Memilih database AWS vektor untuk kasus penggunaan RAG

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Memilih database AWS vektor untuk kasus penggunaan RAG

Table of Contents

Memilih database AWS vektor untuk kasus penggunaan RAG

Mayuri Shinde, Ivan Cui, dan Anand Bukkapatnam Tirumala, Amazon Web Services

Maret 2026 (sejarah dokumen)

Database vektor menjadi semakin penting bagi organisasi yang menerapkan aplikasi AI generatif. Database ini menyimpan dan mengelola vektor, yang merupakan representasi numerik data yang memungkinkan pemrosesan teks, gambar, dan konten lainnya dengan cara yang menangkap makna dan hubungannya.

Saat organisasi mengeksplorasi opsi basis data vektor AWS, mereka perlu memahami kemampuan, pertukaran, dan praktik terbaik untuk solusi yang berbeda. Panduan ini membantu Anda membandingkan penyimpanan vektor yang umum digunakan AWS dan membuat keputusan berdasarkan informasi tentang opsi mana yang paling sesuai dengan kebutuhan spesifik atau kasus penggunaan Anda. Baik Anda menerapkan Retrieval Augmented Generation (RAG), membangun sistem rekomendasi, atau mengembangkan aplikasi AI lainnya, panduan ini menyediakan kerangka kerja untuk membantu Anda mengevaluasi dan memilih solusi database vektor.

Audiens yang dituju

Panduan ini ditujukan untuk orang-orang dalam peran berikut:

- Ilmuwan data dan insinyur pembelajaran mesin (ML) yang menggunakan database vektor untuk menyimpan dan mengambil data dimensi tinggi untuk model ML.
- Insinyur data yang merancang dan mengimplementasikan jaringan data yang mencakup database vektor untuk menyimpan dan memproses data dimensi tinggi.
- MLOps insinyur yang menggunakan database vektor sebagai bagian dari pipa ML untuk menyimpan dan melayani output model atau representasi perantara.
- Insinyur perangkat lunak yang mengintegrasikan database vektor ke dalam aplikasi yang membutuhkan pencarian kesamaan atau sistem rekomendasi.
- DevOps insinyur yang bertanggung jawab untuk menyebarkan dan memelihara database vektor di lingkungan produksi, memastikan skalabilitas dan keandalan.
- Peneliti AI yang menggunakan database vektor untuk menyimpan dan menganalisis kumpulan data besar penyematan atau vektor fitur.

- Manajer produk AI yang perlu memahami kemampuan dan keterbatasan database vektor untuk membuat keputusan berdasarkan informasi tentang fitur dan arsitektur produk.

Ikhtisar vektor

Vektor adalah representasi numerik yang membantu mesin memahami dan memproses data. Dalam AI generatif, mereka melayani dua tujuan utama:

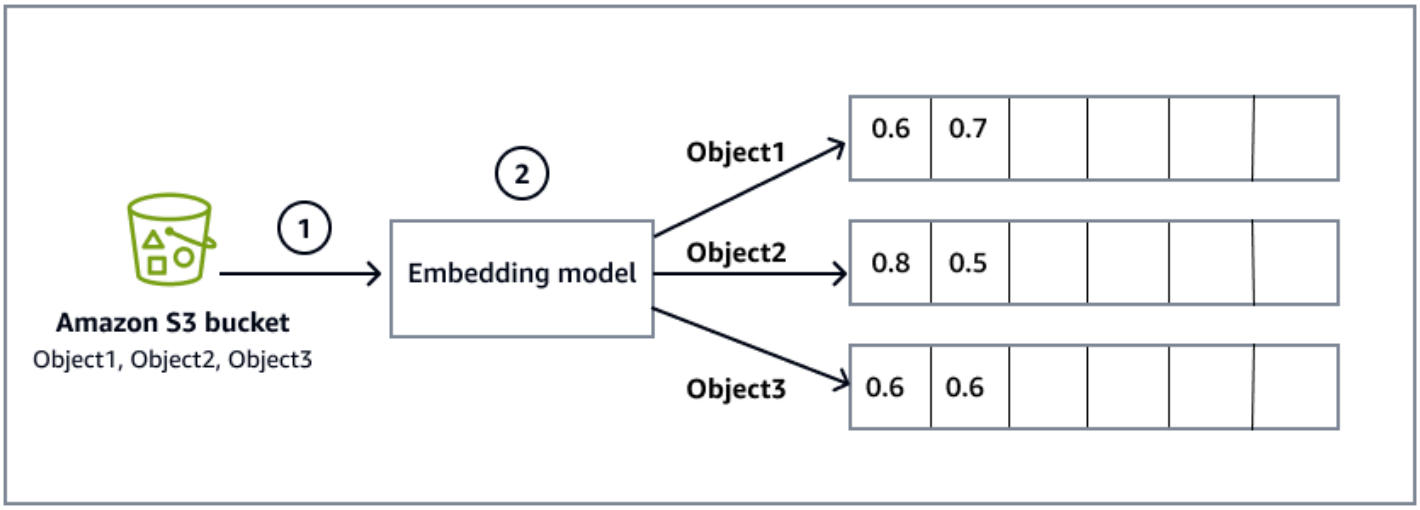
- Mewakili ruang laten yang menangkap struktur data dalam bentuk terkompresi
- Membuat embeddings untuk data, seperti kata-kata, kalimat, dan gambar

Model penyematan seperti Word2Vec,, GloVedan Amazon Titan Text Embeddings mengubah data menjadi vektor melalui proses yang disebut embedding. Model penyematan ini dapat melakukan hal berikut:

- Belajar dari konteks untuk merepresentasikan kata-kata sebagai vektor
- Tempatkan kata-kata serupa lebih dekat bersama-sama dalam ruang vektor
- Memungkinkan mesin untuk memproses data dalam ruang kontinu

Diagram berikut memberikan gambaran tingkat tinggi dari proses penyematan:

1. Bucket Amazon Simple Storage Service (Amazon S3) berisi file yang merupakan sumber data dari mana sistem akan membaca dan memproses informasi. Bucket Amazon S3 ditentukan selama konfigurasi basis pengetahuan Amazon Bedrock, yang juga mencakup sinkronisasi data dengan basis pengetahuan.
2. Model penyematan mengonversi data mentah dari file objek di bucket Amazon S3 menjadi embeddings vektor. Misalnya, `Object1` diubah menjadi vektor `[0.6, 0.7, ...]` yang mewakili isinya dalam ruang multi-dimensi.



