Auswahl einer AWS Vektordatenbank für RAG-Anwendungsfälle

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Auswahl einer AWS Vektordatenbank für RAG-Anwendungsfälle

Table of Contents

Auswahl einer AWS Vektordatenbank für RAG-Anwendungsfälle

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

März 2026 (Geschichte der Dokumente)

Vektordatenbanken werden für Unternehmen, die generative KI-Anwendungen implementieren, immer wichtiger. In diesen Datenbanken werden Vektoren gespeichert und verwaltet. Dabei handelt es sich um numerische Repräsentationen von Daten, die die Verarbeitung von Text, Bildern und anderen Inhalten auf eine Weise ermöglichen, die deren Bedeutung und Beziehungen erfasst.

Wenn Unternehmen die Optionen für Vektordatenbanken untersuchen AWS, müssen sie sich mit den Funktionen, Kompromissen und bewährten Methoden für verschiedene Lösungen vertraut machen. Dieser Leitfaden hilft Ihnen dabei, häufig verwendete Vektorspeicher zu vergleichen AWS und fundierte Entscheidungen darüber zu treffen, welche Optionen Ihren spezifischen Anforderungen oder Ihrem Anwendungsfall am besten entsprechen. Ganz gleich, ob Sie Retrieval Augmented Generation (RAG) implementieren, Empfehlungssysteme erstellen oder andere KI-Anwendungen entwickeln, dieser Leitfaden bietet einen Rahmen, der Ihnen bei der Bewertung und Auswahl einer Vektordatenbanklösung hilft.

Zielgruppe

Dieser Leitfaden richtet sich an Personen in den folgenden Rollen:

- Datenwissenschaftler und Ingenieure für maschinelles Lernen (ML), die Vektordatenbanken verwenden, um hochdimensionale Daten für ML-Modelle zu speichern und abzurufen.
- Dateningenieure, die Datenpipelines entwerfen und implementieren, die Vektordatenbanken zum Speichern und Verarbeiten hochdimensionaler Daten beinhalten.
- MLOps Ingenieure, die Vektordatenbanken als Teil der ML-Pipeline zum Speichern und Bereitstellen von Modellausgaben oder Zwischendarstellungen verwenden.
- Softwareingenieure, die Vektordatenbanken in Anwendungen integrieren, für die Ähnlichkeitssuche oder Empfehlungssysteme erforderlich sind.
- DevOps Ingenieure, die für die Bereitstellung und Wartung von Vektordatenbanken in Produktionsumgebungen verantwortlich sind und dabei Skalierbarkeit und Zuverlässigkeit sicherstellen.

- KI-Forscher, die Vektordatenbanken verwenden, um große Datensätze von Einbettungen oder Merkmalsvektoren zu speichern und zu analysieren.
- KI-Produktmanager, die die Möglichkeiten und Grenzen von Vektordatenbanken verstehen müssen, um fundierte Entscheidungen über Produktmerkmale und Architektur treffen zu können.

Überblick über Vektoren

Vektoren sind numerische Darstellungen, die Maschinen helfen, Daten zu verstehen und zu verarbeiten. In der generativen KI dienen sie zwei Hauptzwecken:

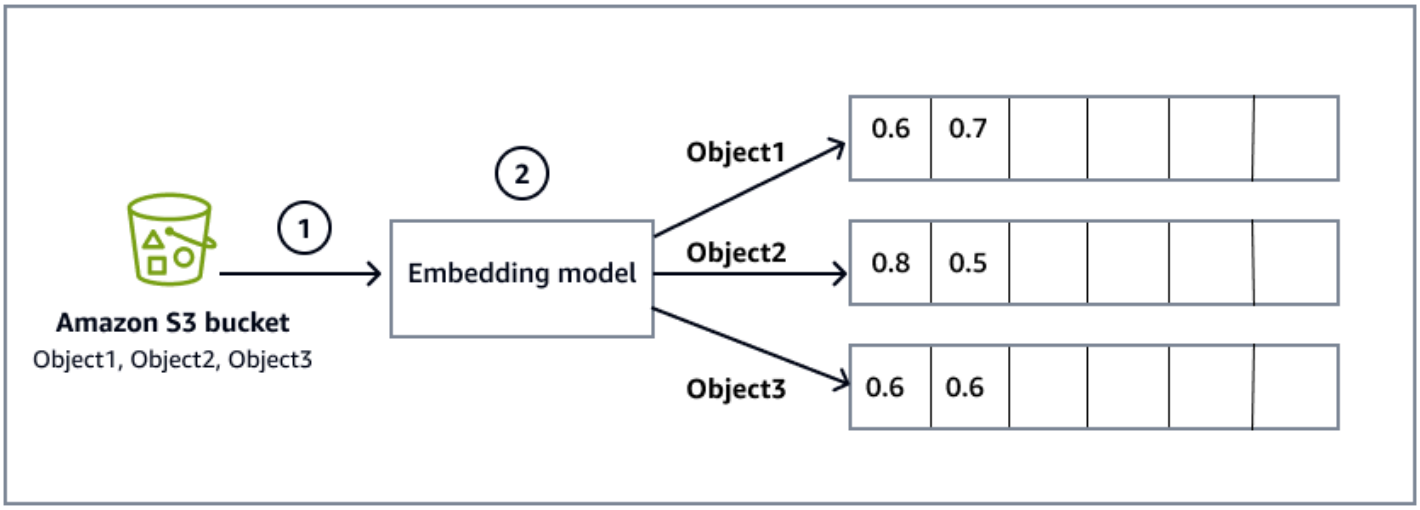
- Darstellung latenter Räume, die Datenstrukturen in komprimierter Form erfassen
- Erstellen von Einbettungen für Daten wie Wörter, Sätze und Bilder

Einbettungsmodelle wie Word2Vec und Amazon Titan Text Embeddings konvertieren Daten mithilfe eines Prozesses GloVe, der als Einbetten bezeichnet wird, in Vektoren. Diese Einbettungsmodelle können Folgendes bewirken:

- Lernen Sie aus dem Kontext, Wörter als Vektoren darzustellen
- Platzieren Sie ähnliche Wörter im Vektorraum näher beieinander
- Ermöglichen Sie Maschinen, Daten in einem kontinuierlichen Raum zu verarbeiten

Das folgende Diagramm bietet einen allgemeinen Überblick über den Einbettungsprozess:

1. Ein Amazon Simple Storage Service (Amazon S3) -Bucket enthält Dateien, die Datenquellen sind, aus denen das System Informationen liest und verarbeitet. Der Amazon S3 S3-Bucket wird während der Konfiguration der Amazon Bedrock-Wissensdatenbank angegeben, was auch die Synchronisierung von Daten mit der Wissensdatenbank beinhaltet.
2. Das Einbettungsmodell konvertiert die Rohdaten aus den Objektdateien im Amazon S3 S3-Bucket in Vektoreinbettungen. Wird beispielsweise in einen Vektor `[0.6, 0.7, ...]` umgewandelt, `Object1` der seinen Inhalt in einem mehrdimensionalen Raum darstellt.



