



RAG 사용 사례를 위한 AWS 벡터 데이터베이스 선택

AWS 권장 가이드



AWS 권장 가이드: RAG 사용 사례를 위한 AWS 벡터 데이터베이스 선택

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

소개	1
대상 독자	1
벡터 개요	2
벡터 데이터베이스 개요	4
벡터 데이터베이스 옵션	6
개별 벡터 데이터베이스 옵션	6
Amazon Kendra	6
Amazon OpenSearch Service	7
pgvector를 사용하는 Amazon RDS for PostgreSQL	7
Amazon DocumentDB	8
Amazon MemoryDB	9
Amazon Neptune Analytics	9
Amazon S3 Vectors	10
관리형 서비스 옵션	11
올바른 벡터 데이터베이스 선택	12
벡터 데이터베이스 비교	13
개별 벡터 데이터베이스	13
관리형 서비스 - Amazon Bedrock 지식 기반	15
개별 옵션과 관리형 옵션 중에서 선택	18
비용 비교 및 고려 사항	19
벡터 데이터베이스 사용 사례	22
Amazon Kendra를 사용한 지식 관리	22
OpenSearch Serverless를 사용한 실시간 분석	22
다음 단계 및 리소스	24
리소스	24
AWS 블로그 게시물	24
AWS 서비스 설명서	25
기타 AWS 리소스	25
기타 리소스	25
문서 기록	26
.....	xxvii

RAG 사용 사례를 위한 AWS 벡터 데이터베이스 선택

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

2026년 3월([문서 기록](#))

벡터 데이터베이스는 생성형 AI 애플리케이션을 구현하는 조직에 점점 더 중요해지고 있습니다. 이러한 데이터베이스는 의미와 관계를 캡처하는 방식으로 텍스트, 이미지 및 기타 콘텐츠를 처리할 수 있는 데이터의 수치 표현인 벡터를 저장하고 관리합니다.

조직이 벡터 데이터베이스 옵션을 탐색 AWS할 때 다양한 솔루션의 기능, 장단점 및 모범 사례를 이해해야 합니다. 이 가이드는에서 일반적으로 사용되는 벡터 스토어를 비교 AWS 하고 특정 요구 사항 또는 [사용 사례에](#) 가장 적합한 옵션에 대해 정보에 입각한 결정을 내리는 데 도움이 됩니다. Retrieval Augmented Generation(RAG)을 구현하든, 추천 시스템을 구축하든, 다른 AI 애플리케이션을 개발하든 이 가이드는 벡터 데이터베이스 솔루션을 평가하고 선택하는 데 도움이 되는 프레임워크를 제공합니다.

대상 독자

이 가이드는 다음 역할의 사용자를 대상으로 합니다.

- 벡터 데이터베이스를 사용하여 ML 모델의 고차원 데이터를 저장하고 검색하는 데이터 과학자 및 기계 학습(ML) 엔지니어.
- 고차원 데이터를 저장하고 처리하기 위한 벡터 데이터베이스가 포함된 데이터 파이프라인을 설계하고 구현하는 데이터 엔지니어입니다.
- 벡터 데이터베이스를 ML 파이프라인의 일부로 사용하여 모델 출력 또는 중간 표현을 저장하고 제공하는 MLOps 엔지니어입니다.
- 유사성 검색 또는 권장 시스템이 필요한 애플리케이션에 벡터 데이터베이스를 통합하는 소프트웨어 엔지니어.
- 프로덕션 환경에서 벡터 데이터베이스를 배포하고 유지 관리하여 확장성과 신뢰성을 보장하는 DevOps 엔지니어.
- 벡터 데이터베이스를 사용하여 임베딩 또는 특성 벡터의 대규모 데이터 세트를 저장하고 분석하는 AI 연구원.
- 제품 기능 및 아키텍처에 대해 정보에 입각한 결정을 내리기 위해 벡터 데이터베이스의 기능과 제한 사항을 이해해야 하는 AI 제품 관리자.

벡터 개요

벡터는 머신이 데이터를 이해하고 처리하는 데 도움이 되는 수치 표현입니다. 생성형 AI에서는 두 가지 주요 목적을 수행합니다.