Penyematan kata sangat penting untuk pemrosesan bahasa alami (NLP) karena mereka melakukan hal berikut:

- Tangkap hubungan semantik antar kata
- Aktifkan pembuatan teks yang relevan secara kontekstual
- Kekuatan model bahasa besar (LLM) untuk menghasilkan respons seperti manusia

Ikhtisar database vektor

Database vektor adalah sistem khusus yang menyimpan dan menanyakan vektor dimensi tinggi secara efisien. Database ini sangat penting untuk aplikasi Retrieval Augmented Generation (RAG).

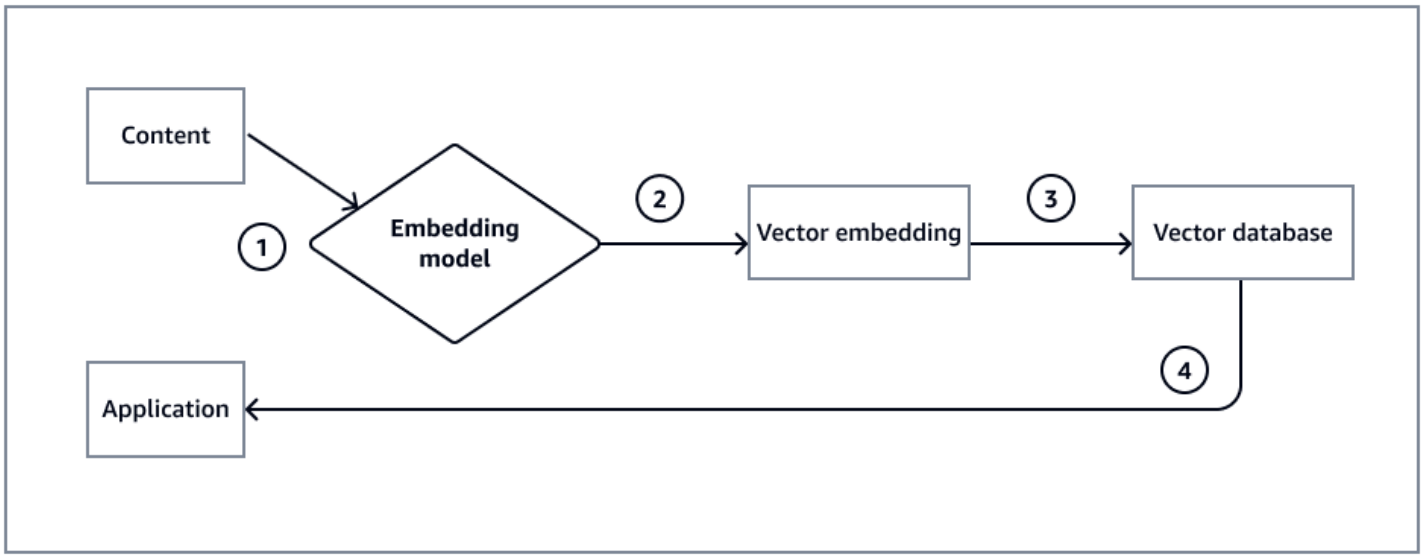
Database vektor menangani konversi dan penyimpanan data dengan cara berikut:

- Objek (seperti audio, gambar, dan file teks) dikonversi ke vektor dengan menggunakan model embedding.
- Vektor disimpan dalam format data khusus.
- Database vektor memungkinkan pencarian kesamaan yang cepat.

Database vektor menawarkan beberapa keunggulan utama dibandingkan database tradisional, membuatnya sangat cocok untuk tantangan data modern. Mereka secara khusus dioptimalkan untuk operasi vektor dan menangani data dimensi tinggi secara efisien. Mereka juga mengkhususkan diri dalam pencarian kesamaan yang diperjuangkan oleh basis data tradisional. Di luar kemampuan inti ini, database vektor dibangun untuk memenuhi permintaan yang terus berkembang dari aplikasi AI dan AI generatif. Mereka unggul dalam penyimpanan vektor skala besar dan menggunakan komputasi terdistribusi untuk menyeimbangkan beban kerja di beberapa node. Ini memberikan skalabilitas dan kinerja seiring pertumbuhan volume data.

Diagram berikut menunjukkan implementasi RAG:

1. Konten, seperti dokumen, PDF, atau file teks, dimasukkan ke dalam model embedding sebagai data mentah untuk diproses.
2. Model embedding mengubah data mentah menjadi vektor numerik, yang mewakili makna semantik dari konten.
3. Embeddings vektor yang dihasilkan disimpan dalam database vektor yang dioptimalkan untuk penyimpanan dan pengambilan vektor dimensi tinggi.
4. Aplikasi sekarang dapat menanyakan database vektor sebagai respons terhadap kasus penggunaan seperti pencarian semantik dan rekomendasi konten.



Memilih database vektor yang tidak tepat untuk solusi RAG dapat menyebabkan perjuangan dan keterbatasan yang signifikan termasuk yang berikut:

- Kinerja kueri yang buruk
- Kemacetan skalabilitas
- Tantangan konsumsi data
- Kurangnya fitur canggih, seperti penyaringan dan peringkat
- Kesulitan integrasi dengan sistem lain
- Masalah ketekunan dan daya tahan
- Masalah konkurensi dan konsistensi di lingkungan dengan banyak pengguna
- Biaya lisensi yang lebih tinggi atau penguncian vendor
- Dukungan dan sumber daya komunitas terbatas
- Potensi risiko keamanan dan kepatuhan

Opsi basis data vektor

AWS menawarkan beragam solusi basis data vektor untuk mendukung berbagai kasus penggunaan dan persyaratan dalam aplikasi AI generatif. Opsi ini dapat dikategorikan secara luas ke dalam layanan database individu dan penawaran layanan terkelola, masing-masing dengan karakteristik dan keunggulan yang berbeda. Memahami opsi ini sangat penting bagi organisasi yang ingin menerapkan kemampuan pencarian vektor secara efektif sambil mempertahankan kinerja, skalabilitas, dan efisiensi biaya yang optimal.