Worteinbettungen sind für die Verarbeitung natürlicher Sprache (NLP) von entscheidender Bedeutung, da sie Folgendes bewirken:

- Erfassen Sie semantische Beziehungen zwischen Wörtern
- Ermöglichen Sie die Generierung von kontextrelevantem Text
- Unterstützen Sie große Sprachmodelle (LLMs), um menschenähnliche Antworten zu erzeugen

Überblick über Vektordatenbanken

Eine Vektordatenbank ist ein spezialisiertes System, das hochdimensionale Vektoren effizient speichert und abfragt. Diese Datenbanken sind von grundlegender Bedeutung für RAG-Anwendungen (Retrieval Augmented Generation).

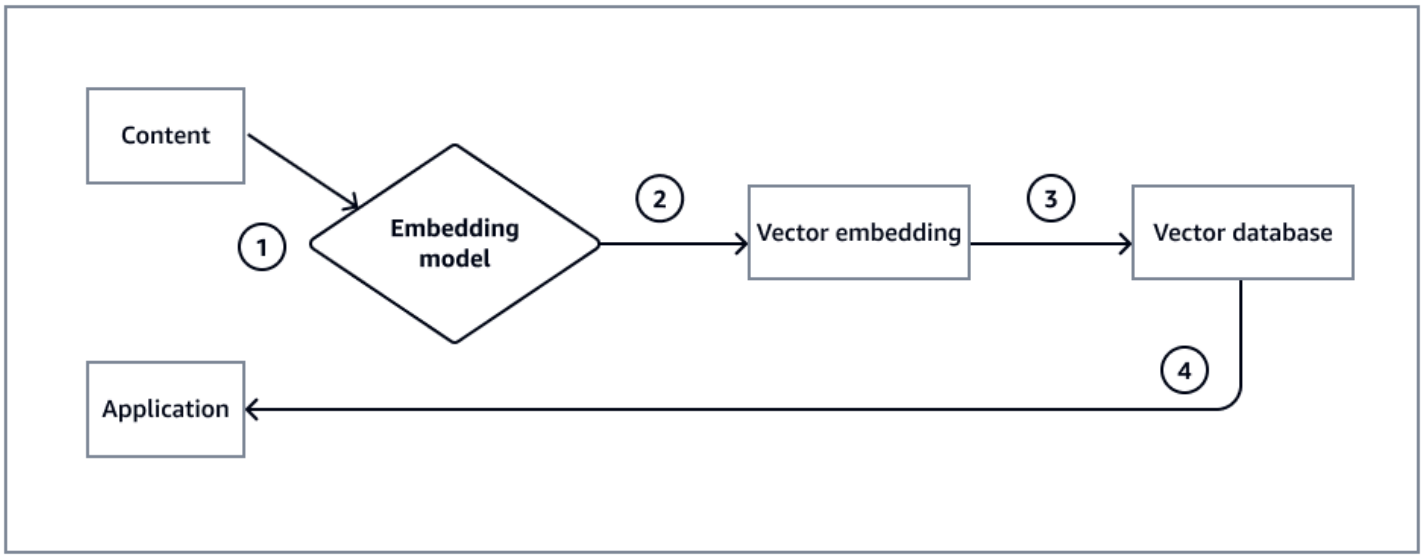
Vektordatenbanken handhaben die Datenkonvertierung und -speicherung auf folgende Weise:

- Objekte (wie Audio-, Bild- und Textdateien) werden mithilfe von Einbettungsmodellen in Vektoren konvertiert.
- Vektoren werden in speziellen Datenformaten gespeichert.
- Vektordatenbanken ermöglichen schnelle Ähnlichkeitssuchen.

Vektordatenbanken bieten mehrere wichtige Vorteile gegenüber herkömmlichen Datenbanken und eignen sich daher besonders gut für moderne Datenherausforderungen. Sie sind speziell für Vektoroperationen optimiert und verarbeiten hochdimensionale Daten effizient. Sie sind auch auf Ähnlichkeitssuchen spezialisiert, mit denen herkömmliche Datenbanken zu kämpfen haben. Neben diesen Kernfunktionen sind Vektordatenbanken darauf ausgelegt, den sich wandelnden Anforderungen von ML- und generativen KI-Anwendungen gerecht zu werden. Sie zeichnen sich durch großflächige Vektorspeicher aus und nutzen verteiltes Computing, um Workloads auf mehrere Knoten zu verteilen. Dies bietet Skalierbarkeit und Leistung bei wachsenden Datenmengen.

Das folgende Diagramm zeigt eine RAG-Implementierung:

1. Inhalte, wie Dokumente, PDFs oder Textdateien, werden als Rohdaten zur Verarbeitung in das Einbettungsmodell eingespeist.
2. Das Einbettungsmodell wandelt die Rohdaten in numerische Vektoren um, die die semantische Bedeutung des Inhalts darstellen.
3. Die generierten Vektoreinbettungen werden in einer Vektordatenbank gespeichert, die für das Speichern und Abrufen von hochdimensionalen Vektoren optimiert ist.
4. Anwendungen können nun als Reaktion auf Anwendungsfälle wie semantische Suche und Inhaltsempfehlungen die Vektordatenbank abfragen.



Die Auswahl einer ungeeigneten Vektordatenbank für eine RAG-Lösung kann zu erheblichen Problemen und Einschränkungen führen, darunter die folgenden:

- Schlechte Abfrageleistung
- Engpässe bei der Skalierbarkeit
- Herausforderungen bei der Datenaufnahme
- Fehlende erweiterte Funktionen wie Filterung und Rangfolge
- Schwierigkeiten bei der Integration mit anderen Systemen
- Bedenken hinsichtlich Beständigkeit und Dauerhaftigkeit
- Gleichzeitigkeits- und Konsistenzprobleme in Umgebungen mit mehreren Benutzern
- Höhere Lizenzkosten oder Bindung an einen bestimmten Anbieter
- Eingeschränkte Unterstützung und Ressourcen durch die Community
- Potenzielle Sicherheits- und Compliance-Risiken

Vektor-Datenbankoptionen

AWS bietet eine breite Palette von Vektordatenbanklösungen zur Unterstützung verschiedener Anwendungsfälle und Anforderungen in generativen KI-Anwendungen. Diese Optionen lassen sich grob in einzelne Datenbankdienste und Managed-Services-Angebote unterteilen, die jeweils unterschiedliche Merkmale und Vorteile aufweisen. Das Verständnis dieser Optionen ist für Unternehmen von entscheidender Bedeutung, die Vektorsuchfunktionen effektiv implementieren und gleichzeitig optimale Leistung, Skalierbarkeit und Kosteneffizienz gewährleisten möchten.

