Elegir una base de datos AWS vectorial para casos de uso de RAG

AWS Orientación prescriptiva

AWS Orientación prescriptiva: Elegir una base de datos AWS vectorial para casos de uso de RAG

Table of Contents

Elegir una base de datos AWS vectorial para casos de uso de RAG

Mayuri Shinde, Ivan Cui y Anand Bukkapatnam Tirumala, de Amazon Web Services

Marzo de 2026 (historial del documento)

Las bases de datos vectoriales son cada vez más importantes para las organizaciones que implementan aplicaciones de IA generativa. Estas bases de datos almacenan y administran vectores, que son representaciones numéricas de datos que permiten procesar texto, imágenes y otros contenidos de manera que capten su significado y sus relaciones.

A medida que las organizaciones exploran las opciones de bases de datos vectoriales AWS, deben comprender las capacidades, las ventajas y las mejores prácticas de las diferentes soluciones. Esta guía le ayuda a comparar los almacenes de vectores más utilizados AWS y a tomar decisiones informadas sobre qué opciones se adaptan mejor a sus necesidades o casos de uso específicos. Ya sea que esté implementando la generación aumentada de recuperación (RAG), creando sistemas de recomendación o desarrollando otras aplicaciones de IA, esta guía proporciona un marco que le ayudará a evaluar y elegir una solución de base de datos vectorial.

Destinatarios previstos

Esta guía está destinada a personas que desempeñan las siguientes funciones:

- Científicos de datos e ingenieros de aprendizaje automático (ML) que utilizan bases de datos vectoriales para almacenar y recuperar datos de alta dimensión para modelos de ML.
- Ingenieros de datos que diseñan e implementan canalizaciones de datos que incluyen bases de datos vectoriales para almacenar y procesar datos de alta dimensión.
- MLOps ingenieros que utilizan bases de datos vectoriales como parte del proceso de aprendizaje automático para almacenar y entregar los resultados de los modelos o las representaciones intermedias.
- Ingenieros de software que integran bases de datos vectoriales en aplicaciones que requieren sistemas de búsqueda o recomendación de similitudes.
- DevOps ingenieros que se encargan de implementar y mantener las bases de datos vectoriales en los entornos de producción, garantizando la escalabilidad y la fiabilidad.

- Investigadores de IA que utilizan bases de datos vectoriales para almacenar y analizar grandes conjuntos de datos de incrustaciones o vectores de características.
- Directores de productos de IA que necesitan comprender las capacidades y limitaciones de las bases de datos vectoriales para tomar decisiones informadas sobre las características y la arquitectura de los productos.

Descripción general de los vectores

Los vectores son representaciones numéricas que ayudan a las máquinas a entender y procesar los datos. En la IA generativa, cumplen dos propósitos clave:

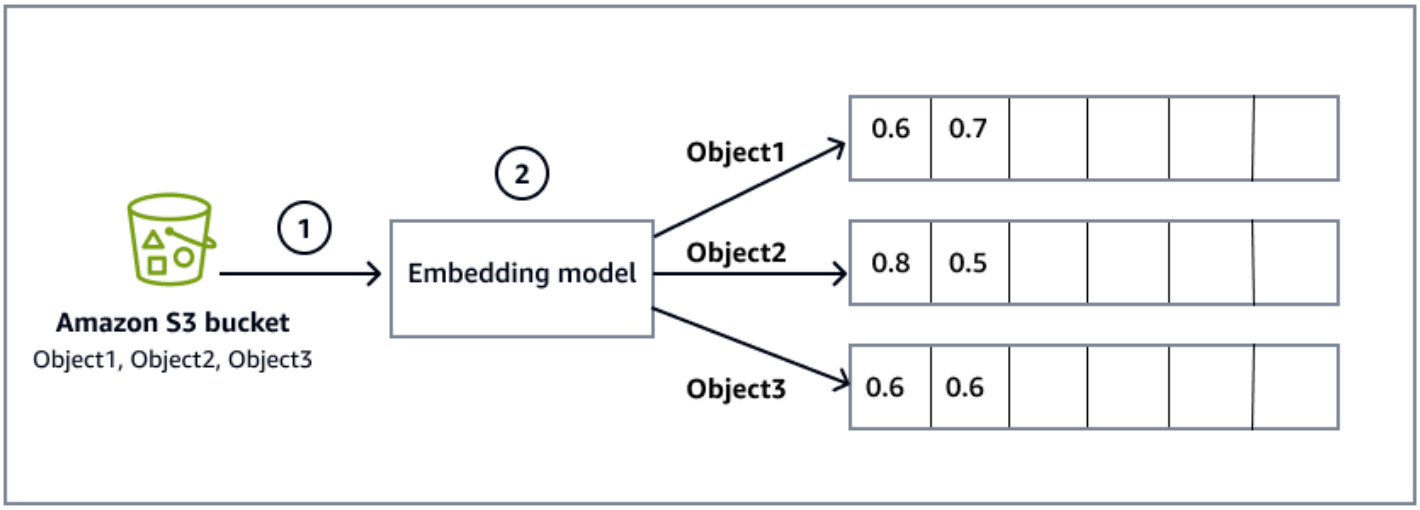
- Representar espacios latentes que capturan la estructura de datos en forma comprimida
- Crear incrustaciones de datos, como palabras, oraciones e imágenes

Los modelos de incrustación, como Word2Vec GloVey Amazon Titan Text Embeddings, convierten los datos en vectores mediante un proceso denominado incrustación. Estos modelos de incrustación pueden hacer lo siguiente:

- Aprenda del contexto a representar palabras como vectores
- Coloca palabras similares más cerca unas de otras en el espacio vectorial
- Permita que las máquinas procesen datos en un espacio continuo

El siguiente diagrama proporciona una visión general de alto nivel del proceso de incrustación:

1. Un bucket de Amazon Simple Storage Service (Amazon S3) contiene archivos que son las fuentes de datos desde las que el sistema leerá y procesará la información. El bucket de Amazon S3 se especifica durante la configuración de la base de conocimientos de Amazon Bedrock, que también incluye la sincronización de datos con la base de conocimientos.
2. El modelo de incrustación convierte los datos sin procesar de los archivos de objetos del bucket de Amazon S3 en incrustaciones vectoriales. Por ejemplo, `Object1` se convierte en un vector `[0.6, 0.7, ...]` que representa su contenido en un espacio multidimensional.