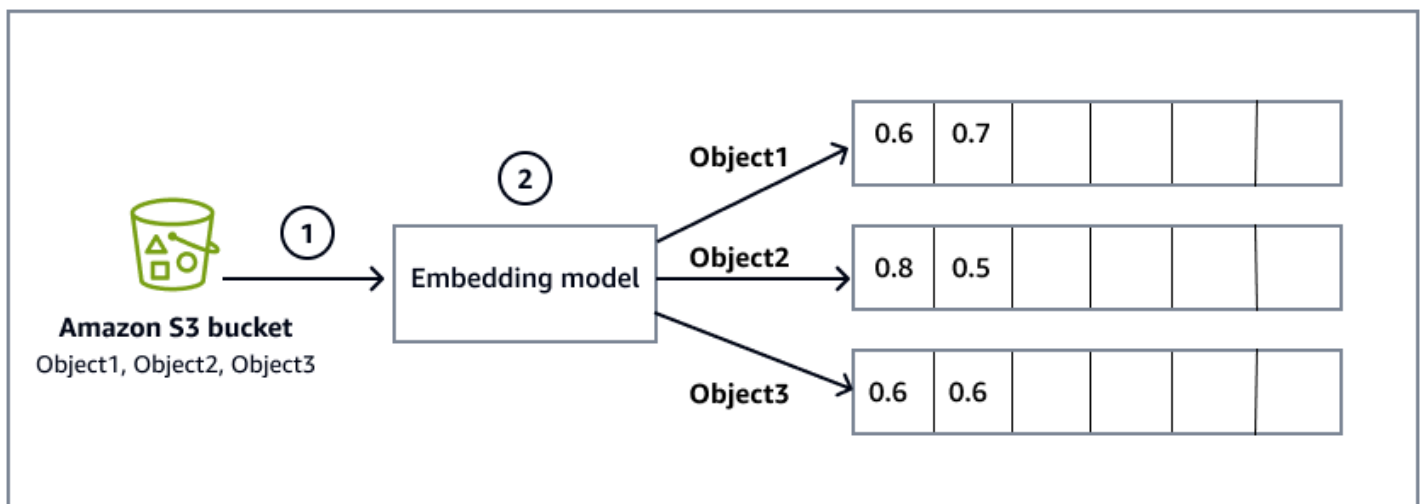
- 압축된 형식으로 데이터 구조를 캡처하는 잠재적 공간 표시
- 단어, 문장 및 이미지와 같은 데이터에 대한 임베딩 생성

[Word2Vec](#), [GloVe](#) 및 [Amazon Titan Text Embeddings](#)와 같은 임베딩 모델은 임베딩이라는 프로세스를 통해 데이터를 벡터로 변환합니다. 이러한 임베딩 모델은 다음을 수행할 수 있습니다.

- 컨텍스트에서 학습하여 단어를 벡터로 표현
- 벡터 공간에 비슷한 단어를 더 가깝게 배치
- 머신이 연속 공간에서 데이터를 처리할 수 있도록 지원

다음 다이어그램은 임베딩 프로세스에 대한 개략적인 개요를 제공합니다.

1. [Amazon Simple Storage Service\(Amazon S3\)](#) 버킷에는 시스템이 정보를 읽고 처리할 데이터 소스인 파일이 포함되어 있습니다. Amazon S3 버킷은 [Amazon Bedrock](#) 지식 기반 구성 중에 지정되며, 여기에는 [데이터를 지식 기반과 동기화하는](#) 것도 포함됩니다.
2. 임베딩 모델은 Amazon S3 버킷의 객체 파일에서 원시 데이터를 벡터 임베딩으로 변환합니다. 예를 들어 Object1는 다차원 공간에서 콘텐츠를 [0.6, 0.7, ...] 나타내는 벡터로 변환됩니다.



단어 임베딩은 다음을 수행하기 때문에 자연어 처리(NLP)에 매우 중요합니다.

- 단어 간 의미 관계 캡처
- 컨텍스트 관련 텍스트 생성 활성화
- 대규모 언어 모델(LLMs) 강화하여 인간과 유사한 응답 생성

벡터 데이터베이스 개요

벡터 데이터베이스는 고차원 벡터를 효율적으로 저장하고 쿼리하는 특수 시스템입니다. 이러한 데이터베이스는 검색 증강 생성(RAG) 애플리케이션의 기본입니다.

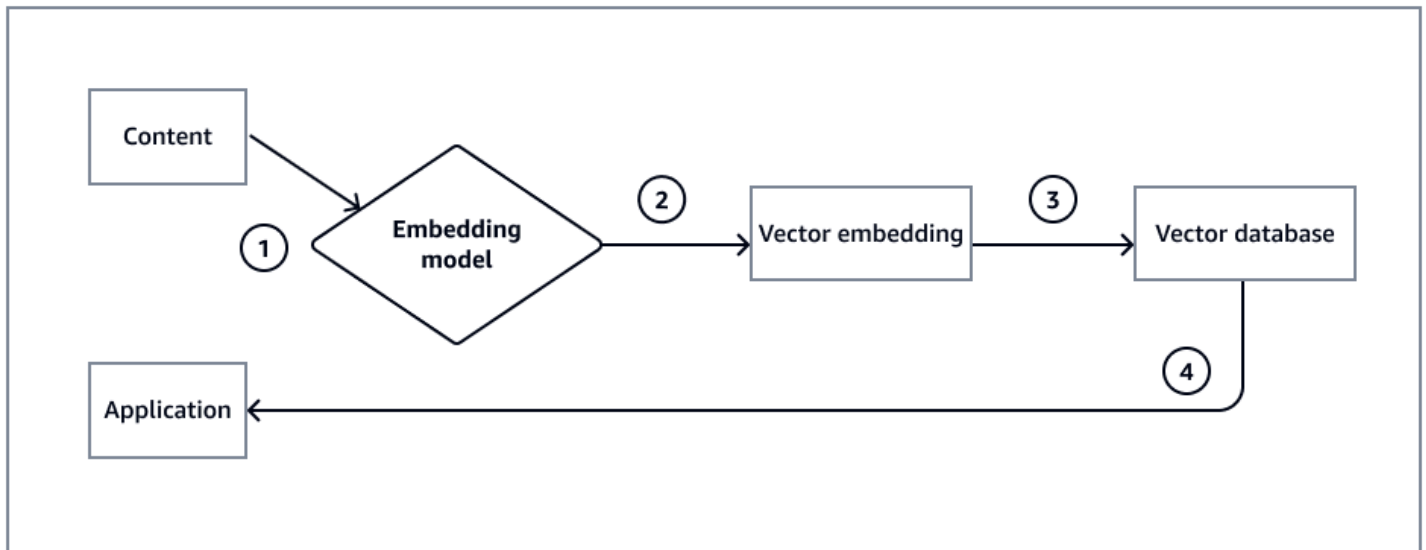
벡터 데이터베이스는 다음과 같은 방식으로 데이터 변환 및 스토리지를 처리합니다.

