application development by reducing the effort for builders to operationalize, manage, and integrate AI-generated assets.

Repo: <https://github.com/opensearch-project/OpenSearch>

License: Apache-2.0 license

GitHub stars: ~10K

Contributors: 363+

Key features of OpenSearch



introduction of [GPU-based acceleration](#), powered by NVIDIA cuVS, dramatically reduces indexing times by up to 9.3x while lowering operational costs by 3.75x. It's performance tailored to meet the demands of AI-driven applications, from recommendation engines to hybrid search.

Semantic sentence highlighting: Enhance search result relevance by using machine learning to identify and highlight relevant sentences based on meaning – not just keyword matches.

Seamless Integration with gRPC and MCP: OpenSearch 3.0 streamlines integration with gRPC support for faster data communication and MCP support for effortless AI agent interaction, enabling next-gen AI solutions.

Scalability: Thanks to features like reader and writer separation, which isolates workloads to ensure high efficiency even during resource-heavy tasks. Additionally, pull-based ingestion lets users retrieve data on demand from sources like Apache Kafka, creating adaptive control for large-scale data streams. These advancements prepare businesses to handle the exponential growth of vector data while optimizing resources.

[Create your first OpenSearch cluster](#) and get pro tips for success

2. Facebook AI Similarity Search (Faiss)



Faiss is a library for similarity search and clustering of dense vectors. Developed primarily at Meta's Fundamental AI Research group, Faiss supports searching in sets of vectors of any size, including those too large to fit into RAM. It also offers tools for evaluation and parameter tuning. Written in C++, Faiss includes complete wrappers for Python/numpy.

Repo: <https://github.com/facebookresearch/faiss>

License: MIT license

GitHub stars: ~30K

Contributors: 100+

Key features of Faiss:

Similarity search methods: Supports several methods for similarity search, assuming instances are represented as vectors identified by integers. Allows comparison of vec



Compressed vector representations: Some methods in Faiss use binary vectors and compact quantization codes, allowing for a compressed representation of vectors without retaining the originals.

Indexing structures: Improves search efficiency by adding indexing structures on top of raw vectors, such as HNSW and NSG. These structures help in managing large datasets and speeding up the search process.

GPU implementation: Supports GPU implementations that can take input from either CPU or GPU memory. GPU indexes can replace CPU indexes (e.g., replacing IndexFlatL2 with GpuIndexFlatL2), with automatic handling of memory transfers.

3. Chroma



Chroma is an AI-native open-source vector database used to simplify the development of LLM (Large Language Model) applications. It supports building these applications by making knowledge, facts, and skills easily pluggable for LLMs.

Repo: <https://github.com/chroma-core/chroma>

License: Apache-2.0 license

GitHub stars: 14K

Contributors: 100+

Key features of Chroma:

Storage of embeddings and metadata: Allows efficient storage and management of embeddings and their associated metadata, ensuring easy retrieval and organization of high-dimensional data.

Document and query embedding: Provides tools for embedding documents and queries, enabling similarity searches and ensuring relevant results.

Search embeddings: Supports searching within embeddings to find relevant data points quickly, enhancing the performance of applications that rely on rapid data retrieval.

Speed: Offers high performance with quick data processing capabilities, ensuring that applications can handle large volumes of data without latency issues.





Milvus is an open-source vector database for embedding similarity search and AI applications. It aims to make unstructured data search more accessible and provides a consistent user experience across different deployment environments, including laptops, local clusters, and the cloud.

Repo: <https://github.com/milvus-io/milvus>

License: Apache-2.0 license

GitHub stars: 28K

Contributors: 250+

Key features of Milvus:

Millisecond search on trillion vector datasets: Capable of performing searches with average latency measured in milliseconds, even on trillion-vector datasets.