Las incrustaciones de palabras son fundamentales para el procesamiento del lenguaje natural (PNL) porque hacen lo siguiente:

- Capture las relaciones semánticas entre palabras
- Permita la generación de texto relevante desde el punto de vista contextual
- Impulse los modelos de lenguaje extensos (LLM) para producir respuestas similares a las humanas

Descripción general de las bases de datos vectoriales

Una base de datos vectorial es un sistema especializado que almacena y consulta vectores de alta dimensión de manera eficiente. Estas bases de datos son fundamentales para las aplicaciones de generación aumentada de recuperación (RAG).

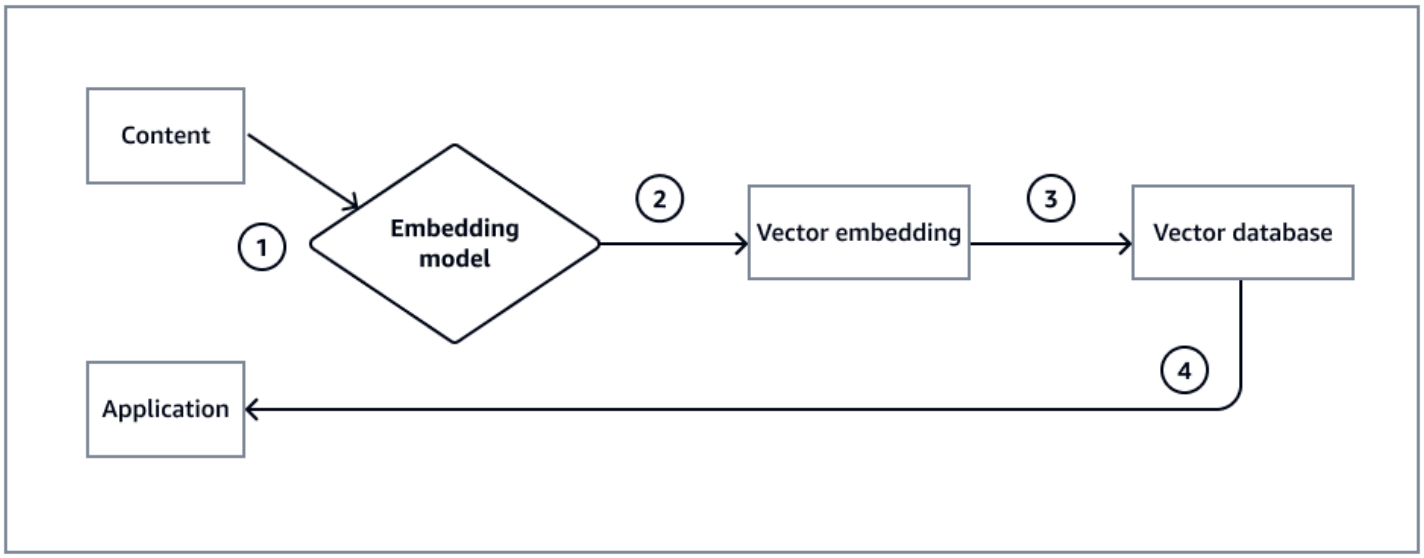
Las bases de datos vectoriales gestionan la conversión y el almacenamiento de datos de las siguientes maneras:

- Los objetos (como archivos de audio, imágenes y texto) se convierten en vectores mediante modelos de incrustación.
- Los vectores se almacenan en formatos de datos especializados.
- Las bases de datos vectoriales permiten realizar búsquedas rápidas de similitudes.

Las bases de datos vectoriales ofrecen varias ventajas clave con respecto a las bases de datos tradicionales, lo que las hace especialmente adecuadas para los desafíos de datos modernos. Están optimizadas específicamente para operaciones vectoriales y gestionan datos de alta dimensión de forma eficiente. También se especializan en las búsquedas por similitud que suelen hacer las bases de datos tradicionales. Más allá de estas capacidades básicas, las bases de datos vectoriales están diseñadas para satisfacer las demandas cambiantes de las aplicaciones de aprendizaje automático y de IA generativa. Destacan en el almacenamiento vectorial a gran escala y utilizan la computación distribuida para equilibrar las cargas de trabajo en varios nodos. Esto proporciona escalabilidad y rendimiento a medida que aumentan los volúmenes de datos.

El siguiente diagrama muestra una implementación de RAG:

1. El contenido, como documentos, archivos PDF o archivos de texto, se introduce en el modelo de incrustación como datos sin procesar para su procesamiento.
2. El modelo de incrustación transforma los datos sin procesar en vectores numéricos, que representan el significado semántico del contenido.
3. Las incrustaciones vectoriales generadas se almacenan en una base de datos vectorial optimizada para el almacenamiento y la recuperación de vectores de alta dimensión.
4. Las aplicaciones ahora pueden consultar la base de datos vectorial en respuesta a casos de uso como la búsqueda semántica y la recomendación de contenido.



La elección de una base de datos vectorial inadecuada para una solución RAG puede provocar importantes dificultades y limitaciones, entre las que se incluyen las siguientes:

- Rendimiento deficiente de las consultas
- Cuellos de botella en la escalabilidad
- Retos de la ingestión de datos
- Falta de funciones avanzadas, como el filtrado y la clasificación
- Dificultades de integración con otros sistemas
- Problemas de persistencia y durabilidad
- Problemas de concurrencia y coherencia en entornos con varios usuarios
- Costes de licencia más altos o dependencia de un proveedor
- El apoyo y los recursos de la comunidad son limitados
- Posibles riesgos de seguridad y cumplimiento

Opciones de bases de datos vectoriales

AWS ofrece una amplia gama de soluciones de bases de datos vectoriales para respaldar diferentes casos de uso y requisitos en las aplicaciones de IA generativa. Estas opciones se pueden clasificar en términos generales en servicios de bases de datos individuales y ofertas de servicios gestionados, cada uno con características y ventajas distintas. Comprender estas opciones es crucial para las organizaciones que desean implementar las capacidades de búsqueda vectorial de manera eficaz y, al mismo tiempo, mantener un rendimiento, una escalabilidad y una rentabilidad óptimos.