- 객체(예: 오디오, 이미지 및 텍스트 파일)는 임베딩 모델을 사용하여 벡터로 변환됩니다.
- 벡터는 특수 데이터 형식으로 저장됩니다.
- 벡터 데이터베이스를 사용하면 빠른 유사성 검색을 수행할 수 있습니다.

벡터 데이터베이스는 기존 데이터베이스보다 몇 가지 주요 이점을 제공하므로 최신 데이터 문제에 특히 적합합니다. 벡터 작업에 특히 최적화되어 있으며 고차원 데이터를 효율적으로 처리합니다. 또한 기존 데이터베이스에서 어려움을 겪는 유사성 검색을 전문으로 합니다. 이러한 핵심 기능 외에도 벡터 데이터베이스는 ML 및 생성형 AI 애플리케이션의 변화하는 요구 사항을 충족하도록 구축되었습니다. 대규모 벡터 스토리지에서 우수하며 분산 컴퓨팅을 사용하여 여러 노드에서 워크로드의 균형을 맞춥니다. 이렇게 하면 데이터 볼륨이 증가함에 따라 확장성과 성능이 제공됩니다.

다음 다이어그램은 RAG 구현을 보여줍니다.

1. 문서, PDFs 또는 텍스트 파일과 같은 콘텐츠는 처리를 위한 원시 데이터로 임베딩 모델에 제공됩니다.
2. 임베딩 모델은 원시 데이터를 콘텐츠의 의미론적 의미를 나타내는 숫자 벡터로 변환합니다.
3. 생성된 벡터 임베딩은 고차원 벡터의 저장 및 검색에 최적화된 벡터 데이터베이스에 저장됩니다.
4. 이제 애플리케이션은 의미 체계 검색 및 콘텐츠 권장 사항과 같은 사용 사례에 대한 응답으로 벡터 데이터베이스를 쿼리할 수 있습니다.



RAG 솔루션에 부적절한 벡터 데이터베이스를 선택하면 다음과 같은 심각한 문제와 제한 사항이 발생할 수 있습니다.

- 쿼리 성능 저하
- 확장성 병목 현상
- 데이터 수집 문제
- 필터링 및 순위와 같은 고급 기능 부족
- 다른 시스템과의 통합 문제
- 지속성 및 내구성 문제
- 여러 사용자가 있는 환경에서의 동시성 및 일관성 문제
- 더 높은 라이선스 비용 또는 공급업체 잠금
- 제한된 커뮤니티 지원 및 리소스
- 잠재적 보안 및 규정 준수 위험

벡터 데이터베이스 옵션

AWS는 생성형 AI 애플리케이션의 다양한 사용 사례 및 요구 사항을 지원하는 다양한 벡터 데이터베이스 솔루션을 제공합니다. 이러한 옵션은 개별 데이터베이스 서비스 및 관리형 서비스 상품으로 광범위하게 분류할 수 있으며, 각 서비스에는 고유한 특성과 장점이 있습니다. 이러한 옵션을 이해하는 것은 최적의 성능, 확장성 및 비용 효율성을 유지하면서 벡터 검색 기능을 효과적으로 구현하려는 조직에 매우 중요합니다.

벡터 데이터베이스 솔루션에 대한 자세한 내용은 다음 섹션을 참조하세요.