Untuk informasi selengkapnya tentang solusi database vektor, lihat bagian berikut:

- Opsi basis data vektor individu
- Opsi layanan terkelola
- Memilih database vektor yang tepat

Opsi basis data vektor individu

Opsi database vektor individual AWS termasuk Amazon Kendra, Amazon OpenSearch Service, AmazonRDS for PostgreSQL dengan pgvector, Amazon MemoryDB, AmazonDocumentDB, Amazon Neptune Analytics, dan Amazon S3 Vector. (Ekstensi sumber terbuka, pgvector menambahkan kemampuan untuk menyimpan dan mencari embeddings vektor yang dihasilkan ML.) Solusi ini menawarkan pendekatan yang berbeda untuk pencarian vektor, memungkinkan organisasi untuk memilih berdasarkan infrastruktur yang ada, persyaratan teknis, dan kasus penggunaan spesifik.

Amazon Kendra

Amazon Kendra adalah layanan pencarian cerdas tingkat perusahaan yang menggunakan pemrosesan bahasa alami dan algoritme pembelajaran mesin canggih untuk mengembalikan jawaban spesifik atas pertanyaan penelusuran dari data Anda. Amazon Kendra menyederhanakan implementasi fungsionalitas pencarian, menjadikannya solusi backend yang efektif untuk aplikasi AI generatif.

Fitur utama lainnya dari Amazon Kendra adalah sebagai berikut:

- Koneksi asli ke lebih dari 40 sumber data
- Kemampuan persiapan data bawaan

- Pengaturan cepat yang tidak memerlukan keahlian teknis yang mendalam

Manfaat Amazon Kendra meliputi yang berikut ini

- Pemrosesan data otomatis (chunking, konsumsi, pengambilan)
- Opsi kustomisasi yang kuat:
  - Pencarian facet
  - Analisis pencarian
  - Menyetel relevansi pencarian
- Akses terprogram sederhana melalui AWS SDK untuk Python (Boto3)

Untuk informasi lebih lanjut, lihat Manfaat Amazon Kendra di dokumentasi Amazon Kendra.

OpenSearch Layanan Amazon

Amazon OpenSearch Service adalah layanan terkelola yang membantu Anda menyebarkan, mengoperasikan, dan menskalakan kluster OpenSearch Layanan di. AWS Cloud

Kemampuan inti OpenSearch Layanan meliputi:

- Mesin pencari dan analitik sumber terbuka
- Arsitektur terdistribusi
- Pemrosesan data waktu nyata

Beberapa keuntungan menggunakan OpenSearch Layanan antara lain sebagai berikut:

- Skalabilitas horisontal
- RESTful Dukungan API
- Menangani data terstruktur dan tidak terstruktur
- Analisis data waktu nyata
- Cocokkan untuk berbagai ukuran penyebaran

Untuk informasi selengkapnya, lihat Fitur OpenSearch Layanan Amazon di dokumentasi OpenSearch Layanan.

Amazon RDS untuk PostgreSQL dengan pgvector

Amazon RDS for PostgreSQL dengan AWS pgvector menggabungkan layanan database relasional terkelola dengan ekstensi pemrosesan vektor PostgreSQL. Kombinasi ini memungkinkan organisasi untuk menyimpan dan menanyakan vektor dimensi tinggi sambil mempertahankan Amazon RDS. Solusi ini sangat cocok untuk aplikasi AI generatif yang memerlukan operasi vektor real-time tanpa overhead mengelola infrastruktur database.

Manfaat utama Amazon RDS untuk PostgreSQL dengan pgvector meliputi:

- Ketersediaan tinggi
- Failover otomatis
- Hemat biaya () pay-per-use
- Pemantauan bawaan
- Integrasi data vektor waktu nyata

Untuk informasi selengkapnya, lihat Keuntungan Amazon RDS dalam dokumentasi Amazon RDS.

Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah database dokumen yang menawarkan kemampuan pencarian vektor asli di versi 5.0 dan yang lebih baru. Ini menggabungkan fleksibilitas penyimpanan dokumen berbasis JSON dengan pencarian vektor, mendukung metode pengindeksan dunia kecil yang dapat dinavigasi hierarkis (HNSW) dan Inverted File Flat (). IVFFlat

Kemampuan inti Amazon DocumentDB meliputi yang berikut:

- Simpan dan indeks vektor hingga 2.000 dimensi (hingga 16.000 dimensi tanpa pengindeksan)
- Waktu respons milidetik untuk pencarian kesamaan vektor
- Support untuk metrik jarak produk euclidean, cosinus, dan dot
- Integrasi mulus dengan aplikasi yang kompatibel dengan MongoDB yang ada

Gunakan Amazon DocumentDB dalam situasi berikut:

- Untuk aplikasi yang sudah menggunakan APIs MongoDB dan yang membutuhkan kemampuan pencarian vektor

- Untuk kasus penggunaan yang memerlukan struktur data dokumen fleksibel yang dikombinasikan dengan pencarian semantik
- Untuk skenario yang membutuhkan kueri dokumen tradisional dan pencarian kesamaan vektor
- Untuk aplikasi yang memberikan rekomendasi produk, personalisasi, asisten obrolan, dan deteksi penipuan

Untuk informasi selengkapnya, lihat Pencarian vektor untuk Amazon DocumentDB di dokumentasi Amazon DocumentDB.

Amazon MemoryDB

Amazon MemoryDB adalah database dalam memori yang kompatibel dengan REDIS yang memberikan kinerja pencarian vektor tercepat di antara basis data vektor populer. AWS Ini menyediakan latensi kueri sub-milidetik dengan daya tahan Zona Ketersediaan Multi-ketersediaan.

Kemampuan inti dari MemoryDB meliputi:

- Simpan data aplikasi dan jutaan vektor dalam satu database
- Kueri milidetik satu digit dan perbarui waktu respons
- Tingkat penarikan tertinggi pada kinerja tercepat AWS
- Support hingga 32.768 dimensi per vektor
- Kemampuan pencarian dan caching semantik waktu nyata

Gunakan MemoryDB dalam situasi berikut:

- Untuk aplikasi real-time yang membutuhkan latensi ultra-rendah (sub-10ms)
- Untuk beban kerja throughput tinggi dengan jutaan permintaan per hari
- Untuk kasus penggunaan seperti mesin rekomendasi real-time, caching semantik, dan deteksi anomali
- Untuk aplikasi yang membutuhkan penyimpanan data dalam memori dan kemampuan pencarian vektor

Untuk informasi selengkapnya, lihat Pencarian vektor di dokumentasi MemoryDB.