Para obtener más información sobre las soluciones de bases de datos vectoriales, consulte las siguientes secciones:

- Opciones de bases de datos vectoriales individuales
- Opción de servicio gestionado
- Elegir la base de datos vectorial adecuada

Opciones de bases de datos vectoriales individuales

Las opciones de bases de datos vectoriales individuales disponibles AWS incluyen Amazon Kendra, Amazon OpenSearch Service, AmazonRDS for PostgreSQL con pgvector, Amazon MemoryDB, Amazon DocumentDB, AmazonNeptune Analytics y Amazon S3 Vector. (Una extensión de código abierto, pgvector añade la capacidad de almacenar y buscar incrustaciones vectoriales generadas por ML). Estas soluciones ofrecen diferentes enfoques de búsqueda vectorial, lo que permite a las organizaciones elegir en función de la infraestructura existente, los requisitos técnicos y los casos de uso específicos.

Amazon Kendra

Amazon Kendra es un servicio de búsqueda inteligente de nivel empresarial que utiliza el procesamiento del lenguaje natural y algoritmos avanzados de aprendizaje automático para devolver respuestas específicas a las preguntas de búsqueda a partir de sus datos. Amazon Kendra simplifica la implementación de la funcionalidad de búsqueda y la convierte en una solución de backend eficaz para aplicaciones de IA generativa.

Otras características clave de Amazon Kendra son las siguientes:

- Conexiones nativas a más de 40 fuentes de datos

- Capacidades de preparación de datos integradas
- Configuración rápida que no requiere una gran experiencia técnica

Los beneficios de Amazon Kendra incluyen los siguientes

- Procesamiento automatizado de datos (fragmentación, ingestión, recuperación)
- Potentes opciones de personalización:
  - Búsqueda por facetas
  - Análisis de búsquedas
  - Ajustar la relevancia de las búsquedas
- Acceso programático sencillo a través del AWS SDK para Python (Boto3)

Para obtener más información, consulte Ventajas de Amazon Kendra en la documentación de Amazon Kendra.

OpenSearch Servicio Amazon

Amazon OpenSearch Service es un servicio gestionado que le ayuda a implementar, operar y escalar clústeres de OpenSearch servicios en Nube de AWS.

Las principales capacidades del OpenSearch servicio incluyen las siguientes:

- Motor de búsqueda y análisis de código abierto
- Arquitectura distribuida
- Procesamiento de datos en tiempo real

Algunas de las ventajas de utilizar el OpenSearch Servicio son las siguientes:

- Escalabilidad horizontal
- RESTful Soporte de API
- Maneja datos estructurados y no estructurados
- Análisis en tiempo real
- Adecuado para varios tamaños de despliegue

Para obtener más información, consulta Características de Amazon OpenSearch Service en la documentación del OpenSearch servicio.

Amazon RDS para PostgreSQL con pgvector

Amazon RDS for PostgreSQL con pgvector AWS combina el servicio de bases de datos relacionales gestionadas con la extensión de procesamiento vectorial de PostgreSQL. Esta combinación permite a las organizaciones almacenar y consultar vectores de alta dimensión y, al mismo tiempo, mantener Amazon RDS. La solución es especialmente adecuada para aplicaciones de IA generativa que requieren operaciones vectoriales en tiempo real sin la sobrecarga de administrar la infraestructura de bases de datos.

Entre las principales ventajas de Amazon RDS for PostgreSQL con pgvector se incluyen las siguientes:

- Alta disponibilidad
- Conmutación por error automática
- Rentable () pay-per-use
- Monitorización integrada
- Integración de datos vectoriales en tiempo real

Para obtener más información, consulte Ventajas de Amazon RDS en la documentación de Amazon RDS.

Amazon DocumentDB

Amazon DocumentDB (compatible con MongoDB) es una base de datos de documentos que ofrece funciones de búsqueda vectorial nativas en la versión 5.0 y versiones posteriores. Combina la flexibilidad del almacenamiento de documentos basado en JSON con la búsqueda vectorial y es compatible con los métodos de indexación jerárquica navegable pequeña (HNSW) e inverted File Flat (). IVFFlat

Entre las principales funciones de Amazon DocumentDB se incluyen las siguientes:

- Almacene e indexe vectores de hasta 2000 dimensiones (hasta 16 000 dimensiones sin indexación)
- Tiempos de respuesta en milisegundos para búsquedas de similitud vectorial

- Support para métricas de distancia entre productos euclidianos, cosenoidales y puntuales
- Integración perfecta con las aplicaciones compatibles con MongoDB existentes

Utilice Amazon DocumentDB en las siguientes situaciones:

- Para aplicaciones que ya utilizan APIs MongoDB y que necesitan capacidades de búsqueda vectorial
- Para casos de uso que requieren estructuras de datos de documentos flexibles combinadas con una búsqueda semántica
- Para escenarios que requieren tanto consultas de documentos tradicionales como búsquedas de similitud vectorial
- Para aplicaciones que ofrecen recomendaciones de productos, personalización, asistentes de chat y detección de fraudes

Para obtener más información, consulte Búsqueda vectorial para Amazon DocumentDB en la documentación de Amazon DocumentDB.

Amazon MemoryDB

Amazon MemoryDB es una base de datos en memoria compatible con Redis que ofrece el rendimiento de búsqueda vectorial más rápido entre las bases de datos vectoriales más populares. AWS Proporciona latencias de consulta inferiores a un milisegundo con una durabilidad en varias zonas de disponibilidad.