- [개별 벡터 데이터베이스 옵션](#)
- [관리형 서비스 옵션](#)
- [올바른 벡터 데이터베이스 선택](#)

개별 벡터 데이터베이스 옵션

의 개별 벡터 데이터베이스 옵션에는 [Amazon Kendra](#), [Amazon OpenSearch Service](#), pgvector를 사용하는 [Amazon RDS for PostgreSQL](#), [Amazon MemoryDB](#), [Amazon DocumentDB](#), [Amazon Neptune Analytics](#) 및 [Amazon S3 Vector](#)가 AWS 포함됩니다. (오픈 소스 확장인 pgvector는 ML 생성 벡터 임베딩을 저장하고 검색할 수 있는 기능을 추가합니다.) 이러한 솔루션은 벡터 검색에 대한 다양한 접근 방식을 제공하므로 조직은 기존 인프라, 기술 요구 사항 및 특정 [사용 사례](#)를 기반으로 선택할 수 있습니다.

Amazon Kendra

Amazon Kendra는 자연어 처리 및 고급 기계 학습 알고리즘을 사용하여 데이터의 검색 질문에 대한 특정 답변을 반환하는 엔터프라이즈급 지능형 검색 서비스입니다. Amazon Kendra는 검색 기능의 구현을 간소화하여 생성형 AI 애플리케이션을 위한 효과적인 백엔드 솔루션입니다.

Amazon Kendra의 기타 주요 기능은 다음과 같습니다.

- [40개 이상의 데이터 소스](#)에 대한 기본 연결
- 기본 제공 데이터 준비 기능
- 심층적인 기술 전문 지식이 필요하지 않은 빠른 설정

Amazon Kendra의 이점은 다음과 같습니다.

- 자동 데이터 처리(칭킹, 수집, 검색)
- 강력한 사용자 지정 옵션:
 - [패킷 검색](#)
 - [검색 분석](#)
 - [검색 관련성 조정](#)
- 를 통한 간단한 프로그래밍 방식 액세스 [AWS SDK for Python \(Boto3\)](#)

자세한 내용은 [Amazon Kendra 설명서의 Amazon Kendra의 이점](#)을 참조하세요.

Amazon OpenSearch Service

Amazon OpenSearch Service는에서 OpenSearch Service 클러스터를 배포, 운영 및 확장하는 데 도움이 되는 관리형 서비스입니다 AWS 클라우드.

OpenSearch Service의 핵심 기능은 다음과 같습니다.

- 오픈 소스 검색 및 분석 엔진
- 분산 아키텍처
- 실시간 데이터 처리

OpenSearch Service 사용의 몇 가지 이점은 다음과 같습니다.

- 수평 확장성
- RESTful API 지원
- 정형 및 비정형 데이터 처리
- 실시간 데이터 분석
- 다양한 배포 크기에 적합

자세한 내용은 [OpenSearch Service 설명서의 Amazon OpenSearch Service](#) 기능을 참조하세요.

pgvector를 사용하는 Amazon RDS for PostgreSQL

[pgvector](#)를 사용하는 Amazon RDS for PostgreSQL은 AWS 관리형 관계형 데이터베이스 서비스를 PostgreSQL의 벡터 처리 확장과 결합합니다. 이 조합을 통해 조직은 Amazon RDS를 유지하면서 고차

원 벡터를 저장하고 쿼리할 수 있습니다. 이 솔루션은 데이터베이스 인프라 관리 오버헤드 없이 실시간 벡터 작업이 필요한 생성형 AI 애플리케이션에 특히 적합합니다.

pgvector를 사용하는 Amazon RDS for PostgreSQL의 주요 이점은 다음과 같습니다.

- 높은 가용성
- 자동 장애조치(failover)
- 비용 효율성(pay-per-use)
- 기본 제공 모니터링
- 실시간 벡터 데이터 통합

자세한 내용은 [Amazon RDS 설명서의 Amazon RDS의 이점](#)을 참조하세요.

Amazon DocumentDB

Amazon DocumentDB(MongoDB 호환)는 버전 5.0 이상에서 기본 벡터 검색 기능을 제공하는 문서 데이터베이스입니다. JSON 기반 문서 스토리지의 유연성과 벡터 검색을 결합하여 계층적 탐색 가능 소규모 세계(HNSW) 및 반전 파일 플랫(IVFFlat) 인덱싱 방법을 모두 지원합니다.

Amazon DocumentDB의 핵심 기능은 다음과 같습니다.

- 최대 2,000개의 차원(인덱싱 없이 최대 16,000개의 차원)에 벡터 저장 및 인덱싱
- 벡터 유사성 검색을 위한 밀리초 응답 시간
- 유클리드, 코사인 및 점 제품 거리 지표 지원
- 기존 MongoDB 호환 애플리케이션과의 원활한 통합

다음과 같은 상황에서 Amazon DocumentDB를 사용합니다.

- 이미 MongoDB APIs 사용하고 벡터 검색 기능이 필요한 애플리케이션의 경우
- 의미 체계 검색과 결합된 유연한 문서 데이터 구조가 필요한 사용 사례의 경우
- 기존 문서 쿼리와 벡터 유사성 검색이 모두 필요한 시나리오의 경우
- 제품 추천, 개인 맞춤, 채팅 도우미 및 사기 탐지를 제공하는 애플리케이션의 경우

자세한 내용은 [Amazon DocumentDB 설명서의 Amazon DocumentDB에 대한 벡터 검색](#)을 참조하세요. Amazon DocumentDB

Amazon MemoryDB

Amazon MemoryDB는 널리 사용되는 벡터 데이터베이스 중에서 가장 빠른 벡터 검색 성능을 제공하는 Redis 호환 인 메모리 데이터베이스입니다 AWS. 다중 가용 영역 내구성과 함께 밀리초 미만의 쿼리 지연 시간을 제공합니다.