Weitere Informationen zu Vektordatenbanklösungen finden Sie in den folgenden Abschnitten:

- Individuelle Vektordatenbankoptionen
- Option für verwalteten Dienst
- Auswahl der richtigen Vektordatenbank

Individuelle Vektordatenbankoptionen

Zu den einzelnen Vektordatenbankoptionen AWS gehören Amazon Kendra, Amazon OpenSearch Service, AmazonRDS for PostgreSQL mit pgvector, Amazon MemoryDB, Amazon DocumentDB, AmazonNeptune Analytics und Amazon S3 Vector. (Als Open-Source-Erweiterung bietet pgvector die Möglichkeit, ML-generierte Vektoreinbettungen zu speichern und zu durchsuchen.) Diese Lösungen bieten unterschiedliche Ansätze für die Vektorsuche, sodass Unternehmen auf der Grundlage ihrer vorhandenen Infrastruktur, ihrer technischen Anforderungen und ihrer spezifischen Anwendungsfälle eine Auswahl treffen können.

Amazon Kendra

Amazon Kendra ist ein intelligenter Suchdienst für Unternehmen, der natürliche Sprachverarbeitung und fortschrittliche Algorithmen für maschinelles Lernen verwendet, um spezifische Antworten auf Suchfragen aus Ihren Daten zurückzugeben. Amazon Kendra vereinfacht die Implementierung von Suchfunktionen und ist damit eine effektive Backend-Lösung für generative KI-Anwendungen.

Zu den weiteren wichtigen Funktionen von Amazon Kendra gehören:

- Native Verbindungen zu über 40 Datenquellen
- Integrierte Funktionen zur Datenaufbereitung

- Schnelle Einrichtung, für die kein tiefes technisches Fachwissen erforderlich ist

Zu den Vorteilen von Amazon Kendra gehören:

- Automatisierte Datenverarbeitung (Chunking, Ingestion, Abruf)
- Leistungsstarke Anpassungsoptionen:
  - Facettensuche
  - Analytik durchsuchen
  - Optimierung der Suchrelevanz
- Einfacher programmatischer Zugriff über AWS SDK for Python (Boto3)

Weitere Informationen finden Sie unter Vorteile von Amazon Kendra in der Amazon Kendra Kendra-Dokumentation.

OpenSearch Amazon-Dienst

Amazon OpenSearch Service ist ein verwalteter Service, der Sie bei der Bereitstellung, dem Betrieb und der Skalierung von OpenSearch Service-Clustern in der unterstützt AWS Cloud.

Zu den Kernfunktionen von OpenSearch Service gehören die folgenden:

- Open-Source-Such- und Analyse-Engine
- Verteilte Architektur
- Datenverarbeitung in Echtzeit

Zu den Vorteilen der Nutzung des OpenSearch Dienstes gehören die folgenden:

- Horizontale Skalierbarkeit
- RESTful API-Unterstützung
- Verarbeitet strukturierte und unstrukturierte Daten
- Datenanalyse in Echtzeit
- Geeignet für verschiedene Einsatzgrößen

Weitere Informationen finden Sie unter Funktionen von Amazon OpenSearch Service in der OpenSearch Servicedokumentation.

Amazon RDS for PostgreSQL mit pgvector

Amazon RDS for PostgreSQL mit pgvector kombiniert den AWS verwalteten relationalen Datenbankservice mit der Vektorverarbeitungserweiterung von PostgreSQL. Diese Kombination ermöglicht es Unternehmen, hochdimensionale Vektoren zu speichern und abzufragen und gleichzeitig Amazon RDS beizubehalten. Die Lösung eignet sich besonders für generative KI-Anwendungen, die Vektoroperationen in Echtzeit erfordern, ohne den Aufwand für die Verwaltung der Datenbankinfrastruktur.

Zu den wichtigsten Vorteilen von Amazon RDS for PostgreSQL mit pgvector gehören:

- Hohe Verfügbarkeit
- Automatisches Failover
- Kostengünstig () pay-per-use
- Integrierte Überwachung
- Integration von Vektordaten in Echtzeit

Weitere Informationen finden Sie unter Vorteile von Amazon RDS in der Amazon RDS-Dokumentation.

Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist eine Dokumentendatenbank, die native Vektorsuchfunktionen in Version 5.0 und höher bietet. Sie kombiniert die Flexibilität der JSON-basierten Dokumentenablage mit der Vektorsuche und unterstützt sowohl hierarchische Navigable Small World (HNSW) als auch Inverted File Flat () Indexierungsmethoden. IVFFlat

Zu den Kernfunktionen von Amazon DocumentDB gehören:

- Speichern und indizieren Sie Vektoren mit bis zu 2.000 Dimensionen (bis zu 16.000 Dimensionen ohne Indizierung)
- Antwortzeiten in Millisekunden für Vektorähnlichkeitssuchen
- Support für euklidische, Kosinus- und Punktabstandsmetriken
- Nahtlose Integration mit bestehenden MongoDB-kompatiblen Anwendungen

Verwenden Sie Amazon DocumentDB in den folgenden Situationen:

- Für Anwendungen, die MongoDB bereits verwenden APIs und Vektorsuchfunktionen benötigen
- Für Anwendungsfälle, die flexible Dokumentendatenstrukturen in Kombination mit semantischer Suche erfordern
- Für Szenarien, die sowohl herkömmliche Dokumentenabfragen als auch Vektorähnlichkeitssuchen erfordern
- Für Anwendungen, die Produktempfehlungen, Personalisierung, Chat-Assistenten und Betrugserkennung bieten

Weitere Informationen finden Sie unter Vektorsuche für Amazon DocumentDB in der Amazon DocumentDB DocumentDB-Dokumentation.

Amazon MemoryDB

Amazon MemoryDB ist eine Redis-kompatible In-Memory-Datenbank, die unter den gängigen Vektordatenbanken die schnellste Vektor-Suchleistung bietet. AWS Sie bietet Abfragelatenzen von unter einer Millisekunde mit Beständigkeit in mehreren Availability Zones.

Zu den Kernfunktionen von MemoryDB gehören:

- Speichern Sie Anwendungsdaten und Millionen von Vektoren in einer einzigen Datenbank
- Antwortzeiten für Abfragen und Updates im einstelligen Millisekundenbereich
- Höchste Rückrufraten bei schnellster Leistung bei AWS
- Support für bis zu 32.768 Dimensionen pro Vektor
- Semantische Such- und Caching-Funktionen in Echtzeit

Verwenden Sie MemoryDB in den folgenden Situationen:

- Für Echtzeitanwendungen, die eine extrem niedrige Latenz (unter 10 ms) erfordern
- Für Workloads mit hohem Durchsatz und Millionen von Anfragen pro Tag
- Für Anwendungsfälle wie Empfehlungs-Engines in Echtzeit, semantisches Caching und Anomalieerkennung
- Für Anwendungen, die sowohl speicherinterne Datenspeicher- als auch Vektorsuchfunktionen benötigen

Weitere Informationen finden Sie unter Vektorsuche in der MemoryDB-Dokumentation.