Las principales capacidades de MemoryDB incluyen las siguientes:

- Almacene datos de aplicaciones y millones de vectores en una única base de datos
- Tiempos de respuesta de consultas y actualizaciones en milisegundos de un solo dígito
- Las tasas de recuperación más altas con el rendimiento más rápido en AWS
- Support para hasta 32.768 dimensiones por vector
- Capacidades de búsqueda semántica y almacenamiento en caché en tiempo real

Utilice MemoryDB en las siguientes situaciones:

- Para aplicaciones en tiempo real que requieren una latencia ultrabaja (inferior a 10 ms)

- Para cargas de trabajo de alto rendimiento con millones de solicitudes al día
- Para casos de uso como los motores de recomendaciones en tiempo real, el almacenamiento semántico en caché y la detección de anomalías
- Para aplicaciones que necesitan capacidades tanto de almacenamiento de datos en memoria como de búsqueda vectorial

Para obtener más información, consulte Búsqueda vectorial en la documentación de MemoryDB.

Análisis por Amazon Neptune

Amazon Neptune Analytics es un motor de análisis de gráficos que ofrece funciones de búsqueda vectorial nativas, lo que lo hace ideal para los casos de uso de generación aumentada de recuperación de gráficos (GraphRag). Combina la búsqueda por similitud vectorial con algoritmos y recorridos de gráficos.

Entre las principales funciones de Neptune Analytics se incluyen las siguientes:

- Analice decenas de miles de millones de relaciones en cuestión de segundos
- Combine la búsqueda vectorial con algoritmos gráficos (búsqueda de rutas, detección de comunidades, centralidad)
- Support para aplicaciones GraphRag con conocimientos topológicos
- Hasta 80 veces más rápido que las soluciones analíticas gráficas existentes
- Integración con Amazon Bedrock para una gestión completa de GraphRag

Utilice Neptune Analytics en las siguientes situaciones:

- Para aplicaciones de GraphRag que requieren conocimientos: gráficos con incrustaciones vectoriales
- Para casos de uso que requieren atravesar relaciones complejas junto con similitudes vectoriales
- Para aplicaciones que requieren respuestas de IA explicables con un contexto relacional
- Para escenarios como las vistas de 360 grados de los clientes, las redes de detección de fraudes y el descubrimiento de conocimientos

Para obtener más información, consulte la documentación de Amazon Neptune Analytics.

Amazon S3 Vectors

Amazon S3 Vectors es el primer almacén de objetos en la nube AWS con capacidades nativas de almacenamiento vectorial y consulta. Proporciona almacenamiento vectorial diseñado específicamente y con costes optimizados para aplicaciones de IA que requieren una escalabilidad masiva.

Entre las principales capacidades de Amazon S3 Vectors se incluyen las siguientes:

- Almacenamiento para hasta 2 mil millones de vectores por índice con soporte para hasta 10 000 índices por segmento vectorial
- Latencia de consulta inferior a 100 ms optimizada para el almacenamiento a largo plazo y los patrones de acceso poco frecuentes
- Reducción de costes de hasta un 90% en las operaciones vectoriales en comparación con las bases de datos vectoriales especializadas
- Arquitectura sin servidor con escalado automático y una durabilidad del 99,99999% (11 9 s)

Utilice Amazon S3 Vectors en las siguientes situaciones:

- Para aplicaciones que requieren almacenar miles de millones de vectores a un costo mínimo
- Para cargas de trabajo que toleran una latencia de consulta inferior a un segundo (100 ms o más) en lugar de una latencia inferior a 10 ms
- Para casos de uso de archivado y retención vectorial a largo plazo
- Para aplicaciones RAG con patrones de recuperación poco frecuentes
- Para organizaciones que priorizan la economía del almacenamiento por encima de la latencia ultrabaja

Amazon S3 Vectors se integra de forma nativa con las bases de conocimiento de Amazon Bedrock y funciona bien en arquitecturas escalonadas con Amazon Service. OpenSearch Puede usar Amazon S3 Vectors para almacenamiento en frío y usar OpenSearch Service para consultas urgentes.

Para obtener más información, consulte Trabajar con vectores y cubos vectoriales de S3 en la documentación de Amazon S3.

Opción de servicio gestionado

Amazon Bedrock Knowledge Bases representa el enfoque AWS totalmente gestionado para la implementación de bases de datos vectoriales. La flexibilidad del servicio en cuanto a las opciones de almacenamiento, combinada con sus funciones de administración automatizada, lo hacen especialmente valioso para las organizaciones que buscan implementar RAG sin administrar una infraestructura compleja.

Con las bases de conocimiento de Amazon Bedrock, puede crear, mantener y consultar bases de conocimiento que mejoren sus modelos básicos mediante RAG. Este servicio simplifica el complejo proceso de implementación de RAG al administrar todo el proceso de ingesta, vectorización y recuperación de datos.

Entre las principales ventajas de las bases de conocimiento de Amazon Bedrock se incluyen las siguientes:

- Procesamiento de datos simplificado
  - Ingestión y fragmentación automáticas de datos
  - Extracción de texto integrada de varios formatos de archivo
  - Generación gestionada de incrustaciones vectoriales
  - Extracción e indexación automáticas de metadatos
- Implementación simplificada de RAG
  - Estrategias de recuperación preconfiguradas
  - Optimización automática de la ventana de contexto
  - Ajuste de relevancia incorporado
  - Capacidades de búsqueda semántica listas para usar
- Seguridad y gobernanza
  - Controles integrados AWS Identity and Access Management (IAM)
  - Cifrado de datos en reposo y en tránsito
  - Compatibilidad con VPC
  - Audite el registro con AWS CloudTrail

Las bases de conocimiento de Amazon Bedrock admiten varias opciones de almacenamiento vectorial, entre las que se incluyen:

- Amazon Aurora PostgreSQL con pgvector
- Análisis por Amazon Neptune
- Amazon EMR sin servidor
- Amazon S3 Vectors
- Cono de pino
- Redis Enterprise Cloud

Este servicio gestionado gestiona la ingesta, vectorización y recuperación automatizadas de datos. Esto simplifica las implementaciones de RAG.

Para obtener información detallada sobre cada almacén de vectores compatible, consulte la documentación de las bases de conocimiento de Amazon Bedrock.