MemoryDB의 핵심 기능은 다음과 같습니다.

- 애플리케이션 데이터와 수백만 개의 벡터를 단일 데이터베이스에 저장
- 한 자릿수 밀리초 쿼리 및 업데이트 응답 시간
- 에서 가장 빠른 성능으로 가장 높은 재현율 AWS
- 벡터당 최대 32,768개 차원 지원
- 실시간 의미 체계 검색 및 캐싱 기능

다음과 같은 상황에서 MemoryDB를 사용합니다.

- 지연 시간이 매우 짧은(10ms 미만) 실시간 애플리케이션의 경우
- 하루에 수백만 개의 요청이 있는 처리량이 많은 워크로드의 경우
- 실시간 권장 엔진, 의미 체계 캐싱, 이상 탐지와 같은 사용 사례의 경우
- 인 메모리 데이터 스토어와 벡터 검색 기능이 모두 필요한 애플리케이션의 경우

자세한 내용은 MemoryDB 설명서의 [벡터 검색을](#) 참조하세요.

Amazon Neptune Analytics

Amazon Neptune Analytics는 네이티브 벡터 검색 기능을 제공하는 그래프 분석 엔진으로, 그래프 검색 증강 생성(GraphRAG) 사용 사례에 적합합니다. 벡터 유사성 검색을 그래프 순회 및 알고리즘과 결합합니다.

Neptune Analytics의 핵심 기능은 다음과 같습니다.

- 몇 초 안에 수십억 개의 관계 분석
- 벡터 검색과 그래프 알고리즘 결합(경로 조사 결과, 커뮤니티 탐지, 중심성)
- 토폴로지 지식을 갖춘 GraphRAG 애플리케이션 지원
- 기존 그래프 분석 솔루션보다 최대 80배 빠름
- 완전 관리형 GraphRAG를 위해 Amazon Bedrock과 통합

다음과 같은 상황에서 Neptune Analytics를 사용합니다.

- 벡터 임베딩이 포함된 지식 그래프가 필요한 GraphRAG 애플리케이션의 경우
- 벡터 유사성과 함께 복잡한 관계를 통과해야 하는 사용 사례의 경우
- 관계 컨텍스트를 사용하여 설명 가능한 AI 응답이 필요한 애플리케이션의 경우
- 고객 360 보기, 사기 탐지 네트워크, 지식 검색과 같은 시나리오의 경우

자세한 내용은 [Amazon Neptune Analytics 설명서를](#) 참조하세요.

Amazon S3 Vectors

Amazon S3 Vectors는 네이티브 벡터 스토리지 및 쿼리 기능을 AWS 갇춘의 첫 번째 클라우드 객체 스토어입니다. 대규모 규모가 필요한 AI 애플리케이션을 위해 특별히 구축되고 비용 최적화된 벡터 스토리지를 제공합니다.

Amazon S3 벡터의 핵심 기능은 다음과 같습니다.

- 벡터 버킷당 최대 10,000개의 인덱스를 지원하는 인덱스당 최대 20억 개의 벡터에 대한 스토리지
- 장기 스토리지 및 빈번하지 않은 액세스 패턴에 최적화된 Sub-100 쿼리 지연 시간
- 특수 벡터 데이터베이스에 비해 벡터 작업에 대해 최대 90% 비용 절감
- 자동 조정 및 99.99999999%(11 9s) 내구성을 갖춘 서버리스 아키텍처

다음과 같은 상황에서 Amazon S3 벡터를 사용합니다.

- 최소 비용으로 수십억 개의 벡터를 저장해야 하는 애플리케이션의 경우
- 10ms 미만이 아닌 1초 미만의 쿼리 지연 시간(100ms 이상)을 허용하는 워크로드의 경우
- 장기 벡터 보존 및 아카이브 사용 사례의 경우
- 검색 패턴이 빈번하지 않은 RAG 애플리케이션의 경우
- 지연 시간이 매우 짧은 것보다 스토리지 경제성을 우선시하는 조직의 경우

Amazon S3 Vectors는 기본적으로 Amazon Bedrock 지식 기반과 통합되며 Amazon OpenSearch Service의 계층형 아키텍처에서 잘 작동합니다. 콜드 스토리지에는 Amazon S3 벡터를 사용하고 핫 쿼리에는 OpenSearch Service를 사용할 수 있습니다.