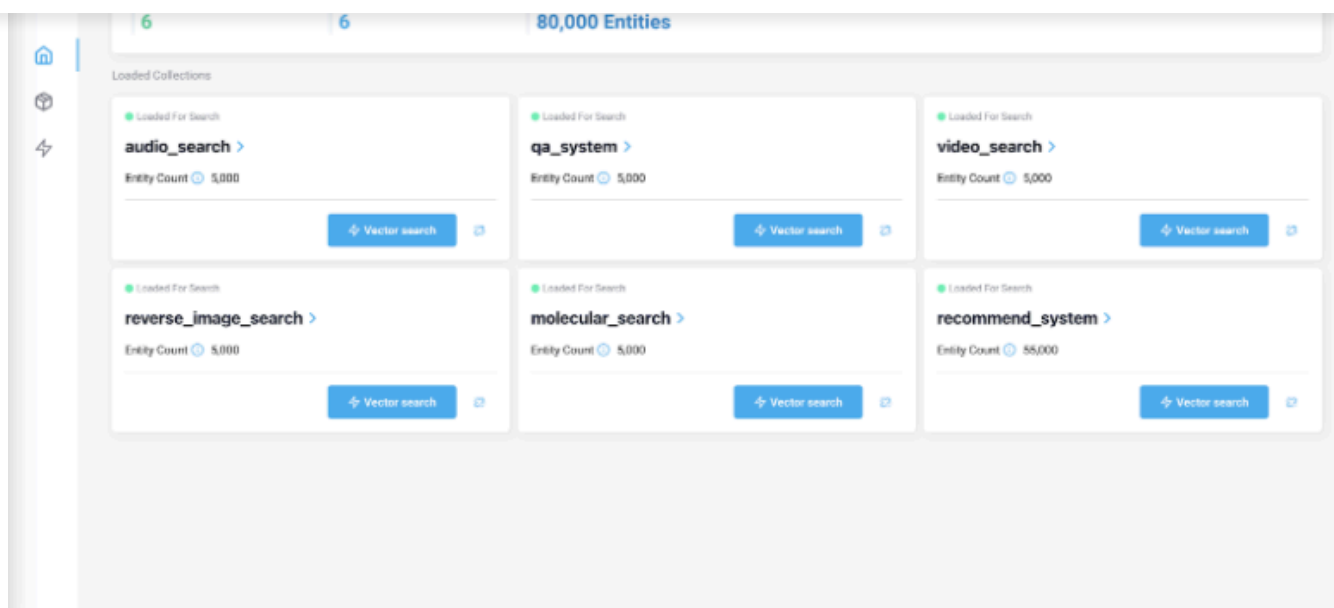
Simplified unstructured data management: Offers rich APIs for data science workflows, simplifying the management and querying of unstructured data.

Consistent user experience: Provides a seamless user experience across various deployment environments.

Always-on database: Features built-in replication and failover/failback mechanisms to maintain business continuity. These features ensure that data and applications remain available and reliable even in the event of disruptions.



NetApp Instaclustr



Source: [Milvus](#)

5. Qdrant



Qdrant (pronounced: *quadrant*) is a vector similarity search engine and vector database offering a production-ready service with an easy-to-use API for storing, searching, and managing vectors along with additional payload data. It provides extended filtering support, making it suitable for neural-network or semantic-based matching, faceted search, and other applications.

Repo: <https://github.com/qdrant/qdrant>

License: Apache-2.0 license

GitHub stars: ~20K

Contributors: 100+

Key features of Qdrant:

Filtering and payload: Allows attaching any JSON payloads to vectors, supporting various data types and query conditions. Enables storage and filtering based on values in these payloads, including keyword matching, full-text filtering, numerical ranges, and geo-locations.