Cómo elegir la base de datos vectorial adecuada

Seleccione su base de datos vectorial en función de estos factores de decisión clave:

- Si necesita una base de datos de documentos compatible con MongoDB con búsqueda vectorial, elija Amazon DocumentDB. Esto es ideal cuando su aplicación usa APIs MongoDB y desea agregar capacidades de búsqueda semántica sin administrar una infraestructura vectorial separada.
- Si necesita una latencia ultrabaja para aplicaciones en tiempo real, elija Amazon MemoryDB. Esto proporciona el rendimiento de búsqueda vectorial más rápido AWS con tiempos de respuesta inferiores a un milisegundo. Es ideal para motores de recomendaciones en tiempo real y aplicaciones de alto rendimiento.
- Si necesita representaciones del conocimiento basadas en gráficos con búsqueda vectorial, elija Amazon Neptune Analytics. Esto es lo mejor para las aplicaciones de GraphRag que necesitan atravesar relaciones complejas y realizar consultas basadas en gráficos junto con búsquedas vectoriales, lo que proporciona respuestas de IA explicables.
- Si necesita combinar consultas relacionales con búsquedas vectoriales, elija Amazon Aurora PostgreSQL con pgvector. Esta opción es ideal cuando la aplicación requiere tanto operaciones de SQL tradicionales como búsquedas de similitud vectorial en la misma base de datos.
- Si necesitas consultas de alto rendimiento con una latencia inferior a 10 ms, elige Amazon Service. OpenSearch Se destaca en el manejo de consultas de alta frecuencia y aplicaciones en tiempo real, e incluye mejoras recientes en la aceleración de la GPU.

- Si necesita almacenar miles de millones de vectores de forma rentable, elija Amazon S3 Vectors. Esta opción ofrece un ahorro de costos de hasta un 90% y es ideal para aplicaciones con patrones de recuperación poco frecuentes (minutos u horas entre consultas) que pueden tolerar una latencia inferior a los 100 ms.
- Si necesita una búsqueda de texto completo junto con una búsqueda vectorial, elija Amazon OpenSearch Service. Esta opción combina potentes capacidades de búsqueda de texto completo con búsqueda vectorial en una sola plataforma.

Comparación de bases de datos vectoriales

AWS proporciona varios enfoques para implementar capacidades de búsqueda vectorial, que van desde bases de datos vectoriales individuales hasta Amazon Bedrock Knowledge Bases, que es un servicio totalmente gestionado. Al evaluar estas opciones, las organizaciones deben tener en cuenta varios aspectos, como la arquitectura, la escalabilidad, las capacidades de integración, las características de rendimiento y las características de seguridad.

Bases de datos vectoriales individuales

La siguiente tabla proporciona una descripción general de las características clave de varias soluciones de bases de datos vectoriales AWS individuales, centrándose en sus arquitecturas, capacidades de escalado, integraciones de fuentes de datos y características de rendimiento.

Característica	Amazon Kendra	OpenSearch Servicio Amazon	Amazon RDS para SQLwith Postgre pgvector	Amazon DocumentDB	Amazon MemoryDB	Análisis de Amazon Neptune	Amazon S3 Vectors
Caso de uso principal	Búsqueda empresarial y RAG	Búsqueda y análisis distribuidos	Base de datos relacional con soporte vectorial	Base de datos de documentos con búsqueda vectorial	Búsqueda vectorial en memoria en tiempo real	Análisis de gráficos con búsqueda vectorial	Almacenamiento vectorial con costes optimizados
Arquitectura	Totalmente administrado	Clúster distribuido	Base de datos relacional	Orientado a documentos	Base de datos en memoria	Motor de análisis gráfico	Almacenamiento de objetos sin servidor

Modelo de datos	Basado en documentos	Documentos JSON	Tablas relacionales	Documentos JSON	Valor clave con JSON	Gráfico de propiedades	Almacenamiento de objetos
Dimensiones vectoriales	Administrado automáticamente	Hasta 16.000	Configurable	Hasta 2000 (indexados); 16.000 (sin indexar)	Hasta 32.768	Configurable	Hasta 4.096
Métodos de indexación	Automático	NSW, SI	HNSW, IVFFlat	HNSW, IVFFlat	HNSW	Gráfico y vector nativos	Automático
Métricas de distancia	Automático	Coseno, euclidiano, producto puntual	Coseno, euclidiano, producto interno	Coseno, euclidiano, producto puntiagudo	Coseno, euclidiano, producto interno	Coseno, euclidiano	Coseno, euclidiano
Latencia de consulta	Subsegundo	Menos de 10 ms (acelerada por la GPU)	10-100 ms	Milisegundos	Inferior a un milisegundo	Subsegundo	Menos de 100 ms
Modelo de escalado	Automático	Horizontal (añadir nodos)	Réplicas verticales y de lectura	Horizontal (añadir instancias)	Vertical y réplicas	Automático	Automático (sin servidor)

Número máximo de vectores	Administrado	Miles de millones (depende del clúster)	Millones (depende de la instancia)	Millones por colección	Millones por base de datos	Miles de millones	2 mil millones por índice; 10 000 índices por segmento
Rendimiento	Alto	Muy alto (miles de QPS)	Medio	Alto	Muy alto (millones de solicitudes por día)	Alto	Medio (optimizado para consultas poco frecuentes)
Duración de los datos	99,999999999 % (11 9s)	Configurable con réplicas	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99% (Multi-AZ)	99,99%	99,999999999 % (11 9s)
Modelo de consistencia	Eventual	Eventual (configurable)	Fuerte (ACID)	Eventual	Fuerte	Fuerte	Fuerte
Capacidades adicionales	40 o más conectores de datos, NLP	Búsqueda de texto completo, análisis, paneles	Consultas SQL, transacciones ACID	Compatibilidad con la API de MongoDB	Compatibilidad con la API de Redis, almacenamiento en caché	Algoritmos gráficos, recorridos	Políticas de ciclo de vida e integración de Amazon S3