Amazon Neptune Analytics

Amazon Neptune Analytics ist eine Graphanalyse-Engine, die native Vektorsuchfunktionen bietet und sich somit ideal für Anwendungsfälle der Graph Retrieval Augmented Generation (GraphRag) eignet. Sie kombiniert die Suche nach Vektorähnlichkeit mit Graphendurchläufen und Algorithmen.

Zu den Kernfunktionen von Neptune Analytics gehören:

- Analysieren Sie Dutzende von Milliarden von Beziehungen innerhalb von Sekunden
- Kombinieren Sie die Vektorsuche mit Graphalgorithmen (Pfadfindung, Erkennung von Gemeinschaften, Zentralität)
- Support für GraphRag-Anwendungen mit topologischem Wissen
- Bis zu 80-mal schneller als bestehende Lösungen für die Graphanalyse
- Integration mit Amazon Bedrock für vollständig verwaltetes GraphRag

Verwenden Sie Neptune Analytics in den folgenden Situationen:

- Für GraphRag-Anwendungen, die Wissensgraphen mit Vektoreinbettungen benötigen
- Für Anwendungsfälle, bei denen neben der Vektorähnlichkeit auch komplexe Beziehungen überwunden werden müssen
- Für Anwendungen, die erklärbare KI-Antworten mit Beziehungskontext erfordern
- Für Szenarien wie 360-Grad-Kundenansichten, Netzwerke zur Betrugserkennung und Wissensentdeckung

Weitere Informationen finden Sie in der Dokumentation zu Amazon Neptune Analytics.

Amazon S3 Vectors

Amazon S3 Vectors ist der erste Cloud-Objektspeicher AWS mit nativen Vektorspeicher- und Abfragefunktionen. Er bietet speziell entwickelten, kostenoptimierten Vektorspeicher für KI-Anwendungen, die eine enorme Skalierung erfordern.

Zu den Kernfunktionen von Amazon S3 Vectors gehören:

- Speicher für bis zu 2 Milliarden Vektoren pro Index mit Unterstützung für bis zu 10.000 Indizes pro Vektor-Bucket
- Abfragelatenz unter 100 ms, die für Langzeitspeicherung und seltene Zugriffsmuster optimiert ist

- Bis zu 90% geringere Kosten für Vektoroperationen im Vergleich zu speziellen Vektordatenbanken
- Serverlose Architektur mit automatischer Skalierung und einer Lebensdauer von 99,999999999% (11 9s)

Verwenden Sie Amazon S3 Vectors in den folgenden Situationen:

- Für Anwendungen, die Milliarden von Vektoren zu minimalen Kosten speichern müssen
- Für Workloads, die eine Abfragelatenz von weniger als einer Sekunde (100 ms oder mehr) statt weniger als 10 ms tolerieren
- Für Anwendungsfälle zur langfristigen Aufbewahrung und Archivierung von Vektoren
- Für RAG-Anwendungen mit seltenen Abrufmustern
- Für Unternehmen, die der Wirtschaftlichkeit des Speichers Vorrang vor extrem niedriger Latenz einräumen

Amazon S3 Vectors lässt sich nativ in Amazon Bedrock Knowledge Bases integrieren und funktioniert gut in mehrstufigen Architekturen mit Amazon Service. OpenSearch Sie können Amazon S3 Vectors für Cold Storage und OpenSearch Service für Hot Queries verwenden.

Weitere Informationen finden Sie unter Arbeiten mit S3-Vektoren und Vektor-Buckets in der Amazon S3 S3-Dokumentation.

Option „Verwalteter Service“

Amazon Bedrock Knowledge Bases steht für den AWS vollständig verwalteten Ansatz zur Implementierung von Vektordatenbanken. Die Flexibilität der Speicheroptionen des Service in Kombination mit seinen automatisierten Verwaltungsfunktionen macht ihn besonders für Unternehmen interessant, die RAG implementieren möchten, ohne eine komplexe Infrastruktur verwalten zu müssen.

Mit Amazon Bedrock Knowledge Bases können Sie Wissensdatenbanken erstellen, verwalten und abfragen, die Ihre Basismodelle mithilfe von RAG verbessern. Dieser Service vereinfacht den komplexen Prozess der Implementierung von RAG, indem er die gesamte Pipeline für Datenaufnahme, Vektorisierung und Datenabruf verwaltet.

Zu den wichtigsten Vorteilen von Amazon Bedrock Knowledge Bases gehören:

- Vereinfachte Datenverarbeitung

    - Automatische Datenaufnahme und -aufteilung
    - Integrierte Textextraktion aus mehreren Dateiformaten
    - Generierung verwalteter Vektor-Einbettungen
    - Automatische Extraktion und Indexierung von Metadaten
- Optimierte RAG-Implementierung
    - Vorkonfigurierte Abrufstrategien
    - Automatische Optimierung des Kontextfensters
    - Integrierte Relevanzoptimierung
    - Semantische Suchfunktionen, sofort einsatzbereit
- Sicherheit und Governance
    - Integrierte AWS Identity and Access Management (IAM) Steuerungen
    - Datenverschlüsselung im Ruhezustand und bei der Übertragung
    - VPC-Unterstützung
    - Audit-Protokollierung mit AWS CloudTrail

Amazon Bedrock Knowledge Bases unterstützt mehrere Vector Store-Optionen, darunter:

- Amazon Aurora PostgreSQL mit pgvector
- Amazon Neptune Analytics
- Amazon EMR Serverless
- Amazon S3 Vectors
- Tannenzapfen
- Redis Enterprise Cloud

Dieser verwaltete Service kümmert sich um die automatische Erfassung, Vektorisierung und den Abruf von Daten. Dies vereinfacht RAG-Implementierungen.

Detaillierte Informationen zu den einzelnen unterstützten Vector Stores finden Sie in der Amazon Bedrock Knowledge Bases-Dokumentation.