Analisis Amazon Neptunus

Amazon Neptune Analytics adalah mesin analisis grafik yang menawarkan kemampuan pencarian vektor asli, membuatnya ideal untuk kasus penggunaan Graph Retrieval Augmented Generation (GraphRag). Ini menggabungkan pencarian kesamaan vektor dengan traversal grafik dan algoritma.

Kemampuan inti Neptunus Analytics meliputi:

- Menganalisis puluhan miliar hubungan dalam hitungan detik
- Gabungkan pencarian vektor dengan algoritma grafik (pencarian jalur, deteksi komunitas, sentralitas)
- Support untuk aplikasi GraphRag dengan pengetahuan topologi
- Hingga 80 kali lebih cepat dari solusi analisis grafik yang ada
- Integrasi dengan Amazon Bedrock untuk GraphRag yang dikelola sepenuhnya

Gunakan Neptune Analytics dalam situasi berikut:

- Untuk aplikasi GraphRag yang membutuhkan grafik pengetahuan dengan embeddings vektor
- Untuk kasus penggunaan yang memerlukan melintasi hubungan kompleks di samping kesamaan vektor
- Untuk aplikasi yang membutuhkan respons AI yang dapat dijelaskan dengan konteks hubungan
- Untuk skenario seperti tampilan 360 pelanggan, jaringan deteksi penipuan, dan penemuan pengetahuan

Untuk informasi selengkapnya, lihat dokumentasi Amazon Neptune Analytics.

Vektor Amazon S3

Amazon S3 Vectors adalah penyimpanan objek cloud pertama AWS dengan penyimpanan vektor asli dan kemampuan kueri. Ini menyediakan penyimpanan vektor yang dibuat khusus dan dioptimalkan biaya untuk aplikasi AI yang membutuhkan skala besar.

Kemampuan inti Vektor Amazon S3 meliputi yang berikut:

- Penyimpanan hingga 2 miliar vektor per indeks dengan dukungan hingga 10.000 indeks per bucket vektor

- Latensi kueri sub-100 ms yang dioptimalkan untuk penyimpanan jangka panjang dan pola akses yang jarang
- Pengurangan biaya hingga 90% untuk operasi vektor dibandingkan dengan database vektor khusus
- Arsitektur tanpa server dengan penskalaan otomatis dan daya tahan 99,999999999% (11 9s)

Gunakan Vektor Amazon S3 dalam situasi berikut:

- Untuk aplikasi yang membutuhkan penyimpanan miliaran vektor dengan biaya minimal
- Untuk beban kerja yang mentolerir latensi kueri sub-detik (100 ms atau lebih) daripada sub-10 ms
- Untuk retensi vektor jangka panjang dan kasus penggunaan arsip
- Untuk aplikasi RAG dengan pola pengambilan yang jarang
- Untuk organisasi yang memprioritaskan ekonomi penyimpanan di atas latensi ultra-rendah

Amazon S3 Vectors terintegrasi secara native dengan Amazon Bedrock Knowledge Bases dan bekerja dengan baik dalam arsitektur berjenjang dengan Amazon Service. OpenSearch Anda dapat menggunakan Vektor Amazon S3 untuk penyimpanan dingin dan menggunakan OpenSearch Layanan untuk kueri panas.

Untuk informasi selengkapnya, lihat Bekerja dengan Vektor S3 dan bucket vektor dalam dokumentasi Amazon S3.

Opsi layanan terkelola

Amazon Bedrock Knowledge Bases mewakili pendekatan yang dikelola AWS sepenuhnya untuk implementasi database vektor. Fleksibilitas layanan dalam opsi penyimpanan, dikombinasikan dengan fitur manajemen otomatisnya, membuatnya sangat berharga bagi organisasi yang ingin menerapkan RAG tanpa mengelola infrastruktur yang kompleks.

Dengan Pangkalan Pengetahuan Amazon Bedrock, Anda dapat membuat, memelihara, dan menanyakan basis pengetahuan yang menyempurnakan model fondasi Anda menggunakan RAG. Layanan ini menyederhanakan proses kompleks penerapan RAG dengan mengelola seluruh konsumsi data, vektorisasi, dan pipa pengambilan.

Manfaat utama dari Pangkalan Pengetahuan Amazon Bedrock meliputi:

- Pemrosesan data yang disederhanakan

  - Konsumsi dan chunking data otomatis
  - Ekstraksi teks bawaan dari berbagai format file
  - Generasi penyematan vektor terkelola
  - Ekstraksi dan pengindeksan metadata otomatis
- Implementasi RAG yang efisien
  - Strategi pengambilan yang telah dikonfigurasi sebelumnya
  - Optimalisasi jendela konteks otomatis
  - Penyetelan relevansi bawaan
  - Kemampuan pencarian semantik di luar kotak
- Keamanan dan tata kelola
  - Kontrol terintegrasi AWS Identity and Access Management (IAM)
  - Enkripsi data saat istirahat dan dalam perjalanan
  - Dukungan VPC
  - Audit logging dengan AWS CloudTrail

Amazon Bedrock Knowledge Bases mendukung beberapa opsi penyimpanan vektor, termasuk:

- Amazon Aurora PostgreSQL dengan pgvector
- Analisis Amazon Neptunus
- Amazon EMR Tanpa Server
- Vektor Amazon S3
- Biji pinus
- Awan Perusahaan Redis

Layanan terkelola ini menangani konsumsi data otomatis, vektorisasi, dan pengambilan. Ini menyederhanakan implementasi RAG.

Untuk informasi mendetail tentang setiap penyimpanan vektor yang didukung, lihat dokumentasi Pangkalan Pengetahuan Amazon Bedrock.