Modelo de precios	Pague por consulta y almacenamiento	Horas de instancia y almacenamiento	Horas y almacenamiento de la instancia	Horas y almacenamiento de la instancia	Horas y almacenamiento de la instancia	Unidades de capacidad y almacenamiento	Almacenamiento, consultas y transferencia de datos
Optimización de costos	Basado en el uso	Instancias reservadas, autoescalado	Instancias reservadas, Aurora Serverless	Instancias reservadas	Instancias reservadas	Escalado automático	Hasta un 90% de ahorro en comparación con las especializadas DBs
Lo mejor para	Búsqueda empresarial con una configuración mínima	Consultas de alto rendimiento y baja latencia	Cargas de trabajo vectoriales y de SQL híbridas	Aplicaciones compatibles con MongoDB que necesitan vectores	Aplicaciones en tiempo real y de latencia ultrabaja	GraphRag y gráficos de conocimiento	Almacenamiento rentable y a largo plazo
Patrón de consulta ideal	Búsquedas empresariales frecuentes	Consultas de alta frecuencia en tiempo real	Consultas vectoriales y de SQL mixtas	Consultas de documentos con búsqueda semántica	Millones de solicitudes al día	Grafica los recorridos con búsqueda vectorial	Consultas poco frecuentes (de minutos a horas)

Complejidad de la configuración	Bajo (totalmente gestionado)	Medio (configuración de clúster)	Medio (configuración de extensión)	Medio (configuración de clúster)	Medio (configuración de clúster)	Bajo (totalmente gestionado)	Bajo (sin servidor)
Se requiere experiencia en equipo	Minimal	OpenSearch o Elasticsearch	PostgreSQL, SQL	MongoDB	Redis	Bases de datos gráficas	Amazon S3, conceptos vectoriales básicos

Servicio gestionado — Amazon Bedrock Knowledge Bases

Amazon Bedrock Knowledge Bases ofrece una solución totalmente gestionada con múltiples opciones de almacenamiento vectorial. En la siguiente tabla se comparan estas opciones de almacenamiento.

Característica	Aurora Postgre pgvector SQLwith	Análisis de Neptune	OpenSearch Servicio sin servidor	Vectores de Amazon S3	Pinecone	RedisEnterprise Cloud
Caso de uso principal	Base de datos relacional con RAG vectorial	Búsqueda vectorial basada en gráficos para GraphRag	Gestión del conocimiento RAG	RAG vectorial optimizado en costes	Búsqueda vectorial de alto rendimiento	Búsqueda vectorial en memoria
Arquitectura	Relacional totalmente gestionado	Análisis de gráficos totalmente gestionado	Sin servidor totalmente gestionado	Almacenamiento de objetos sin servidor	Nube híbrida totalmente gestionada	Almacenamiento en memoria totalmente gestionado

Modelo de datos	Tablas relacionales	Gráfico de propiedades	Documentos JSON	Almacenamiento de objetos	Vectores diseñados específicamente	Valor clave con vectores
Almacenamiento vectorial	A través de la extensión pgvector	Vectores gráficos nativos	A través OpenSearch del motor	Almacenamiento vectorial nativo de Amazon S3	Base de datos vectorial nativa	Vectores en memoria
Integración de Amazon Bedrock	KCL	KCL	KCL	KCL	KCL	KCL
Ingestión automática	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)
Vectorización automática	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)	Sí (a través de Amazon Bedrock)
Escalado	Escalado automático (Aurora Serverless)	Escalado automático de gráficos	Automático sin servidor	Automático (miles de millones de vectores)	Cápsulas de escalado automático	Clústeres con escalado automático
Rendimiento de las consultas	Alto para relacionales o vectoriales	Alto para vectores gráficos	Alto	Medio (latencia de 100 ms o más)	Muy alta	Muy alta
Vectores máximos	Millones (en función de la instancia)	Miles de millones	Miles de millones	2 mil millones por índice	Miles de millones	Millones (depende de la memoria)

Capacidades adicionales	Consultas SQL, transacciones ACID	Algoritmos gráficos, recorridos	Búsqueda de texto completo, análisis	Ciclo de vida de Amazon S3, organización en niveles	Filtrado de metadatos, espacios de nombres	Estructuras de datos de Redis, almacenamiento en caché
Optimización de costos	Moderado (Aurora sin servidor)	Moderado (unidades de capacidad)	Alto (sin servidor, pay-per-use)	Muy alto (hasta un 90% de ahorro)	Moderado (precios basados en cápsulas)	Bajo (versión premium en memoria)
Lo mejor para	Cargas de trabajo híbridas SQL/vector	Gráficos de conocimiento conectados	Texto completo con búsqueda vectorial	Vectores de acceso poco frecuente y a largo plazo	Búsqueda vectorial a escala en tiempo real	Necesidades de latencia ultrabaja
Patrón de consulta ideal	Consultas vectoriales y de SQL mixtas	Grafique los recorridos con vectores	Búsquedas frecuentes con análisis	Recuperación poco frecuente (de minutos a horas)	Consultas de alta frecuencia en tiempo real	Millones de solicitudes por segundo
Configuración con Amazon Bedrock	Simple (gestionado por Amazon Bedrock)	Simple (gestionado por Amazon Bedrock)	Simple (gestionado por Amazon Bedrock)	Simple (gestionado por Amazon Bedrock)	Simple (gestionado por Amazon Bedrock)	Simple (gestionado por Amazon Bedrock)

Residencia de datos	Regiones de AWS	Regiones de AWS	Regiones de AWS	Regiones de AWS	Nube múltiple (AWS y otras)	Nube múltiple (AWS y otras)
Modelo de precios	Horas de instancia y almacenamiento	Unidades de capacidad y almacenamiento	Computación y almacenamiento (sin servidor)	Almacenamiento, consultas y transferencia	Horario y almacenamiento del pod	Horas y almacenamiento de los nodos

Elegir entre opciones individuales y administradas

Consideración	Elija una base de datos vectorial individual	Elija las bases de conocimiento de Amazon Bedrock (administradas)
Implementación de RAG	¿Desea tener el control total sobre la canalización de RAG	Quiere un RAG totalmente gestionado con una configuración mínima
Personalización	Necesita una lógica de recuperación y un preprocesamiento personalizados	Los patrones RAG estándar satisfacen sus necesidades
La infraestructura existente	Ya tiene la base de datos implementada	Está empezando de cero o desea una administración simplificada
Experiencia del equipo	Su equipo tiene experiencia en administración de bases de datos	Prefiere centrarse en la lógica de las aplicaciones, no en la infraestructura