Auswahl der richtigen Vektordatenbank

Wählen Sie Ihre Vektordatenbank auf der Grundlage dieser wichtigen Entscheidungsfaktoren aus:

- Wenn Sie eine MongoDB-kompatible Dokumentendatenbank mit Vektorsuche benötigen, wählen Sie Amazon DocumentDB. Dies ist ideal, wenn Ihre Anwendung MongoDB verwendet APIs und Sie semantische Suchfunktionen hinzufügen möchten, ohne eine separate Vektorinfrastruktur verwalten zu müssen.
- Wenn Sie eine extrem niedrige Latenz für Echtzeitanwendungen benötigen, wählen Sie Amazon MemoryDB. Dies bietet die schnellste Vektor-Suchleistung AWS mit Reaktionszeiten von unter einer Millisekunde. Es ist ideal für Empfehlungsmaschinen in Echtzeit und Anwendungen mit hohem Durchsatz.
- Wenn Sie graphenbasierte Wissensdarstellungen mit Vektorsuche benötigen, entscheiden Sie sich für Amazon Neptune Analytics. Dies eignet sich am besten für GraphRag-Anwendungen, die komplexe Zusammenhänge durchqueren und neben Vektorsuchen auch grafenbasierte Abfragen durchführen müssen, um erklärbare KI-Antworten zu liefern.
- Wenn Sie relationale Abfragen mit Vektorsuche kombinieren müssen, wählen Sie Amazon Aurora PostgreSQL with pgvector. Diese Option ist ideal, wenn Ihre Anwendung sowohl traditionelle SQL-Operationen als auch Vektorähnlichkeitssuchen innerhalb derselben Datenbank erfordert.
- Wenn Sie Abfragen mit hohem Durchsatz und einer Latenz von weniger als 10 ms benötigen, wählen Sie Amazon OpenSearch Service. Es zeichnet sich durch die Verarbeitung von hochfrequenten Abfragen und Echtzeitanwendungen aus und umfasst aktuelle Verbesserungen der GPU-Beschleunigung.
- Wenn Sie Milliarden von Vektoren kostengünstig speichern müssen, entscheiden Sie sich für Amazon S3 Vectors. Diese Option bietet Kosteneinsparungen von bis zu 90% und ist ideal für Anwendungen mit seltenen Abrufmustern (Minuten bis Stunden zwischen Abfragen), die eine Latenz von weniger als 100 ms tolerieren können.
- Wenn Sie neben der Vektorsuche auch eine Volltextsuche benötigen, wählen Sie Amazon OpenSearch Service. Diese Option kombiniert leistungsstarke Funktionen für die Volltextsuche mit der Vektorsuche auf einer einzigen Plattform.

Vergleich von Vektordatenbanken

AWS bietet mehrere Ansätze zur Implementierung von Vektorsuchfunktionen, die von einzelnen Vektordatenbanken bis hin zu Amazon Bedrock Knowledge Bases, einem vollständig verwalteten Service, reichen. Bei der Bewertung dieser Optionen müssen Unternehmen verschiedene Aspekte berücksichtigen, darunter Architektur, Skalierbarkeit, Integrationsmöglichkeiten, Leistungsmerkmale und Sicherheitsmerkmale.

Individuelle Vektordatenbanken

Die folgende Tabelle bietet einen Überblick über die wichtigsten Funktionen verschiedener AWS individueller Vektordatenbanklösungen, wobei der Schwerpunkt auf deren Architekturen, Skalierungsmöglichkeiten, Datenquellenintegrationen und Leistungsmerkmalen liegt.

Merkmal	Amazon Kendra	OpenSearch Amazon-Dienst	Amazon RDS für Postgre-Pvvector SQLwith	Amazon DocumentDB	Amazon MemoryDB	Amazon Neptune Analytics	Amazon S3 Vectors
Primärer Anwendungsfall	Unternehmenssuche und RAG	Verteilte Suche und Analytik	Relationale Datenbank mit Vektorunterstützung	Dokumenten-Datenbank mit Vektorsuche	Vektorsuche im Speicher in Echtzeit	Graphanalyse mit Vektorsuche	Kostenoptimierter Vektorspeicher
Architektur	Vollständig verwaltet	Verteilter Cluster	Relationale Datenbank	Dokumentenorientiert	In-Memory-Datenbanken	Graph-Analyse-Engine	Serverloser Objektspeicher
Datenmodell	Dokumentenbasiert	JSON-Dokumente	Relationale Tabellen	JSON-Dokumente	Schlüsselwert mit JSON	Eigenschaftsdiagramm	Objektspeicher

Abmessungen von Vektoren	Automatisch verwaltet	Bis zu 16.000	Konfigurierbar	Bis zu 2.000 (indexiert); 16.000 (nicht indexiert)	Bis zu 32.768	Konfigurierbar	Bis zu 4.096
Methoden zur Indizierung	Automatisch	HNSW, IVF	HNSW, IVFFlat	HNSW, IVFFlat	HNSW	Nativer Graph und Vektor	Automatisch
Entfernungsmetriken	Automatisch	Kosinus, Euklidisch, Punktprodukt	Kosinus, Euklidisch, inneres Produkt	Kosinus, Euklidisch, Punktprodukt	Kosinus, Euklidisch, inneres Produkt	Kosinus, Euklidisch	Kosinus, Euklidisch
Latenz bei Abfragen	In weniger als einer Sekunde	Unter 10 ms (GPU-beschleunigt)	10-100 ms	Millisekunde	Submillisekunde	In weniger als einer Sekunde	Unter 100 ms
Skalierungsmodell	Automatisch	Horizontal (Knoten hinzufügen)	Vertikale und lesbare Repliken	Horizontal (Instanzen hinzufügen)	Vertikal und Replikate	Automatisch	Automatisch (serverlos)

Maximale Anzahl von Vektoren	Verwaltet	Milliarden (clusterabhängig)	Millionen (instanzabhängig)	Millionen pro Sammlung	Millionen pro Datenbank	Milliarden	2 Milliarden pro Index; 10.000 Indizes pro Bereich
Durchsatz	Hoch	Sehr hoch (Tausende von QPS)	Medium	Hoch	Sehr hoch (Millionen von Anfragen pro Tag)	Hoch	Mittel (optimiert für seltene Abfragen)
Lebensdauer von Daten	99,999999999 % (11x9)	Konfigurierbar mit Replikaten	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99 %	99,999999999 % (11x9)
Konsistenzmodell	Eventuell	Eventual (konfigurierbar)	Stark (ACID)	Irgendwann	Stark	Stark	Stark
Zusätzliche Funktionen	40 oder mehr Datenanschlüsse, NLP	Volltextsuche, Analytik, Dashboards	SQL-Abfragen, ACID-Transaktionen	MongoDB-API-Kompatibilität	Redis-API-Kompatibilität, Caching	Graphalgorithmen, Traversalen	Amazon S3 S3-Integration, Lebenszyklusrichtlinien

Preismodell	Zahlen Sie pro Anfrage und Speicherplatz	Stunden und Speicherplatz der Instanz	Stunden und Speicherplatz der Instanz	Stunden und Speicherplatz der Instanz	Stunden und Speicherplatz der Instanz	Kapazitätseinheiten und Speicherplatz	Speicherung, Abfragen und Datenübertragung
Kostenoptimierung	Nutzungsbasiert	Reservierte Instances, auto-scaling	Reservierte Instanzen, Aurora Serverless	Reserved Instances	Reserved Instances	Auto Scaling	Einsparungen von bis zu 90% im Vergleich zu Spezialprodukten DBs
Am besten geeignet für	Unternehmenssuche mit minimalem Einrichtungsaufwand	Abfragen mit hohem Durchsatz und niedriger Latenz	Hybride SQL- und Vektor-Workloads	MongoDB-kompatible Apps, die Vektoren benötigen	Apps in Echtzeit mit extrem niedriger Latenz	GraphRag und Wissensgraphen	Langfristige, kostengünstige Lagerung
Ideales Abfragemuster	Häufige Suchanfragen in Unternehmen	Hochfrequente Abfragen in Echtzeit	Gemischte SQL- und Vektorabfragen	Dokumentenabfragen mit semantischer Suche	Millionen von Anfragen pro Tag	Traversierungen mit Vektorsuche grafisch darstellen	Seltene Anfragen (Minuten bis Stunden)
Komplexität der Einrichtung	Niedrig (vollständig verwaltet)	Mittel (Clusterkonfiguration)	Mittel (Erweiterungs-Setup)	Mittel (Clusterkonfiguration)	Mittel (Clusterkonfiguration)	Niedrig (vollständig verwaltet)	Niedrig (serverlos)