Memilih database vektor yang tepat

Pilih database vektor Anda berdasarkan faktor keputusan utama ini:

- Jika Anda memerlukan database dokumen yang kompatibel dengan MongoDB dengan pencarian vektor - Pilih Amazon DocumentDB. Ini sangat ideal ketika aplikasi Anda menggunakan APIs MongoDB dan Anda ingin menambahkan kemampuan pencarian semantik tanpa mengelola infrastruktur vektor terpisah.
- Jika Anda membutuhkan latensi ultra-rendah untuk aplikasi waktu nyata - Pilih Amazon MemoryDB. Ini memberikan kinerja pencarian vektor tercepat AWS dengan waktu respons sub-milidetik. Ini ideal untuk mesin rekomendasi real-time dan aplikasi throughput tinggi.
- Jika Anda memerlukan representasi pengetahuan berbasis grafik dengan pencarian vektor — Pilih Amazon Neptune Analytics. Ini terbaik untuk aplikasi GraphRag yang perlu melintasi hubungan kompleks dan melakukan kueri berbasis grafik di samping pencarian vektor, memberikan respons AI yang dapat dijelaskan.
- Jika Anda perlu menggabungkan kueri relasional dengan pencarian vektor — Pilih Amazon Aurora PostgreSQL dengan pgvector. Opsi ini sangat ideal ketika aplikasi Anda memerlukan operasi SQL tradisional dan pencarian kesamaan vektor dalam database yang sama.
- Jika Anda memerlukan kueri throughput tinggi dengan latensi sub-10 ms — Pilih Layanan Amazon. OpenSearch Ini unggul dalam menangani kueri frekuensi tinggi dan aplikasi real-time dan mencakup peningkatan akselerasi GPU baru-baru ini.
- Jika Anda perlu menyimpan miliaran vektor dengan biaya efektif — Pilih Vektor Amazon S3. Opsi ini memberikan penghematan biaya hingga 90% dan sangat ideal untuk aplikasi dengan pola pengambilan yang jarang (menit hingga jam di antara kueri) yang dapat mentolerir latensi sub-100 ms.
- Jika Anda memerlukan pencarian teks lengkap bersama pencarian vektor — Pilih OpenSearch Layanan Amazon. Opsi ini menggabungkan kemampuan pencarian teks lengkap yang kuat dengan pencarian vektor dalam satu platform.

Perbandingan basis data vektor

AWS menyediakan beberapa pendekatan untuk menerapkan kemampuan pencarian vektor, mulai dari database vektor individual hingga Pangkalan Pengetahuan Amazon Bedrock, yang merupakan layanan yang dikelola sepenuhnya. Saat mengevaluasi opsi ini, organisasi harus mempertimbangkan berbagai aspek termasuk arsitektur, skalabilitas, kemampuan integrasi, karakteristik kinerja, dan fitur keamanan.

Database vektor individu

Tabel berikut memberikan ikhtisar fitur utama dari beberapa solusi database vektor AWS individu, dengan fokus pada arsitekturnya, kemampuan penskalaan, integrasi sumber data, dan karakteristik kinerja.

Fitur	Amazon Kendra	OpenSearch Layanan Amazon	Amazon RDS untuk SQLwith Postgre pgvector	Amazon DocumentDB	Amazon MemoryDB	Analisis Amazon Neptunus	Vektor Amazon S3
Kasus penggunaan primer	Pencarian perusahaan dan RAG	Pencarian dan analitik terdistribusi	Relasional DB dengan dukungan vektor	Dokumen DB dengan pencarian vektor	Pencarian vektor dalam memori waktu nyata	Analisis grafik dengan pencarian vektor	Penyimpanan vektor yang dioptimalkan biaya
Arsitektur	Dikelola sepenuhnya	Cluster terdistribusi	Basis data relasional	Berorientasi dokumen	Basis data dalam memori	Mesin analisis grafik	Penyimpanan objek tanpa server
Model Data	Berbasis dokumen	Dokumen JSON	Tabel relasional	Dokumen JSON	Nilai kunci dengan JSON	Grafik properti	Penyimpanan objek

Dimensi vektor	Dikelola secara otomatis	Hingga 16.000	Dapat dikonfigurasi	Hingga 2.000 (diindeks); 16.000 (tidak diindeks)	Hingga 32.768	Dapat dikonfigurasi	Hingga 4.096
Metode pengindeksan	Otomatis	HNSW, IVF	HNSW, IVFFlat	HNSW, IVFFlat	HNSW	Grafik dan vektor asli	Otomatis
Metrik jarak	Otomatis	Cosine, Euclidean, produk dot	Cosine, Euclidean, produk dalam	Cosine, Euclidean, produk dot	Cosine, Euclidean, produk dalam	Kosinus, Euclidean	Kosinus, Euclidean
Latensi kueri	Sub-detik	Sub-10 ms (dipercepat GPU)	10-100 ms	Milidetik	Sub-milidetik	Sub-detik	Sub-100 ms
Model penskalaan	Otomatis	Horisontal (tambahkan node)	Replika vertikal dan baca	Horizontal (tambahkan contoh)	Vertikal dan replika	Otomatis	Otomatis (tanpa server)
Vektor maksimum	Dikelola	Miliaran (tergantung cluster)	Jutaan (tergantung instans)	Jutaan per koleksi	Jutaan per database	Miliaran	2 miliar per indeks; 10.000 indeks per ember

Throughput	Tinggi	Sangat tinggi (ribuan QPS)	Sedang	Tinggi	Sangat tinggi (jutaan permintaan per hari)	Tinggi	Medium (dioptimalkan untuk kueri yang jarang)
Daya tahan data	99,999999999% (11 9 detik)	Dapat dikonfigurasi dengan replika	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99%	99,999999999% (11 9 detik)
Model konsistensi	Akhirnya	Akhirnya (dapat dikonfigurasi)	Kuat (ACID)	Akhirnya	Kuat	Kuat	Kuat
Kemampuan tambahan	40 atau lebih konektor data, NLP	Pencarian teks lengkap, analitik, dasbor	Kueri SQL, transaksi ACID	Kompatibilitas API MongoDB	Kompatibilitas Redis API, caching	Algoritma grafik, traversal	Integrasi Amazon S3, kebijakan siklus hidup
Model penentuan harga	Bayar per kueri dan penyimpanan	Jam dan penyimpanan instans	Jam dan penyimpanan instans	Jam dan penyimpanan instans	Jam dan penyimpanan instans	Unit kapasitas dan penyimpanan	Penyimpanan, kueri, dan transfer data