Complejidad de integración	Necesita una integración profunda con los sistemas existentes	¿Quieres una integración rápida con los modelos de Amazon Bedrock?
Gastos generales operativos	Puede administrar las operaciones de la base de datos	Quiere AWS gestionar las operaciones
Estructura de costos	Prefiere los precios directos de las bases de datos	Prefieres los precios unificados de Amazon Bedrock
¿Hora de comercialización	Tiene tiempo para una implementación personalizada	Necesita un despliegue rápido

Consideraciones y comparaciones de costos

Comprender la estructura de costos de las diferentes opciones de bases de datos vectoriales es esencial para tomar decisiones de implementación informadas. En la siguiente tabla se describen algunas consideraciones clave sobre los costos de varias soluciones de bases de datos vectoriales, incluidas las bases de datos individuales y los servicios gestionados. Cada opción tiene factores de precio distintos que pueden afectar al costo total de propiedad (TCO), desde pay-as-you-go los modelos hasta los costos operativos y de infraestructura.

Base de datos vectorial	Modelo de costes	Consideraciones sobre costos
Amazon Kendra	Pague por uso, en función de las consultas	Los costes pueden variar en función del número de consultas y la cantidad de datos indexados. Se pueden aplicar cargos adicionales por el almacenamiento y la transferencia de datos. Para obtener más informaci ón, consulta los precios de Amazon Kendra.
OpenSearch Servicio Amazon	Pague por uso, según las horas de instancia y el almacenamiento	Los costos incluyen las horas de instancia, el almacenam iento (volúmenes de Amazon EBS), la transferencia de datos y el UltraWarm almacenamiento opcional. Las instancias reservadas pueden ofrecer hasta un 30% de ahorro. Las instancias aceleradas por GPU ofrecen una mejor relación precio-rendimiento para las cargas de trabajo vectoriales. Para obtener más informaci

		ón, consulta los precios OpenSearch de Amazon Service.
Código abierto OpenSearch	De código abierto, sin costo directo (no tiene que pagar para descargar o usar el software, y no hay costos de licencia)	Los costos incluyen la infraestructura (como los servidores y el almacenamiento) y los costos operativos (como el mantenimiento y la supervisión). Las organizaciones necesitan presupuestar el personal necesario para administrar y mantener la infraestructura.
Amazon RDS para PostgreSQL con pgvector	Pague por uso, en función del uso	Los costos incluyen los tipos de instancias de bases de datos, el almacenamiento, la transferencia de datos y las copias de seguridad. Se pueden aplicar cargos adicionales por la transferencia de datos, los tipos de instancias y el almacenamiento más allá del nivel AWS gratuito. Para obtener más información, consulte Precios de Amazon RDS.

Amazon DocumentDB	Pague por uso, según el horario de las instancias y el almacenamiento	Los costos incluyen las horas de instancia, el almacenamiento (GB al mes), I/O las solicitudes, el almacenamiento de copias de seguridad y la transferencia de datos. Los clústeres elásticos permiten el escalado dinámico. Las instancias reservadas están disponibles para la optimización de costos. Para obtener más información, consulte Precios de Amazon DocumentDB.
Amazon MemoryDB	Pague por uso, en función del horario de los nodos y del almacenamiento de datos	Los costos incluyen las horas de los nodos (por tipo de nodo), el almacenamiento de datos (GB-hora), el almacenamiento de instantáneas y la transferencia de datos. Los nodos reservados pueden ofrecer hasta un 55% de ahorro. Optimizado para cargas de trabajo de alto rendimiento y baja latencia. Para obtener más información, consulte los precios de Amazon MemoryDB.

Análisis por Amazon Neptune	Pague por uso, en función de las unidades de capacidad	Los costos incluyen las unidades de capacidad de Neptune (NCUs), el almacenamiento (GB al mes) y la transferencia de datos. Escalamiento automático en función de la carga de trabajo sin compromisos iniciales . Se requiere un mínimo de 128 NCUs . Para obtener más información, consulte los precios de Amazon Neptune.
Amazon S3 Vectors	Pague por uso, en función del almacenamiento y las solicitudes	Los costos incluyen el almacenamiento (GB al mes), las solicitudes PUT y GET, la administración de índices vectoriales y la transferencia de datos. Ofrece un ahorro de costes de hasta un 90% en comparación con las bases de datos vectoriales especializadas. Las políticas de ciclo de vida y estratificación inteligente de Amazon S3 están disponibles para una optimización adicional. Para obtener más información, consulte Precios de Amazon S3.

Bases de conocimiento de Amazon Bedrock	Pague por uso, en función del uso	Los costes pueden variar en función del uso de la base de conocimientos y de servicios adicionales, como Amazon OpenSearch Serverless. Se pueden aplicar cargos adicionales por el almacenamiento de datos, la transferencia de datos y las funciones adicionales. Para obtener más información sobre los precios, consulta los precios OpenSearch de Amazon Service.

Casos prácticos de bases de datos vectoriales

Los siguientes ejemplos destacan cómo se pueden utilizar eficazmente las diferentes opciones de bases de datos vectoriales para mejorar la gestión del conocimiento, mejorar la eficiencia operativa y ofrecer mejores resultados empresariales. Estos casos de uso ilustran las aplicaciones prácticas de las soluciones de bases de datos vectoriales analizadas anteriormente en esta guía y proporcionan información sobre su rendimiento y sus beneficios en el mundo real.

Gestión del conocimiento con Amazon Kendra

Problema con el cliente: uno de los mayores contratistas generales de Japón se enfrentaba a una disminución de personal experimentado. La empresa necesitaba una forma de transferir los conocimientos y las habilidades del personal experimentado a la generación más joven de manera eficiente. Necesitaban una solución para recopilar y difundir conocimientos complejos de ingeniería de la construcción y experiencias pasadas.

AWS solución: para abordar este problema, el cliente recurrió a Amazon Kendra, una solución de IA que podía gestionar de forma rápida y precisa su base de conocimientos interna y permitir consultas en lenguaje natural. Con Amazon Kendra, los empleados ahora pueden encontrar la información que necesitan mucho más rápido, lo que mejora la productividad y facilita la transferencia de conocimientos del personal experimentado al personal más joven.