자세한 내용은 Amazon [S3 설명서의 S3 벡터 및 벡터 버킷 작업을](#) 참조하세요. Amazon S3

관리형 서비스 옵션

Amazon Bedrock 지식 기반은 벡터 데이터베이스 구현에 대한 AWS 완전관리형 접근 방식을 나타냅니다. 이 서비스의 스토리지 옵션 유연성은 자동화된 관리 기능과 결합되어 복잡한 인프라를 관리하지 않고 RAG를 구현하려는 조직에 특히 유용합니다.

Amazon Bedrock 지식 기반을 사용하면 RAG를 사용하여 파운데이션 모델을 개선하는 지식 기반을 생성, 유지 관리 및 쿼리할 수 있습니다. 이 서비스는 전체 데이터 수집, 벡터화 및 검색 파이프라인을 관리하여 RAG 구현의 복잡한 프로세스를 간소화합니다.

Amazon Bedrock 지식 기반의 주요 이점은 다음과 같습니다.

- 간소화된 데이터 처리
 - 자동 데이터 수집 및 청킹
 - 여러 파일 형식에서 기본 제공 텍스트 추출
 - 관리형 벡터 임베딩 생성
 - 자동 메타데이터 추출 및 인덱싱
- 간소화된 RAG 구현
 - 미리 구성된 검색 전략
 - 자동 컨텍스트 창 최적화
 - 기본 제공 관련성 튜닝
 - 시맨틱 검색 기능 즉시 사용 가능
- 보안 및 거버넌스
 - 통합 AWS Identity and Access Management (IAM) 제어
 - 저장 및 전송 중 데이터 암호화
 - VPC 지원
 - 를 사용한 감사 로깅 AWS CloudTrail

Amazon Bedrock 지식 기반은 다음을 포함한 여러 [벡터 스토어 옵션](#)을 지원합니다.

- pgvector를 사용하는 Amazon Aurora PostgreSQL
- Amazon Neptune Analytics
- Amazon EMR Serverless
- Amazon S3 Vectors

- Pinecone
- Redis 엔터프라이즈 클라우드

이 관리형 서비스는 자동 데이터 수집, 벡터화 및 검색을 처리합니다. 이렇게 하면 RAG 구현이 간소화됩니다.

지원되는 각 벡터 저장소에 대한 자세한 내용은 [Amazon Bedrock 지식 기반 설명서를 참조하세요](#).

올바른 벡터 데이터베이스 선택

다음과 같은 주요 결정 요인을 기반으로 벡터 데이터베이스를 선택합니다.

- 벡터 검색과 함께 MongoDB 호환 문서 데이터베이스가 필요한 경우 - Amazon DocumentDB를 선택합니다. 이는 애플리케이션이 MongoDB APIs 사용하고 별도의 벡터 인프라를 관리하지 않고 의미 체계 검색 기능을 추가하려는 경우에 적합합니다.
- 실시간 애플리케이션에 매우 짧은 지연 시간이 필요한 경우 Amazon MemoryDB를 선택합니다. 이렇게 하면 응답 시간이 밀리초 AWS 미만인에서 가장 빠른 벡터 검색 성능이 제공됩니다. 실시간 추천 엔진 및 고처리량 애플리케이션에 적합합니다.
- 벡터 검색으로 그래프 기반 지식 표현이 필요한 경우 - Amazon Neptune Analytics를 선택합니다. 이는 복잡한 관계를 통과하고 벡터 검색과 함께 그래프 기반 쿼리를 수행해야 하는 GraphRAG 애플리케이션에 가장 적합하며 설명 가능한 AI 응답을 제공합니다.
- 관계형 쿼리를 벡터 검색과 결합해야 하는 경우 - pgvector를 사용하여 Amazon Aurora PostgreSQL을 선택합니다. 이 옵션은 애플리케이션에 동일한 데이터베이스 내에서 기존 SQL 작업과 벡터 유사성 검색이 모두 필요한 경우에 적합합니다.
- 지연 시간이 10ms 미만인 처리량이 많은 쿼리가 필요한 경우 Amazon OpenSearch Service를 선택합니다. 고빈도 쿼리 및 실시간 애플리케이션을 처리하는 데 뛰어나며 최근 GPU 가속화 개선 사항이 포함되어 있습니다.
- 수십억 개의 벡터를 비용 효율적으로 저장해야 하는 경우 - Amazon S3 벡터를 선택합니다. 이 옵션은 최대 90%의 비용 절감을 제공하며 100ms 미만의 지연 시간을 허용할 수 있는 검색 패턴(쿼리 간 몇 분에서 몇 시간)이 자주 발생하지 않는 애플리케이션에 적합합니다.
- 벡터 검색과 함께 전체 텍스트 검색이 필요한 경우 - Amazon OpenSearch Service를 선택합니다. 이 옵션은 강력한 전체 텍스트 검색 기능을 단일 플랫폼에서 벡터 검색과 결합합니다.

벡터 데이터베이스 비교

AWS 는 개별 벡터 데이터베이스부터 완전 관리형 서비스인 Amazon Bedrock 지식 기반에 이르기까지 벡터 검색 기능을 구현하기 위한 여러 접근 방식을 제공합니다. 이러한 옵션을 평가할 때 조직은 아키텍처, 확장성, 통합 기능, 성능 특성 및 보안 기능을 비롯한 다양한 측면을 고려해야 합니다.