Optimalisasi biaya	Berbasis penggunaan	Contoh cadangan, auto-scaling	Contoh yang dicadangkan, Aurora Tanpa Server	Instans cadangan	Instans cadangan	Penskalaan otomatis	Penghematan hingga 90% vs khusus DBs
Terbaik untuk	Pencarian perusahaan dengan pengaturan minimal	Kueri throughput tinggi, latensi rendah	Hybrid SQL dan beban kerja vektor	Aplikasi yang kompatibel dengan MongoDB yang membutuhkan vektor	Aplikasi real-time, latensi ultra-rendah	GraphRag dan grafik pengetahuan	Penyimpanan jangka panjang dan hemat biaya
Pola kueri yang ideal	Pencarian perusahaan yang sering	Kueri real-time frekuensi tinggi	SQL campuran dan kueri vektor	Kueri dokumen dengan pencarian semantik	Jutaan permintaan per hari	Grafik traversal dengan pencarian vektor	Pertanyaan yang jarang terjadi (menit hingga jam)
Kompleksitas pengaturan	Rendah (dikelola sepenuhnya)	Sedang (konfigurasi cluster)	Sedang (pengaturan ekstensi)	Sedang (konfigurasi cluster)	Sedang (konfigurasi cluster)	Rendah (dikelola sepenuhnya)	Rendah (tanpa server)
Diperlukan keahlian tim	Minimal	OpenSearch atau Elasticsearch	PostgreSQL, SQL	MongoDB	Redis	Database grafik	Amazon S3, konsep vektor dasar

Layanan terkelola - Basis Pengetahuan Amazon Bedrock

Amazon Bedrock Knowledge Bases menyediakan solusi yang dikelola sepenuhnya dengan beberapa opsi penyimpanan vektor. Tabel berikut membandingkan opsi penyimpanan ini.

Fitur	Aurora Postgre pgvektor SQLwith	Analisis Neptunus	OpenSearc h Layanan Tanpa Server	Vektor Amazon S3	Biji pinus	RedisEnte rprise Awan
Kasus penggunaa n primer	Relasional DB dengan vektor RAG	Pencarian vektor berbasis grafik untuk GraphRag	Manajemen pengetahu an RAG	Vektor RAG yang dioptimal kan biaya	Pencarian vektor kinerja tinggi	Pencarian vektor dalam memori
Arsitektur	Relasiona l yang dikelola sepenuhny a	Analisis grafik yang dikelola sepenuhny a	Tanpa server yang dikelola sepenuhny a	Penyimpan an objek tanpa server	Cloud hybrid yang dikelola sepenuhny a	Dikelola sepenuhny a dalam memori
Model Data	Tabel relasional	Grafik properti	Dokumen JSON	Penyimpan an objek	Vektor yang dibuat khusus	Nilai kunci dengan vektor
Penyimpan an vektor	Melalui ekstensi pgvector	Vektor grafik asli	Melalui OpenSearc h mesin	Penyimpan an vektor Amazon S3 asli	Database vektor asli	Vektor dalam memori
Integrasi Amazon Bedrock	Asli	Asli	Asli	Asli	Asli	Asli

Konsumsi otomatis	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)
Vektorisasi otomatis	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)	Ya (melalui Amazon Bedrock)
Penskalaan	Penskalaan otomatis (Aurora Tanpa Server)	Penskalaan grafik otomatis	Otomatis tanpa server	Otomatis (miliaran vektor)	Pod penskalaan otomatis	Cluster penskalaan otomatis
Kinerja kueri	Tinggi untuk relasional atau vektor	Tinggi untuk vektor grafik	Tinggi	Sedang (100 ms atau lebih latensi)	Sangat tinggi	Sangat tinggi
Vektor maksimum	Jutaan (tergantung instans)	Miliaran	Miliaran	2 miliar per indeks	Miliaran	Jutaan (tergantung memori)
Kemampuan tambahan	Kueri SQL, transaksi ACID	Algoritma grafik, traversal	Pencarian teks lengkap, analitik	Siklus hidup Amazon S3, tingkat	Pemfilteran metadata, ruang nama	Struktur data Redis, caching
Optimalisasi biaya	Moderat (Aurora Tanpa Server)	Sedang (unit kapasitas)	Tinggi (tanpa server,) pay-per-use	Sangat tinggi (penghematan hingga 90%)	Sedang (harga berbasis pod)	Rendah (premium dalam memori)

Terbaik untuk	Beban SQL/vector kerja hibrida	Grafik pengetahuan yang terhubung	Teks lengkap dengan pencarian vektor	Vektor jangka panjang dan jarang diakses	Pencarian vektor real-time pada skala	Kebutuhan latensi sangat rendah
Pola kueri yang ideal	SQL campuran dan kueri vektor	Grafik traversal dengan vektor	Pencarian yang sering dilakukan dengan analitik	Pengambilan yang jarang (menit hingga jam)	Kueri real-time frekuensi tinggi	Jutaan permintaan per detik
Pengaturan dengan Amazon Bedrock	Sederhana (dikelola oleh Amazon Bedrock)	Sederhana (dikelola oleh Amazon Bedrock)	Sederhana (dikelola oleh Amazon Bedrock)	Sederhana (dikelola oleh Amazon Bedrock)	Sederhana (dikelola oleh Amazon Bedrock)	Sederhana (dikelola oleh Amazon Bedrock)
Data residensi	Wilayah AWS	Wilayah AWS	Wilayah AWS	Wilayah AWS	Multi-cloud (AWS dan lainnya)	Multi-cloud (AWS dan lainnya)
Model penentuan harga	Jam dan penyimpanan instans	Unit kapasitas dan penyimpanan	Komputasi dan penyimpanan (tanpa server)	Penyimpanan, kueri, dan transfer	Jam dan penyimpanan pod	Jam dan penyimpanan node

Memilih antara opsi individu dan terkelola

Pertimbangan	Pilih vektor individu DB	Pilih Pangkalan Pengetahuan Amazon Bedrock (dikelola)

Implementasi RAG	Anda ingin kontrol penuh atas pipa RAG	Anda ingin RAG yang dikelola sepenuhnya dengan pengaturan minimal
Kustomisasi	Anda memerlukan logika pengambilan khusus dan preprocessing	Pola RAG standar memenuhi kebutuhan Anda
Infrastruktur yang ada	Anda sudah memiliki database yang digunakan	Anda mulai segar atau ingin manajemen yang disederha nakan
Keahlian tim	Tim Anda memiliki keahlian administrasi database	Anda lebih suka fokus pada logika aplikasi, bukan infrastru ktur
Kompleksitas integrasi	Anda membutuhkan integrasi mendalam dengan sistem yang ada	Anda ingin integrasi cepat dengan model Amazon Bedrock
Overhead operasional	Anda dapat mengelola operasi database	Anda AWS ingin menangani operasi
Struktur biaya	Anda lebih suka harga database langsung	Anda lebih suka harga Amazon Bedrock terpadu
Waktunya ke pasar	Anda punya waktu untuk implementasi khusus	Anda membutuhkan penyebaran yang cepat

Perbandingan biaya dan pertimbangan

Memahami struktur biaya opsi database vektor yang berbeda sangat penting untuk membuat keputusan implementasi yang tepat. Tabel berikut menguraikan beberapa pertimbangan biaya utama untuk berbagai solusi database vektor, termasuk database individu dan layanan terkelola. Setiap opsi memiliki faktor penetapan harga yang berbeda yang dapat memengaruhi total biaya kepemilikan (TCO), dari pay-as-you-go model hingga biaya infrastruktur dan operasional.