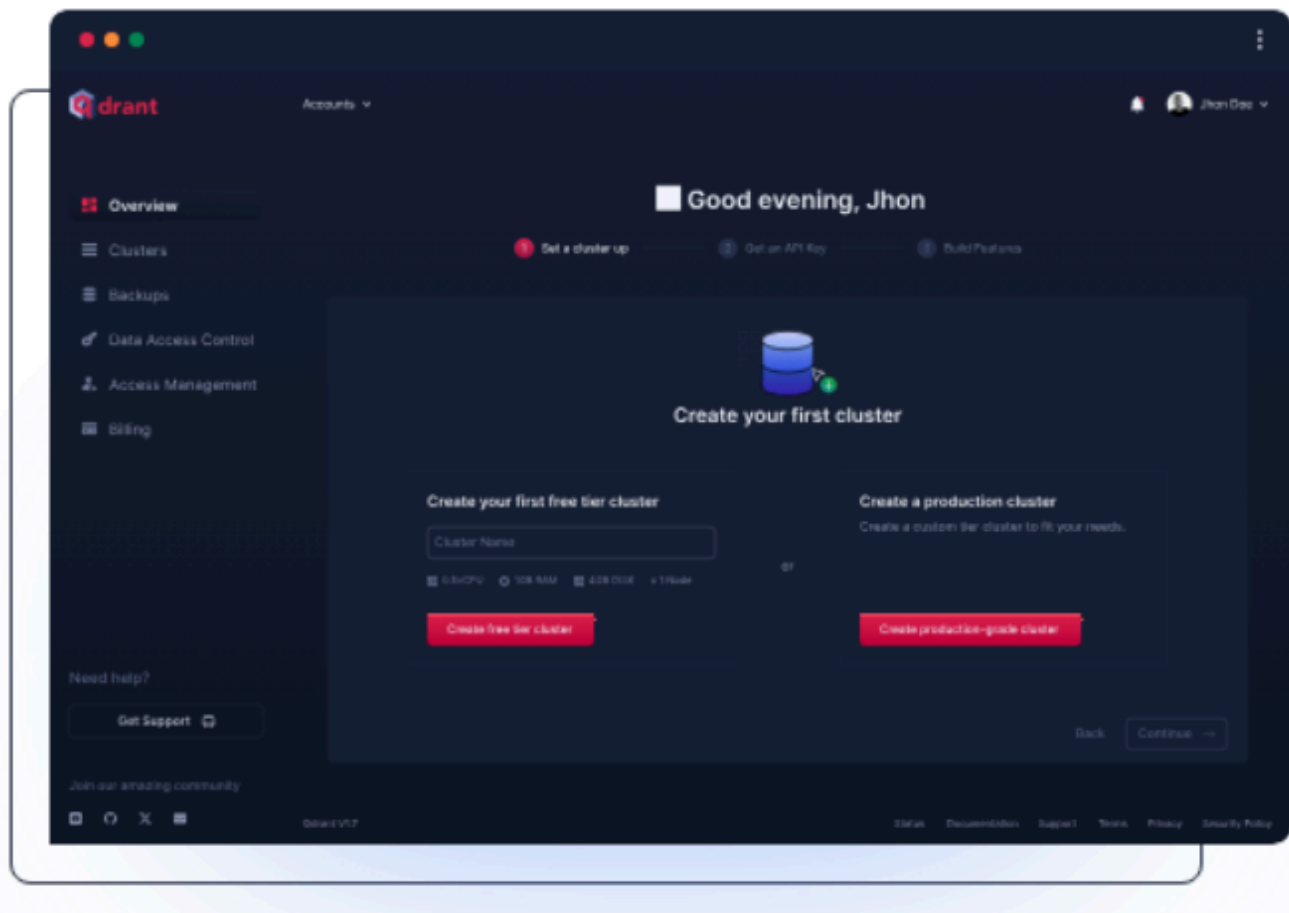


NetApp Instaclustr

functionality of traditional BM25 or TF-IDF ranking methods, allowing for effective token weighting using transformer-based neural networks.

Vector quantization and on-disk storage: Offers multiple options for making vector searches more cost-effective and resource-efficient. Built-in vector quantization reduces RAM usage by up to 97%, dynamically balancing search speed and precision.

Distributed deployment: Supports horizontal scaling through sharding and replication, enabling size expansion and throughput enhancement. Provides zero-downtime rolling updates and dynamic scaling of collection.



Source: [Qdrant](#)

6. Weaviate



NetApp Instaclustr

and more—into a highly searchable vector database.