개별 벡터 데이터베이스

다음 표에서는 아키텍처, 조정 기능, 데이터 소스 통합 및 성능 특성에 중점을 둔 여러 AWS 개별 벡터 데이터베이스 솔루션의 주요 기능에 대한 개요를 제공합니다.

Feature	Amazon Kendra	Amazon OpenSearch Service	pgvector를 PostgreSQLwith	Amazon DocumentDB	Amazon MemoryDB	Amazon Neptune Analytics	Amazon S3 Vectors
기본 사용 사례	엔터프라이즈 검색 및 RAG	분산 검색 및 분석	벡터가 지원되는 관계형 DB	벡터 검색이 포함된 문서 DB	실시간 인 메모리 벡터 검색	벡터 검색을 사용한 그래프 분석	비용 최적화 벡터 스토리지
아키텍처	완전관리형	분산 클러스터	관계형 데이터베이스	문서 지향	인 메모리 데이터베이스	그래프 분석 엔진	서버리스 객체 스토리지
데이터 모델	문서 기반	JSON 문서	관계형 테이블	JSON 문서	JSON을 사용한 키-값	속성 그래프	객체 스토리지
벡터 차원	자동으로 관리됨	최대 16,000개	구성 가능	최대 2,000(인덱싱됨), 16,000(인덱싱되지 않음)	최대 32,768	구성 가능	최대 4,096

인덱싱 방법	자동	HNSW, IVF	HNSW, IVFFlat	HNSW, IVFFlat	HNSW	네이티브 그래프 및 벡터	자동
거리 지표	자동	코사인, 유클리드, 점 제품	코사인, 유클리드, 내부 제품	코사인, 유클리드, 점 제품	코사인, 유클리드, 내부 제품	코사인, 유클리드	코사인, 유클리드
쿼리 지연 시간	1초 미만	Sub-10(GPU 가속)	10~100ms	밀리초	1밀리초 미만	1초 미만	Sub-100
크기 조정 모델	자동	가로(노드 추가)	수직 및 읽기 전용 복제본	가로(인스턴스 추가)	수직 및 복제본	자동	자동(서버리스)
최대 벡터	관리형	수십억(클러스터 종속)	수백만(인스턴스에 따라 다름)	컬렉션당 수백만 개	데이터베이스당 수백만	수십억	인덱스당 20억 개, 버킷당 인덱스 10,000개
처리량	높음	매우 높음(수천 QPS)	중간	높음	매우 높음(하루에 수백만 개의 요청)	높음	중간(빈번하지 않은 쿼리에 최적화됨)
데이터 내구성	99.999999	복제본으로 구성 가능(11 9초)	99.99%(다중 AZ)	99.99%(다중 AZ)	99.99%(다중 AZ)	99.99%	99.999999 999%(11 9초)
일관성 모델	최종	최종(구성 가능)	강력한 (ACID)	최종	강함	강함	강함
추가 기능	40개 이상의 데이터 커넥터, NLP	전체 텍스트 검색, 분석, 대시보드	SQL 쿼리, ACID 트랜잭션	MongoDB API 호환성	Redis API 호환성, 캐싱	그래프 알고리즘, 순회	Amazon S3 통합, 수명 주기 정책

요금 모델	쿼리 및 스토리지 당 지불	인스턴스 시간 및 스토리지	인스턴스 시간 및 스토리지	인스턴스 시간 및 스토리지	인스턴스 시간 및 스토리지	용량 단위 및 스토리지	스토리지, 쿼리 및 데이터 전송
비용 최적화	사용량 기반	예약 인스턴스, Auto Scaling	예약 인스턴스, Aurora Serverless	예약 인스턴스	예약 인스턴스	Auto Scaling	특수 DBs 대비 최대 90% 절감
최적의 용도	최소한의 설정으로 엔터프라이즈 검색	처리량이 많고 지연 시간이 짧은 쿼리	하이브리드 SQL 및 벡터 워크로드	벡터가 필요한 MongoDB 호환 앱	지연 시간이 매우 짧은 실시간 앱	GraphRAG 및 지식 그래프	장기적이고 비용 효율적인 스토리지
이상적인 쿼리 패턴	빈번한 엔터프라이즈 검색	고주파 실시간 쿼리	혼합 SQL 및 벡터 쿼리	의미 체계 검색을 사용하여 쿼리 문서화	하루에 수백만 개의 요청	벡터 검색을 사용한 그래프 순회	자주 수행하지 않는 쿼리(분~시간)
설정 복잡성	낮음(완전 관리형)	중간(클러스터 구성)	중간(확장 설정)	중간(클러스터 구성)	중간(클러스터 구성)	낮음(완전 관리형)	낮음(서버리스)
팀 전문 지식 필요	최소화	OpenSearch 또는 Elasticsearch	PostgreSQL, SQL	MongoDB	Redis	그래프 데이터베이스	Amazon S3, 기본 벡터 개념

관리형 서비스 - Amazon Bedrock 지식 기반

Amazon Bedrock 지식 기반은 다양한 벡터 스토리지 옵션을 갖춘 완전관리형 솔루션을 제공합니다. 다음 표에서는 이러한 스토리지 옵션을 비교합니다.