Fachwissen des Teams erforderlich	Minimal	OpenSearch oder Elasticsearch	PostgreSQL, SQL	MongoDB	Redis	Graph-Datenbanken	Amazon S3, grundlegende Vektorkonzepte

Verwalteter Service — Amazon Bedrock Knowledge Bases

Amazon Bedrock Knowledge Bases bietet eine vollständig verwaltete Lösung mit mehreren Vektorspeicheroptionen. In der folgenden Tabelle werden diese Speicheroptionen verglichen.

Merkmal	Aurora Postgre-Pvektor SQLwith	Neptun-Analytik	OpenSearch Serverloser Service	Amazon S3 S3-Vektoren	Tannenzapfen	RedisEnterprise Cloud
Primärer Anwendungsfall	Relationale Datenbank mit Vektor-RAG	Graphbasierte Vektorsuche für GraphRag	Wissensmanagement RAG	Kostenoptimierter Vektor-RAG	Hochleistungsfähige Vektorsuche	Vektorsuche im Speicher
Architektur	Vollständig verwaltetes relationales	Vollständig verwaltete Graphenanalysen	Vollständig verwaltetes, serverloses System	Serverloser Objektspeicher	Vollständig verwaltete Hybrid-Cloud	Vollständig verwalteter In-Memory-Speicher
Datenmodell	Relationale Tabellen	Eigenschaftsdiagramm	JSON-Dokumente	Objektspeicher	Speziell entwickelte Vektoren	Schlüssel wert mit Vektoren
Vektor-Speicher	Durch die Erweiteru	Native Graphvektoren	Durch den OpenSearch Motor	Nativer Amazon S3	Native Vektordatenbank	Vektoren im Speicher

	ng pgvector			S3-Vektor speicher		
Integration mit Amazon Bedrock	Nativ	Nativ	Nativ	Nativ	Nativ	Nativ
Automatische Einnahme	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)
Automatische Vektorisierung	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)	Ja (über Amazon Bedrock)
Skalierung	Automatische Skalierung (Aurora Serverless)	Automatische Skalierung von Diagrammen	Automatisch serverlos	Automatisch (Milliarden von Vektoren)	Automatische Skalierung von Pods	Automatische Skalierung von Clustern
Abfrageleistung	Hoch für relational oder vektoriell	Hoch für Graphvektoren	Hoch	Mittel (Latenz von 100 ms oder mehr)	Very high (Sehr hoch)	Very high (Sehr hoch)
Maximale Vektoren	Millionen (instanzabhängig)	Milliarden	Milliarden	2 Milliarden pro Index	Milliarden	Millionen (speicherabhängig)
Zusätzliche Funktionen	SQL-Abfragen, ACID-Transaktionen	Graphalgorithmen, Traversalen	Volltextsuche, Analytik	Amazon S3 S3-Lebenszyklus, Tiering	Filterung von Metadaten, Namespaces	Redis-Datenstrukturen, Caching

Kostenoptimierung	Moderat (Aurora Serverless)	Moderat (Kapazitätseinheiten)	Hoch (serverlos, pay-per-use)	Sehr hoch (Einsparungen von bis zu 90%)	Moderat (pod-basierte Preisgestaltung)	Niedrig (In-Memory-Premium)
Am besten geeignet für	Hybride Workloads SQL/vector	Verbundene Wissensgraphen	Volltext mit Vektorsuche	Langfristige Vektoren mit seltenem Zugriff	Vektorsuche in Echtzeit in großem Maßstab	Anforderungen an extrem niedrige Latenz
Ideales Abfragemuster	Gemischte SQL- und Vektorabfragen	Traversierungen mit Vektoren grafisch darstellen	Häufige Suchanfragen mit Analysen	Seltener Abruf (Minuten bis Stunden)	Hochfrequente Abfragen in Echtzeit	Millionen von Anfragen pro Sekunde
Einrichtung mit Amazon Bedrock	Einfach (verwaltet von Amazon Bedrock)	Einfach (verwaltet von Amazon Bedrock)	Einfach (verwaltet von Amazon Bedrock)	Einfach (verwaltet von Amazon Bedrock)	Einfach (verwaltet von Amazon Bedrock)	Einfach (verwaltet von Amazon Bedrock)
Datenresidenz	AWS-Regionen	AWS-Regionen	AWS-Regionen	AWS-Regionen	Multi-Cloud (AWS und andere)	Multi-Cloud (AWS und andere)
Preismodell	Stunden und Speicherplatz der Instanz	Kapazitätseinheiten und Speicherplatz	Rechenleistung und Speicher (serverlos)	Speicherung, Abfragen und Übertragung	Öffnungszeiten und Speicherplatz	Öffnungszeiten und Speicherplatz der Knoten

Wählen Sie zwischen individuellen und verwalteten Optionen

Überlegung	Wählen Sie eine individuelle Vektor-DB	Wählen Sie Amazon Bedrock Knowledge Bases (verwaltet)
RAG-Implementierung	Sie möchten die volle Kontrolle über die RAG-Pipeline	Sie möchten ein vollständig verwaltetes RAG mit minimalem Einrichtungsaufwand
Anpassung	Sie benötigen eine benutzerdefinierte Abruflogik und Vorverarbeitung	Standard-RAG-Muster erfüllen Ihre Anforderungen
Bestehende Infrastruktur	Sie haben die Datenbank bereits bereitgestellt	Sie fangen neu an oder wünschen sich eine vereinfachte Verwaltung
Fachwissen im Team	Ihr Team verfügt über Fachkenntnisse in der Datenbankadministration	Sie konzentrieren sich lieber auf die Anwendungslogik als auf die Infrastruktur
Komplexität der Integration	Sie benötigen eine tiefe Integration mit bestehenden Systemen	Sie möchten eine schnelle Integration mit Amazon Bedrock-Modellen
Operativer Overhead	Sie können Datenbankoperationen verwalten	Sie AWS möchten Operationen abwickeln
Kostenstruktur	Sie bevorzugen direkte Datenbankpreise	Sie bevorzugen eine einheitliche Amazon Bedrock-Preisgestaltung
Zeit bis zur Markteinführung	Sie haben Zeit für eine kundenspezifische Implementierung	Sie benötigen eine schnelle Bereitstellung

Kostenvergleiche und Überlegungen

Um fundierte Implementierungsentscheidungen treffen zu können, ist es wichtig, die Kostenstruktur verschiedener Vektordatenbankoptionen zu verstehen. In der folgenden Tabelle sind einige wichtige Kostenaspekte für verschiedene Vektordatenbanklösungen aufgeführt, die sowohl einzelne Datenbanken als auch verwaltete Dienste umfassen. Jede Option hat unterschiedliche Preisfaktoren, die sich auf die Gesamtbetriebskosten (TCO) auswirken können, angefangen bei pay-as-you-go Modellen bis hin zu Infrastruktur- und Betriebskosten.