Repo: <https://github.com/weaviate/weaviate>

License: BSD-3-Clause license

GitHub stars: 10K+

Contributors: 100+

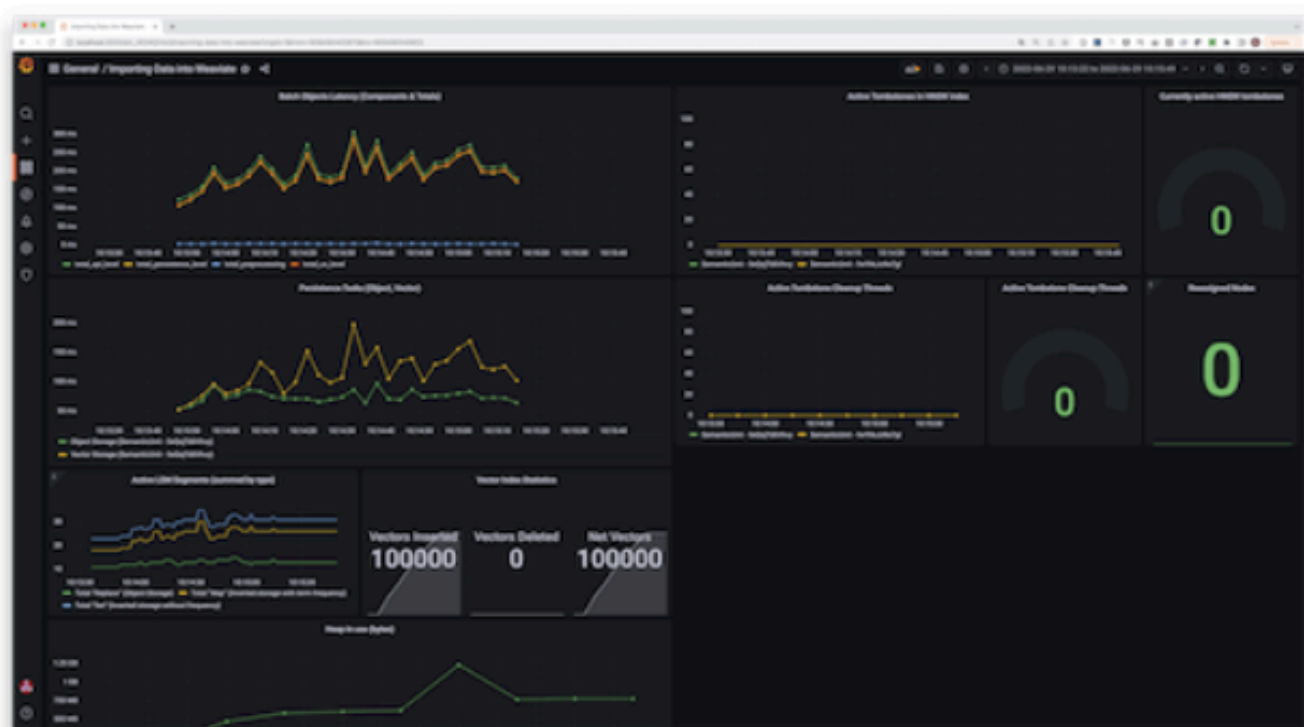
Key features of Weaviate:

Speed: Has a core engine capable of performing a 10-NN nearest neighbor search on millions of objects in milliseconds.

Flexibility: Can vectorize data during the import process or allow users to upload pre-vectorized data. The system's modular architecture provides more than two dozen modules that connect to popular services and model hubs, including OpenAI, Cohere, VoyageAI, and HuggingFace.

Production-readiness: Built with a focus on scaling, replication, and security. Smoothly transitions from rapid prototyping to full-scale production. This ensures that applications can grow without compromising performance or reliability.

Beyond search: Its capabilities extend to recommendations, summarization, and integration with neural search frameworks.



Source: [Weaviate](https://weaviate.io/)



General purpose databases supporting vector data

7. PostgreSQL



PostgreSQL is an open-source relational database that supports vector data through extensions like pgvector. This extension enables efficient similarity search on vector data, integrating with PostgreSQL's ecosystem.