Database vektor	Model biaya	Pertimbangan biaya
Amazon Kendra	Bayar saat Anda pergi, berdasarkan pertanyaan	Biaya dapat bervariasi berdasarkan jumlah kueri dan jumlah data yang diindeks. Biaya tambahan dapat berlaku untuk penyimpanan data dan transfer data. Untuk informasi lebih lanjut, lihat harga Amazon Kendra.
OpenSearch Layanan Amazon	Bayar saat Anda pergi, berdasarkan jam dan penyimpanan instans	Biaya termasuk jam instans, penyimpanan (volume Amazon EBS), transfer data, dan UltraWarm penyimpanan opsional. Instans Cadangan dapat memberikan penghemat an hingga 30%. Instans yang dipercepat GPU menawarka n kinerja harga yang lebih baik untuk beban kerja vektor. Untuk informasi selengkap nya, lihat harga OpenSearch Layanan Amazon.
Sumber terbuka OpenSearch	Sumber terbuka, tanpa biaya langsung (Anda tidak perlu membayar untuk mengunduh atau menggunakan perangkat	Biaya termasuk infrastruktur (seperti server dan penyimpan an) dan biaya operasional (seperti pemeliharaan dan

	lunak, dan tidak ada biaya lisensi)	pemantauan). Organizat ions perlu menganggarkan personel untuk mengelola dan memelihara infrastruktur.
Amazon RDS untuk PostgreSQL dengan pgvector	Bayar saat Anda pergi, berdasarkan penggunaan	Biaya termasuk jenis instance database, penyimpanan, transfer data, dan backup. Biaya tambahan dapat berlaku untuk transfer data, jenis instans, dan penyimpanan di luar Tingkat AWS Gratis. Untuk informasi selengkapnya, lihat Harga Amazon RDS.
Amazon DocumentDB	Bayar saat Anda pergi, berdasarkan jam dan penyimpanan instans	Biaya termasuk jam instans, penyimpanan (GB-bulan), I/ O permintaan, penyimpanan cadangan, dan transfer data. Cluster elastis memungkinkan penskalaan dinamis. Instans Cadangan tersedia untuk pengoptimalan biaya. Untuk informasi selengkapnya, lihat harga Amazon DocumentDB.
Amazon MemoryDB	Bayar saat Anda pergi, berdasarkan jam simpul dan penyimpanan data	Biaya termasuk jam node (per tipe node), penyimpan an data (GB-jam), penyimpan an snapshot, dan transfer data. Node Cadangan dapat memberikan penghematan hingga 55%. Dioptimalkan untuk throughput tinggi, beban kerja latensi rendah. Untuk informasi selengkapnya, lihat harga Amazon MemoryDB.

Analisis Amazon Neptunus	Bayar saat Anda pergi, berdasarkan unit kapasitas	Biaya termasuk Unit Kapasitas Neptunus NCUs (), penyimpanan (GB-bulan), dan transfer data. Penskalaan otomatis berdasarkan beban kerja tanpa komitmen di muka. Minimal 128 NCUs diperlukan. Untuk informasi selengkapnya, lihat harga Amazon Neptunus.
Vektor Amazon S3	Bayar saat Anda pergi, berdasarkan penyimpanan dan permintaan	Biaya termasuk penyimpanan (GB-bulan), permintaan PUT dan GET, manajemen indeks vektor, dan transfer data. Memberikan penghematan biaya hingga 90% dibandingkan dengan database vektor khusus. Kebijakan Tingkat Cerdas dan siklus hidup Amazon S3 tersedia untuk pengoptimalan tambahan. Untuk informasi selengkapnya, lihat Harga Amazon S3.
Basis Pengetahuan Amazon Bedrock	Bayar saat Anda pergi, berdasarkan penggunaan	Biaya dapat bervariasi berdasarkan penggunaan basis pengetahuan dan layanan tambahan seperti Amazon Tanpa OpenSearch Server. Biaya tambahan dapat berlaku untuk penyimpanan data, transfer data, dan fitur tambahan. Untuk informasi selengkapnya tentang harga, lihat harga OpenSearch Layanan Amazon.

Kasus penggunaan basis data vektor

Contoh berikut menyoroti bagaimana opsi database vektor yang berbeda dapat digunakan secara efektif untuk meningkatkan manajemen pengetahuan, meningkatkan efisiensi operasional, dan memberikan hasil bisnis yang lebih baik. Kasus penggunaan ini menggambarkan aplikasi praktis dari solusi basis data vektor yang dibahas sebelumnya dalam panduan ini dan memberikan wawasan tentang kinerja dan manfaat dunia nyata mereka.

Manajemen pengetahuan dengan Amazon Kendra

Masalah pelanggan — Salah satu kontraktor umum terbesar di Jepang menghadapi penurunan personel berpengalaman. Perusahaan membutuhkan cara untuk mentransfer pengetahuan dan keterampilan personel pengalaman kepada generasi muda secara efisien. Mereka membutuhkan solusi untuk menangkap dan menyebarkan pengetahuan teknik konstruksi yang kompleks dan pengalaman masa lalu.

AWS Solusi — Untuk mengatasi masalah ini, pelanggan beralih ke Amazon Kendra, solusi AI yang dapat dengan cepat dan akurat menangani basis pengetahuan internal mereka dan memungkinkan kueri bahasa alami. Dengan Amazon Kendra, karyawan sekarang dapat menemukan informasi yang mereka butuhkan lebih cepat, meningkatkan produktivitas dan memfasilitasi transfer pengetahuan dari personel berpengalaman ke staf yang lebih muda.