Vektor-Datenbank	Kostenmodell	Überlegungen zu den Kosten
Amazon Kendra	Pay-as-you-go, basierend auf Anfragen	Die Kosten können je nach Anzahl der Abfragen und Menge der indexierten Daten variieren. Für Datenspeicherung und Datenübertragung können zusätzliche Gebühren anfallen. Weitere Informationen finden Sie unter Amazon Kendra — Preise.
OpenSearch Amazon-Dienst	Pay-as-you-go, basierend auf Instanzstunden und Speicherplatz	Zu den Kosten gehören Instance-Stunden, Speicher (Amazon EBS-Volumes), Datenübertragung und optionaler UltraWarm Speicher. Reserved Instances können Einsparungen von bis zu 30% ermöglichen. GPU-beschleunigte Instances bieten ein besseres Preis-Leistungs-Verhältnis für Vektor-Workloads. Weitere Informationen finden Sie unter Amazon OpenSearch Service-Preise.

Open-Source-Software OpenSearch	Open Source, keine direkten Kosten (Sie müssen nicht bezahlen, um die Software herunterzuladen oder zu verwenden, und es fallen keine Lizenzkosten an)	Die Kosten umfassen Infrastruktur (wie Server und Speicher) und Betriebskosten (wie Wartung und Überwachung). Organizations müssen Personal für die Verwaltung und Wartung der Infrastruktur budgetieren.
Amazon RDS for PostgreSQL mit pgvector	Zahlen Sie nach Bedarf, basierend auf der Nutzung	Zu den Kosten gehören Datenbankinstanztypen, Speicher, Datenübertragung und Backups. Zusätzliche Gebühren können für Datentransfer, Instance-Typen und Speicher anfallen, die über das AWS kostenlose Kontingent hinausgehen. Weitere Informationen finden Sie unter Amazon RDS – Preise.
Amazon DocumentDB	Pay-as-you-go, basierend auf den Instanzstunden und dem Speicherplatz	Zu den Kosten gehören Instanzstunden, Speicher (GB-Monat), I/O Anfragen, Backup-Speicher und Datenübertragung. Elastische Cluster ermöglichen dynamische Skalierung. Reserved Instances sind zur Kostenoptimierung verfügbar. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Amazon MemoryDB	Pay-as-you-go, basierend auf Knotenstunden und Datenspeicher	Zu den Kosten gehören Knotenstunden (pro Knotentyp), Datenspeicher (GB-Stunde), Snapshot-Speicher und Datenübertragung. Reservierte Knoten können Einsparungen von bis zu 55% ermöglichen. Optimiert für Workloads mit hohem Durchsatz und niedriger Latenz. Weitere Informationen finden Sie unter Amazon MemoryDB-Preise.
Amazon Neptune Analytics	Pay-as-you-go, basierend auf Kapazitätseinheiten	Die Kosten beinhalten Neptun-Kapazitätseinheiten (NCUs), Speicher (GB-Monat) und Datenübertragung. Automatische Skalierung auf der Grundlage der Arbeitslast ohne Vorabverpflichtungen. Mindestens 128 NCUs erforderlich. Weitere Informationen finden Sie unter Amazon Neptune — Preise.

Amazon S3 Vectors	Pay-as-you-go, basierend auf Speicherplatz und Anfragen	Zu den Kosten gehören Speicherplatz (GB-Monat), PUT- und GET-Anfragen, Vektorindexverwaltung und Datenübertragung. Bietet Kosteneinsparungen von bis zu 90% im Vergleich zu spezialisierten Vektordat enbanken. Amazon S3 Intelligent-Tiering- und Lebenszyklusrichtlinien sind für zusätzliche Optimierungen verfügbar. Weitere Informati onen finden Sie unter Amazon S3 – Preise.
Amazon Bedrock Wissensda tenbanken	Zahlen Sie nutzungsabhängig	Die Kosten können je nach Nutzung der Wissensda tenbank und zusätzlic her Dienste wie Amazon OpenSearch Serverles s variieren. Zusätzliche Gebühren können für Datenspeicherung, Datenüber tragung und zusätzliche Funktionen anfallen. Weitere Informationen zu den Preisen finden Sie unter Amazon OpenSearch Service-Preise.

Anwendungsfälle für Vektordatenbanken

Die folgenden Beispiele zeigen, wie verschiedene Vektordatenbankoptionen effektiv genutzt werden können, um das Wissensmanagement zu verbessern, die betriebliche Effizienz zu verbessern und bessere Geschäftsergebnisse zu erzielen. Diese Anwendungsfälle veranschaulichen die praktischen Anwendungen der weiter oben in diesem Leitfaden erörterten Vektordatenbanklösungen und bieten Einblicke in deren Leistung und Vorteile in der Praxis.

Wissensmanagement mit Amazon Kendra

Kundenproblem — Einer der größten Generalunternehmer in Japan sah sich mit einem Rückgang an erfahrenem Personal konfrontiert. Das Unternehmen benötigte eine Möglichkeit, das Wissen und die Fähigkeiten der erfahrenen Mitarbeiter effizient an die jüngere Generation weiterzugeben. Sie benötigten eine Lösung zur Erfassung und Verbreitung komplexer bautechnischer Kenntnisse und Erfahrungen aus der Vergangenheit.

AWS Lösung — Um dieses Problem zu lösen, wandte sich der Kunde an Amazon Kendra, eine KI-Lösung, die seine interne Wissensdatenbank schnell und präzise verwalten und Abfragen in natürlicher Sprache ermöglichen konnte. Mit Amazon Kendra können Mitarbeiter die benötigten Informationen jetzt viel schneller finden, was die Produktivität verbessert und den Wissenstransfer von erfahrenen Mitarbeitern zu jüngeren Mitarbeitern erleichtert.