Repo: <https://github.com/postgres/postgres>

License: PostgreSQL License

GitHub stars: 15K+

Contributors: 50+

Key features of PostgreSQL for Vector Data:

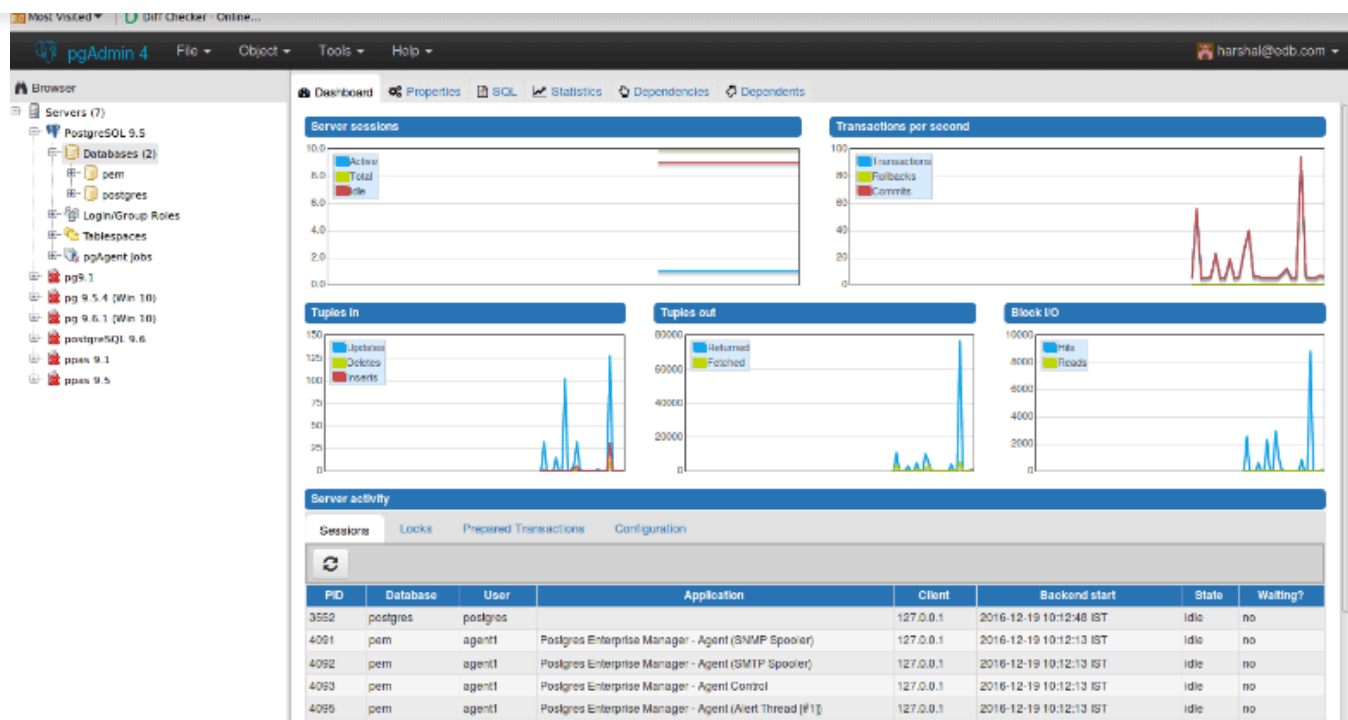
pgvector extension: Allows storage and querying of vector embeddings, facilitating similarity searches within the PostgreSQL environment.

Indexing: Supports various indexing methods, such as ivfflat, to optimize vector search performance.

Scalability: Offers scalability and support for large datasets, making it suitable for vector data applications.

Flexibility: Enables complex queries combining vector searches with traditional SQL operations, providing a unified platform for diverse data types.





Source: [PostgreSQL](#)

8. Cassandra



[Apache Cassandra](#) is a scalable NoSQL database for handling large amounts of data across many commodity servers, providing high availability with no single point of failure. With the introduction of vector search capabilities, Cassandra can efficiently manage vector data.