Dampak — Dengan menerapkan chatbot AI generatif yang didukung oleh Amazon Kendra, perusahaan menciptakan platform pengetahuan terpadu. Chatbot memungkinkan karyawan untuk dengan cepat mengakses pengetahuan teknis dan pengalaman masa lalu tentang teknik konstruksi. Solusi ini telah secara signifikan meningkatkan efisiensi transfer pengetahuan dan proses pengambilan keputusan dalam organisasi, membantu memastikan bahwa keahlian yang berharga dipertahankan dan mudah diakses oleh semua karyawan. Biaya solusi ini dapat bervariasi tergantung pada penggunaan dan konfigurasi Anda. Untuk perkiraan biaya terperinci, lihat AWS Kalkulator Harga. Untuk estimasi biaya basis data vektor, lihat bagian Perbandingan biaya dan pertimbangan pada panduan ini atau lihat harga Amazon Kendra.

Untuk informasi tentang kasus penggunaan pelanggan lainnya, lihat pelanggan Amazon Kendra.

Analitik real-time dengan Tanpa OpenSearch Server

Masalah pelanggan — Penyedia layanan keuangan terkemuka menghadapi tantangan dalam mengelola ekosistem data yang sangat besar. Ini memproses 300 juta otorisasi dan 90 miliar

transaksi setiap tahun, mengumpulkan sekitar 1,1 petabyte (PB) data. Sistem yang ada, melayani 300.000 pengguna yang membutuhkan akses ke lebih dari 6.000 laporan, membutuhkan modernisasi untuk memberikan konsistensi global dan memungkinkan pengambilan keputusan secara real-time.

AWS solusi — Arsitektur solusi menggunakan model dasar yang tersedia melalui Amazon Bedrock (termasuk Anthropic, Sonnet 3, Sonnet 3.5, dan Haiku) untuk pemrosesan bahasa alami. Pelanggan memilih OpenSearch Tanpa Server sebagai basis data vektor karena skalabilitas dan kemampuannya yang unggul untuk menangani volume data yang besar secara efisien. Arsitektur ini memungkinkan pemrosesan kueri kompleks dan pembuatan laporan dinamis tanpa batas.

Dampak — Implementasi mencapai peningkatan produktivitas sebesar 50 persen dengan menghilangkan kebutuhan akan pembuatan manual lebih dari 100 dasbor intelijen bisnis. Pengguna sekarang dapat menghasilkan laporan melalui kueri bahasa alami dengan waktu respons antara 20-40 detik. Biaya solusi ini dapat bervariasi tergantung pada penggunaan dan konfigurasi Anda. Untuk perkiraan biaya terperinci, lihat AWS Kalkulator Harga. Untuk estimasi biaya basis data vektor, lihat bagian Perbandingan biaya dan pertimbangan pada panduan ini atau lihat harga OpenSearch Layanan Amazon. Untuk informasi tentang kasus penggunaan pelanggan lainnya, lihat Amazon Tanpa OpenSearch Server.

Langkah dan sumber daya selanjutnya

Setelah meninjau panduan ini, pertimbangkan tindakan berikut untuk beralih dari pemahaman ke implementasi:

1. Evaluasi kebutuhan Anda saat ini:
   - Menilai infrastruktur database yang ada dan keahlian.
   - Dokumentasikan persyaratan pencarian vektor spesifik Anda.
   - Tentukan kinerja, penskalaan, dan target biaya Anda.
2. Pilih salah satu opsi berikut untuk menguji opsi basis data vektor:
   - Opsi 1: Siapkan bukti konsep menggunakan solusi basis data vektor pilihan Anda.
   - Opsi 2: Bereksperimenlah dengan kumpulan data sampel di Pangkalan Pengetahuan Amazon Bedrock. Cobalah pengalaman cepat untuk Basis Pengetahuan Amazon Bedrock. Sebagai contoh, lihat Cepat membuat Pangkalan Pengetahuan PostgreSQL Aurora untuk Amazon Bedrock di dokumentasi Aurora.
3. Tinjau sumber daya tambahan.
4. Dapatkan bantuan ahli:
   - Hubungi Akun AWS tim atau Arsitek AWS Solusi Anda untuk panduan implementasi.
   - Terlibat dengan AWS Mitra yang berspesialisasi dalam database vektor.
5. Rencanakan penyebaran produksi Anda:
   - Buat strategi migrasi jika berpindah dari database yang ada.
   - Kembangkan rencana penskalaan untuk solusi pilihan Anda.
   - Rancang prosedur pemantauan dan pemeliharaan Anda.

Sumber daya

Sumber daya berikut dapat membantu Anda dalam memilih database vektor.

AWS posting blog

- Percepat pengembangan aplikasi AI generatif Anda dengan Amazon Bedrock Knowledge Bases Quick Create dan Amazon Aurora Tanpa Server
- Kemampuan basis data vektor Amazon OpenSearch Service dijelaskan

- Selami jauh ke dalam penyimpanan data vektor menggunakan Pangkalan Pengetahuan Amazon Bedrock
- Manfaatkan pgvector dan Amazon Aurora PostgreSQL untuk Pemrosesan Bahasa Alami, Chatbots, dan Analisis Sentimen

AWS dokumentasi layanan

- Memilih layanan AWS basis data
- Cara kerja basis pengetahuan Amazon Bedrock
- Dokumentasi Neptunus Analytics
- Ikhtisar Amazon Web Services: Database
- Menggunakan Aurora PostgreSQL sebagai Basis Pengetahuan untuk Amazon Bedrock
- Bekerja dengan Amazon Aurora PostgreSQL
- Amazon DocumentDB
- Amazon MemoryDB
- Vektor Amazon S3

AWS Sumber daya lainnya

- Basis Pengetahuan Amazon Bedrock
- Database Vektor & Embeddings
- Database Vektor untuk aplikasi AI generatif
- Apa itu Embeddings dalam Machine Learning?

Sumber daya lainnya

- Tentang PostgreSQL
- dokumentasi pgvector
- Biji Pinus sebagai Basis Pengetahuan untuk Amazon Bedrock
- Redis Enterprise Cloud aktif AWS

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini, Memilih database AWS vektor untuk kasus penggunaan RAG. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Ditambahkan Layanan AWS	Kami menambahkan informasi tentang penggunaan Amazon DocumentDB, Amazon MemoryDB, Amazon S3 Vektor, dan Amazon Neptune Analytics.	Maret 30, 2026
Publikasi awal	—	6 Maret 2025

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.