Impacto: al implementar un chatbot de IA generativo impulsado por Amazon Kendra, la empresa creó una plataforma de conocimiento unificada. El chatbot permite a los empleados acceder rápidamente a los conocimientos técnicos y a las experiencias pasadas en ingeniería de la construcción. Esta solución ha mejorado significativamente la eficiencia de la transferencia de conocimientos y los procesos de toma de decisiones dentro de la organización, lo que ha ayudado a garantizar que se conserve la valiosa experiencia y que todos los empleados puedan acceder fácilmente a ella. El costo de esta solución puede variar en función del uso y la configuración. Para obtener una estimación de costos detallada, consulte la Calculadora de precios de AWS. Para obtener una estimación de los costos de las bases de datos vectoriales, consulte la sección Comparación de costos y consideraciones de esta guía o consulte los precios de Amazon Kendra.

Para obtener información sobre otros casos de uso de clientes, consulte Clientes de Amazon Kendra.

Análisis en tiempo real con Serverless OpenSearch

Problema del cliente: un proveedor líder de servicios financieros se enfrentó al desafío de administrar un enorme ecosistema de datos. Procesaba 300 millones de autorizaciones y 90 000 millones de transacciones al año, acumulando aproximadamente 1,1 petabytes (PB) de datos. El sistema existente, que prestaba servicio a 300 000 usuarios que necesitaban acceder a más de 6 000 informes, necesitaba modernizarse para ofrecer coherencia global y permitir la toma de decisiones en tiempo real.

AWS solución: la arquitectura de la solución utilizó modelos básicos disponibles en Amazon Bedrock (incluidos Anthropic, Sonnet 3, Sonnet 3.5 y Haiku) para el procesamiento del lenguaje natural. El cliente eligió OpenSearch Serverless como base de datos vectorial por su escalabilidad superior y su capacidad para gestionar el enorme volumen de datos de manera eficiente. Esta arquitectura permitía el procesamiento fluido de consultas complejas y la generación dinámica de informes.

Impacto: la implementación logró un aumento del 50 por ciento en la productividad al eliminar la necesidad de generar manualmente más de 100 paneles de inteligencia empresarial. Los usuarios ahora pueden generar informes mediante consultas en lenguaje natural con tiempos de respuesta de entre 20 y 40 segundos. El coste de esta solución puede variar en función del uso y la configuración. Para obtener una estimación de costos detallada, consulte la Calculadora de precios de AWS. Para obtener una estimación de los costes de las bases de datos vectoriales, consulta la sección Comparación de costes y consideraciones de esta guía o consulta los precios OpenSearch de Amazon Service. Para obtener información sobre otros casos de uso de clientes, consulte Amazon OpenSearch Serverless.

Próximos pasos y recursos

Tras revisar esta guía, considere las siguientes acciones para pasar de la comprensión a la implementación:

1. Evalúe sus necesidades actuales:
   - Evalúe su infraestructura de bases de datos y sus conocimientos actuales.
   - Documente sus requisitos específicos de búsqueda vectorial.
   - Defina sus objetivos de rendimiento, escalamiento y costes.
2. Elija una de las siguientes opciones para probar las opciones de bases de datos vectoriales:
   - Opción 1: configure una prueba de concepto con la solución de base de datos vectorial que prefiera.
   - Opción 2: Experimente con conjuntos de datos de muestra en las bases de conocimiento de Amazon Bedrock. Pruebe la experiencia de creación rápida de una base de conocimientos de Amazon Bedrock. Para ver un ejemplo, consulte Creación rápida de una base de conocimiento de Aurora PostgreSQL para Amazon Bedrock en la documentación de Aurora.
3. Consulte los recursos adicionales.
4. Obtenga la ayuda de un experto:
   - Póngase en contacto con su Cuenta de AWS equipo o con los arquitectos de AWS soluciones para obtener orientación sobre la implementación.
   - Póngase en contacto con AWS socios que se especializan en bases de datos vectoriales.
5. Planifique su despliegue de producción:
   - Cree una estrategia de migración si se muda de bases de datos existentes.
   - Desarrolle un plan de escalado para la solución que elija.
   - Diseñe sus procedimientos de supervisión y mantenimiento.

Recursos

Los siguientes recursos pueden ayudarle a elegir una base de datos vectorial.

AWS publicaciones de blog

- Acelere el desarrollo de aplicaciones de IA generativa con Amazon Bedrock Knowledge Bases, Quick Create y Amazon Aurora Serverless
- Explicación de las capacidades de las bases de datos vectoriales de Amazon OpenSearch Service
- Profundice en los almacenes de datos vectoriales con las bases de conocimiento de Amazon Bedrock
- Aproveche pgvector y Amazon Aurora PostgreSQL para el procesamiento de lenguaje natural, los chatbots y el análisis de opiniones

AWS documentación de servicio

- Elegir un servicio de AWS base de datos
- Cómo funcionan las bases de conocimiento de Amazon Bedrock
- Documentación de Neptune Analytics
- Descripción general de Amazon Web Services: bases de datos
- Uso de Aurora PostgreSQL como base de conocimientos para Amazon Bedrock
- Uso de Amazon Aurora PostgreSQL
- Amazon DocumentDB
- Amazon MemoryDB
- Amazon S3 Vectors

Otros recursos AWS

- Bases de conocimiento de Amazon Bedrock
- Bases de datos vectoriales e incrustaciones
- Bases de datos vectoriales para aplicaciones de IA generativa
- ¿Qué son las incrustaciones en Machine Learning?

Otros recursos de

- Acerca de PostgreSQL

- Documentación de pgvector
- Pinecone como base de conocimientos para Amazon Bedrock
- Redis Enterprise Cloud está activada AWS

Historial de documentos

En la siguiente tabla se describen los cambios más importantes en esta guía, titulada Cómo elegir una base de datos AWS vectorial para los casos de uso de RAG. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Añadido Servicios de AWS	Hemos agregado información sobre el uso de Amazon DocumentDB, Amazon MemoryDB, Amazon S3 Vectors y Amazon Neptune Analytics.	30 de marzo de 2026
Publicación inicial	—	6 de marzo de 2025

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.