Repo: <https://github.com/apache/cassandra>

License: Apache-2.0 license

GitHub stars: 8K+

Contributors: ~450

Key features of Cassandra for Vector Data:

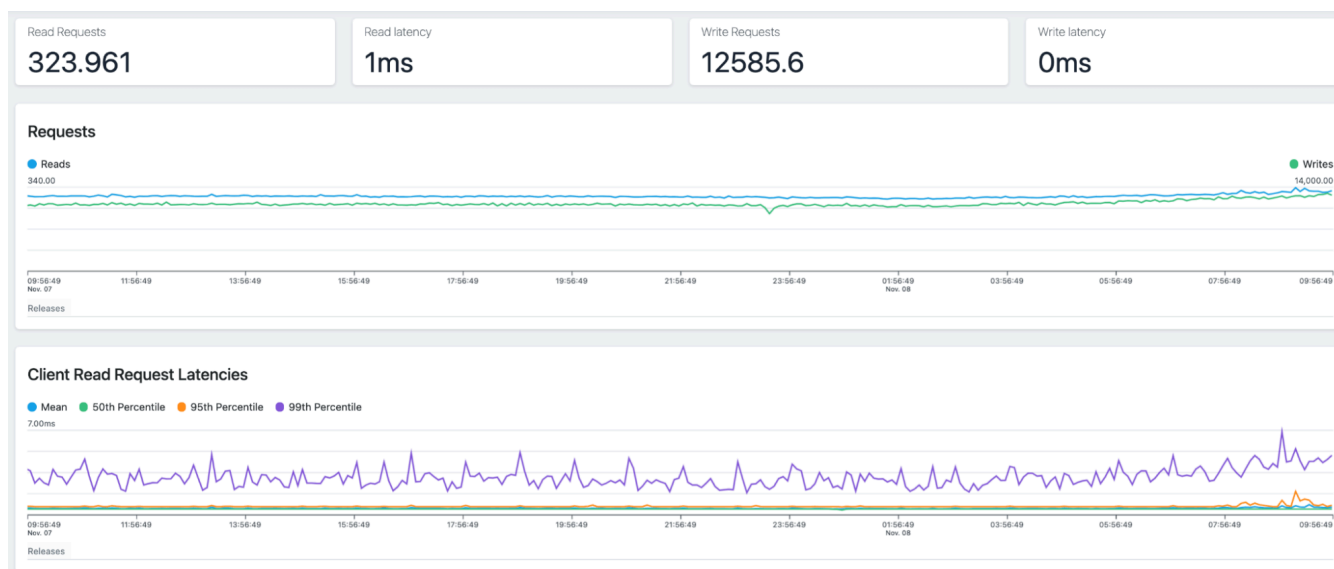
Scalability: Capable of handling petabytes of data, making it suitable for applications requiring large-scale vector storage and retrieval.





Vector search: Supports vector search functionalities, enabling similarity search within its distributed database framework.

Integration: Can integrate with various machine learning frameworks, leveraging its data storage capabilities for AI applications.



Source: [Apache](#)

9. Valkey



Valkey is a specialized vector database designed to manage and search high-dimensional vector data efficiently. It offers a suite of tools and APIs tailored for vector data management and retrieval.

Repo: <https://github.com/valkey-io/valkey>

License: BSD 3-Clause license

GitHub stars: 15K

Contributors: 70+

Key features of Valkey:

Optimized storage: Uses advanced data structures and indexing methods to store and retrieve vector data efficiently.



Rich API: Provides APIs for managing vector data, supporting a range of use cases from AI to search engines.

Scalability: Supports distributed deployments, allowing scaling as data and query loads increase.

10. CockroachDB



CockroachDB is a cloud-native, distributed SQL database designed for ultra-resilient, global applications. It has recently added support for vector data, making it a versatile choice for managing both traditional and vector-based data.

Repo: <https://github.com/cockroachdb/cockroach>

License: BSL 1.1, MIT license, CockroachDB Community License (CCL)

GitHub stars: ~30K

Contributors: 750+

Key features of CockroachDB for Vector Data:

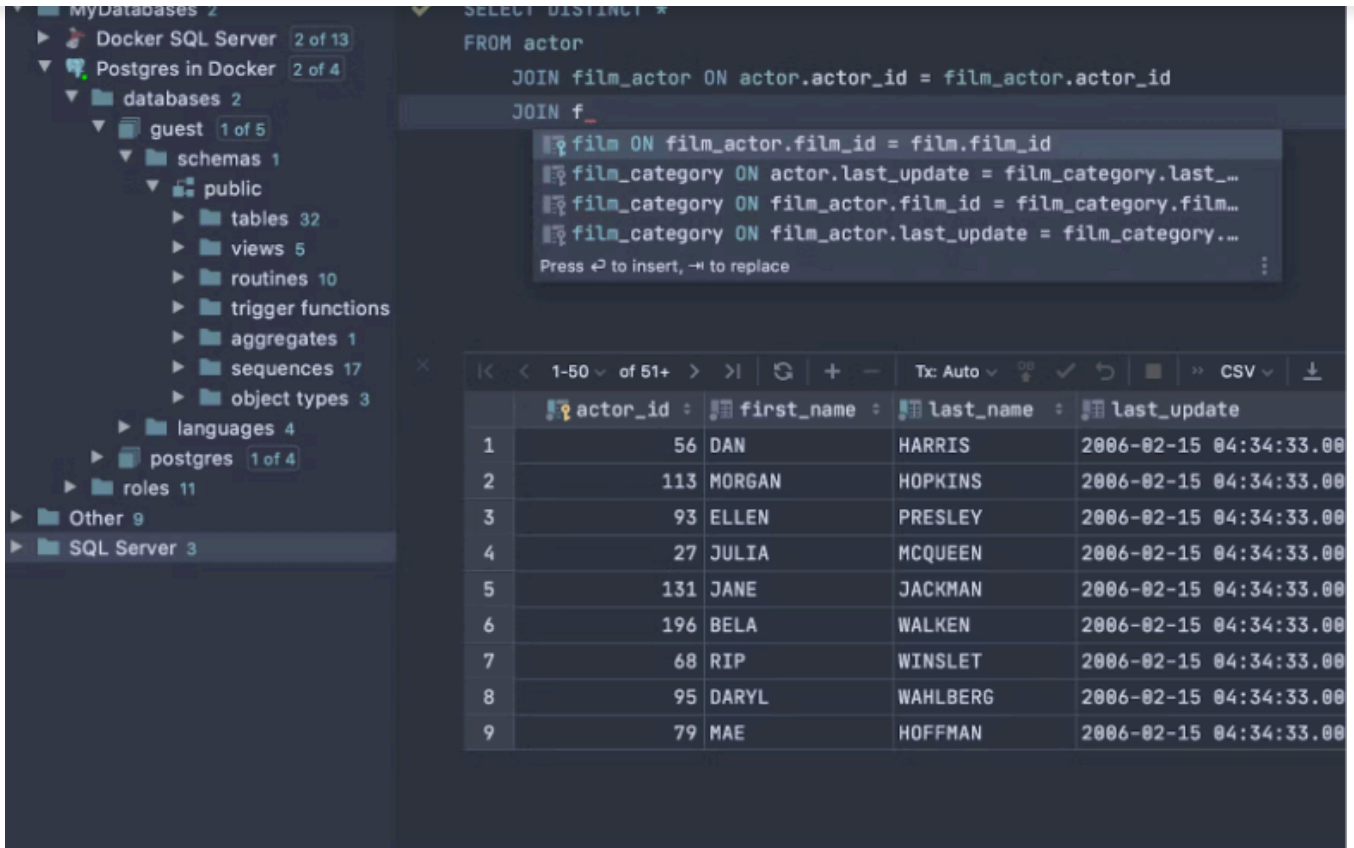
Distributed architecture: Ensures data resilience and availability across multiple geographic locations.

Vector search support: Enables efficient storage and retrieval of vector data, supporting similarity search operations.

SQL compatibility: Combines vector data handling with traditional SQL queries, providing a unified data management solution.

Scalability and resilience: Built to scale out horizontally with strong consistency guarantees, ensuring reliable performance for vector data applications.