Wirkung — Durch die Implementierung eines generativen KI-Chatbots auf Basis von Amazon Kendra schuf das Unternehmen eine einheitliche Wissensplattform. Der Chatbot ermöglicht es Mitarbeitern, schnell auf technisches Wissen und frühere Erfahrungen im Bauwesen zuzugreifen. Diese Lösung hat die Effizienz des Wissenstransfers und der Entscheidungsprozesse innerhalb des Unternehmens erheblich verbessert und dazu beigetragen, dass wertvolles Fachwissen erhalten bleibt und für alle Mitarbeiter leicht zugänglich ist. Die Kosten dieser Lösung können je nach Nutzung und Konfiguration variieren. Einen detaillierten Kostenvoranschlag finden Sie unter AWS -Preisrechner. Informationen zur Kostenschätzung für Vektordatenbanken finden Sie im Abschnitt Kostenvergleich und Überlegungen in diesem Handbuch oder unter Amazon Kendra — Preise.

Informationen zu anderen Anwendungsfällen von Kunden finden Sie unter Amazon Kendra Kendra-Kunden.

Echtzeitanalysen mit Serverless OpenSearch

Kundenproblem — Ein führender Finanzdienstleister stand vor der Herausforderung, ein riesiges Datenökosystem zu verwalten. Das Unternehmen verarbeitete jährlich 300 Millionen Autorisierungen und 90 Milliarden Transaktionen, was einer Datenmenge von etwa 1,1 Petabyte (PB) entspricht. Das bestehende System für 300.000 Benutzer, die Zugriff auf über 6.000 Berichte benötigten, musste modernisiert werden, um globale Konsistenz zu gewährleisten und Entscheidungen in Echtzeit zu ermöglichen.

AWS Lösung — Die Lösungsarchitektur verwendete Basismodelle, die über Amazon Bedrock verfügbar sind (einschließlich Anthropic, Sonnet 3, Sonnet 3.5 und Haiku) für die Verarbeitung natürlicher Sprache. Der Kunde entschied sich aufgrund der überragenden Skalierbarkeit und der Fähigkeit, das riesige Datenvolumen effizient zu handhaben, für OpenSearch Serverless als Vektordatenbank. Diese Architektur ermöglichte die nahtlose Verarbeitung komplexer Abfragen und die dynamische Generierung von Berichten.

Wirkung — Durch die Implementierung konnte die Produktivität um 50 Prozent gesteigert werden, da die manuelle Generierung von über 100 Business Intelligence-Dashboards überflüssig wurde. Benutzer können jetzt Berichte mithilfe von Abfragen in natürlicher Sprache mit Antwortzeiten zwischen 20 und 40 Sekunden erstellen. Die Kosten für diese Lösung können je nach Nutzung und Konfiguration variieren. Einen detaillierten Kostenvoranschlag finden Sie unter AWS -Preisrechner. Informationen zur Kostenschätzung für Vektordatenbanken finden Sie im Abschnitt Kostenvergleich und Überlegungen in diesem Handbuch oder unter Amazon OpenSearch Service-Preise. Informationen zu anderen Anwendungsfällen von Kunden finden Sie unter Amazon OpenSearch Serverless.

Nächste Schritte und Ressourcen

Nachdem Sie diesen Leitfaden gelesen haben, sollten Sie die folgenden Maßnahmen in Betracht ziehen, um vom Verständnis zur Implementierung überzugehen:

1. Beurteilen Sie Ihre aktuellen Bedürfnisse:
   - Beurteilen Sie Ihre bestehende Datenbankinfrastruktur und Ihr Fachwissen.
   - Dokumentieren Sie Ihre spezifischen Anforderungen an die Vektorsuche.
   - Definieren Sie Ihre Leistungs-, Skalierungs- und Kostenziele.
2. Wählen Sie eine der folgenden Optionen, um die Vektordatenbankoptionen zu testen:
   - Option 1: Richten Sie einen Machbarkeitsnachweis mit Ihrer bevorzugten Vektordatenbanklösung ein.
   - Option 2: Experimentieren Sie mit Beispieldatensätzen in den Amazon Bedrock Knowledge Bases. Probieren Sie die Schnellerstellung für eine Amazon Bedrock Knowledge Base aus. Ein Beispiel finden Sie unter Schnelles Erstellen einer Aurora PostgreSQL-Wissensdatenbank für Amazon Bedrock in der Aurora-Dokumentation.
3. Sehen Sie sich zusätzliche Ressourcen an.
4. Holen Sie sich Expertenhilfe:
   - Wenden Sie sich an Ihr AWS-Konto Team oder Ihre AWS Solutions Architects, um Unterstützung bei der Implementierung zu erhalten.
   - Arbeiten Sie mit AWS Partnern zusammen, die sich auf Vektordatenbanken spezialisiert haben.
5. Planen Sie Ihren Produktionseinsatz:
   - Erstellen Sie eine Migrationsstrategie, wenn Sie von bestehenden Datenbanken migrieren.
   - Entwickeln Sie einen Skalierungsplan für die von Ihnen gewählte Lösung.
   - Entwerfen Sie Ihre Überwachungs- und Wartungsverfahren.

Ressourcen

Die folgenden Ressourcen können Ihnen bei der Auswahl einer Vektordatenbank helfen.

AWS Blog-Beiträge

- Beschleunigen Sie Ihre generative KI-Anwendungsentwicklung mit Amazon Bedrock Knowledge Bases, Quick Create und Amazon Aurora Serverless
- Die Vektordatenbankfunktionen von Amazon OpenSearch Service erklärt
- Tauchen Sie mithilfe der Amazon Bedrock Knowledge Bases tief in Vektordatenspeicher ein
- Nutzen Sie pgvector und Amazon Aurora PostgreSQL für die Verarbeitung natürlicher Sprache, Chatbots und Stimmungsanalyse

AWS Servicedokumentation

- Auswahl eines AWS Datenbankdienstes
- So funktionieren die Amazon Bedrock-Wissensdatenbanken
- Neptune Analytics-Dokumentation
- Überblick über Amazon Web Services: Datenbanken
- Verwendung von Aurora PostgreSQL als Wissensdatenbank für Amazon Bedrock
- Arbeiten mit Amazon Aurora PostgreSQL
- Amazon DocumentDB
- Amazon MemoryDB
- Amazon S3 Vectors

Andere Ressourcen AWS

- Amazon Bedrock Wissensdatenbanken
- Vektor-Datenbanken und Einbettungen
- Vektor-Datenbanken für generative KI-Anwendungen
- Was sind Einbettungen beim Machine Learning?

Sonstige Ressourcen

- Über PostgreSQL
- pgvector-Dokumentation

- Pinecone als Wissensdatenbank für Amazon Bedrock
- Redis Enterprise Cloud auf AWS

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen an diesem Handbuch „Auswahl einer AWS Vektordatenbank für RAG-Anwendungsfälle“ beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Hinzugefügt AWS-Services	Wir haben Informationen zur Verwendung von Amazon DocumentDB, Amazon MemoryDB, Amazon S3 Vectors und Amazon Neptune Analytics hinzugefügt.	30. März 2026
Erste Veröffentlichung	—	6. März 2025

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.