The screenshot shows a database management interface. On the left, a tree view displays the database structure: MyDatabases 2, Docker SQL Server 2 of 13, Postgres in Docker 2 of 4, databases 2, guest 1 of 5, schemas 1, public, tables 32, views 5, routines 10, trigger functions, aggregates 1, sequences 17, object types 3, languages 4, postgres 1 of 4, roles 11, Other 9, and SQL Server 3. The main area shows a SQL query: `SELECT DISTINCT * FROM actor JOIN film_actor ON actor.actor_id = film_actor.actor_id JOIN f_ JOIN film ON film_actor.film_id = film.film_id JOIN film_category ON actor.last_update = film_category.last_... JOIN film_category ON film_actor.film_id = film_category.film... JOIN film_category ON film_actor.last_update = film_category...`. Below the query, a table displays the results with columns: actor_id, first_name, last_name, and last_update. The table contains 9 rows of data.

	actor_id	first_name	last_name	last_update
1	56	DAN	HARRIS	2006-02-15 04:34:33.00
2	113	MORGAN	HOPKINS	2006-02-15 04:34:33.00
3	93	ELLEN	PRESLEY	2006-02-15 04:34:33.00
4	27	JULIA	MCQUEEN	2006-02-15 04:34:33.00
5	131	JANE	JACKMAN	2006-02-15 04:34:33.00
6	196	BELA	WALKEN	2006-02-15 04:34:33.00
7	68	RIP	WINSLET	2006-02-15 04:34:33.00
8	95	DARYL	WAHLBERG	2006-02-15 04:34:33.00
9	79	MAE	HOFFMAN	2006-02-15 04:34:33.00

Source: [CockroachDB](#)

Streamlining performance and reliability: Instaclustr's managed approach to vector database management

Instaclustr is a leading provider of managed solutions for open-source technologies offering comprehensive management services for vector databases. These databases are commonly used in applications such as machine learning, data analytics, and recommendation systems.

Instaclustr's management of vector databases encompasses various aspects to ensure optimal performance, scalability, and reliability. Instaclustr provides a fully managed service





the database themselves, allowing them to focus on their core business objectives.

One key aspect of Instaclustr's management approach is expertise in tuning and optimizing vector databases. Instaclustr has a deep knowledge of the underlying technologies and understands the intricacies of configuring the database for specific use cases. This expertise enables fine-tuning of database parameters, indexing strategies, and query optimization techniques to maximize performance and minimize query latency.

Scalability is another critical aspect of Instaclustr's management of vector databases. As datasets grow in size or the workload demands increase, Instaclustr can seamlessly scale the vector database infrastructure to handle the additional load. It employs horizontal scaling techniques, such as sharding and replication, to distribute the data and workload across multiple nodes, ensuring high availability and efficient utilization of resources.

Instaclustr also places a strong emphasis on security and data protection. It implements robust security measures to safeguard the vector database infrastructure, including encryption at rest and in transit, access controls, and regular security audits. Additionally, it provides automated backups and disaster recovery solutions to ensure data integrity and availability in the event of any unforeseen incidents.

For more information on Instaclustr and vector databases, see:

[Instaclustr for Managed Apache Cassandra 5.0](#)

[Use Your Data in LLMs with the Vector Database You already Have: The New Stack](#)

[Instaclustr for PostgreSQL Releases Support for pgvector](#)

[Powering AI Workloads with Intelligent Data Infrastructure and Open Source](#)

Related content

Vector databases: 13 use cases—from traditional to next-gen

Vector database AWS: Comparing service options and how to get started

Vector databases and LLMs: Better together

Vector databases explained: Use cases, algorithms and key features

Vector databases in Azure: 4 service options



What is vector similarity search? Pros, cons, and 5 tips for success

How a vector index works and 5 critical best practices

Related content

What is vector similarity search? Pros, cons, and 5 tips for success

Vector similarity search is a technique used in data processing to find vectors within a dataset that are most similar to a query ...



What are managed database services and 7 key capabilities

A managed database service (MDS) allows organizations to outsource the maintenance and management of database systems to a third-...



Vector search vs semantic search: 4 key differences and how to choose





SIMPLIFY YOUR *Vector database management*

Learn how Instaclustr gives you the expertise and tools you need to seamlessly manage your vector databases and optimize your infrastructure for GenAI and RAG.

[Book a free consultation](#)



Platform	+
Pricing	
Services	+
About	+
Resources	+
Policies	+

©2025 NetApp Copyright. NETAPP, the NETAPP logo, Instaclustr and the marks listed at <https://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners. Apache®, Apache Cassandra®, Apache Kafka®, Apache Spark™, and Apache ZooKeeper™ are trademarks of The Apache Software Foundation.