Feature	pgvector를 PostgreSQLwith	Neptune 분 석	OpenSearch Service Serverless	Amazon S3 벡터	Pinecone	RedisEnterprise 클라우드
기본 사용 사례	벡터 RAG를 사용한 관계형 DB	GraphRAG에 대한 그래프 기반 벡터 검색	지식 관리 RAG	비용 최적화 벡터 RAG	고성능 벡터 검색	인 메모리 벡터 검색
아키텍처	완전 관리형 관계형	완전 관리형 그래프 분석	완전 관리형 서버리스	서버리스 객체 스토리지	완전 관리형 하이브리드 클라우드	완전 관리형 인 메모리
데이터 모델	관계형 테이블	속성 그래프	JSON 문서	객체 스토리지	목적별 벡터	벡터가 있는 키-값
벡터 스토리지	pgvector 확장을 통해	네이티브 그래프 벡터	OpenSearch 엔진을 통해	네이티브 Amazon S3 벡터 스토리지	네이티브 벡터 데이터베이스	인 메모리 벡터
Amazon Bedrock 통합	기본	기본	기본	기본	기본	기본
자동 수집	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)
자동 벡터화	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)	예(Amazon Bedrock을 통해)
규모 조정	Auto Scaling(Aurora Serverless)	자동 그래프 조정	자동 서버리스	자동(벡터 수집액 개)	Auto Scaling 포드	클러스터 자동 크기 조정

쿼리 성능	관계형 또는 벡터의 경우 높음	그래프 벡터의 경우 높음	높음	중간 (100ms 이상 지연 시간)	매우 높음	매우 높음
최대 벡터	수백만(인스턴스에 따라 다름)	수십억	수십억	인덱스당 20억	수십억	수백만(메모리에 따라 다름)
추가 기능	SQL 쿼리, ACID 트랜잭션	그래프 알고리즘, 순회	전체 텍스트 검색, 분석	Amazon S3 수명 주기, 계층화	메타데이터 필터링, 네임스페이스	Redis 데이터 구조, 캐싱
비용 최적화	보통 (Aurora Serverless)	보통(용량 단위)	높음(서버리스, pay-per-use)	매우 높음 (최대 90% 절감)	보통(포드 기반 요금)	낮음(인 메모리 프리미엄)
최적의 용도	하이브리드 SQL/벡터 워크로드	연결된 지식 그래프	벡터 검색이 포함된 전체 텍스트	장기적이고 액세스 빈도가 낮은 벡터	대규모 실시간 벡터 검색	지연 시간이 매우 짧은 요구 사항
이상적인 쿼리 패턴	혼합 SQL 및 벡터 쿼리	벡터를 사용한 그래프 순회	분석을 통한 빈번한 검색	자주 검색하지 않음 (분~시간)	고주파 실시간 쿼리	초당 수백만 개의 요청
Amazon Bedrock을 사용한 설정	단순 (Amazon Bedrock에서 관리)	단순 (Amazon Bedrock에서 관리)	단순 (Amazon Bedrock에서 관리)	단순 (Amazon Bedrock에서 관리)	단순 (Amazon Bedrock에서 관리)	단순 (Amazon Bedrock에서 관리)
데이터 레지던시	AWS 리전	AWS 리전	AWS 리전	AWS 리전	멀티 클라우드(AWS 및 기타)	멀티 클라우드(AWS 및 기타)
요금 모델	인스턴스 시간 및 스토리지	용량 단위 및 스토리지	컴퓨팅 및 스토리지 (서버리스)	스토리지, 쿼리 및 전송	포드 시간 및 스토리지	노드 시간 및 스토리지

개별 옵션과 관리형 옵션 중에서 선택

고려 사항	개별 벡터 DB 선택	Amazon Bedrock 지식 기반(관리형) 선택
RAG 구현	RAG 파이프라인을 완전히 제어하려는 경우	최소한의 설정으로 완전 관리형 RAG를 원하는 경우
사용자 지정	사용자 지정 검색 로직 및 사전 처리가 필요합니다.	요구 사항을 충족하는 표준 RAG 패턴
기존 인프라	데이터베이스가 이미 배포되어 있습니다.	새로 시작하거나 간소화된 관리를 원하는 경우
팀 전문 지식	팀에 데이터베이스 관리 전문 지식이 있음	인프라가 아닌 애플리케이션 로직에 집중하는 것을 선호합니다.
통합 복잡성	기존 시스템과의 심층 통합이 필요합니다.	Amazon Bedrock 모델과 빠르게 통합하려는 경우
운영 오버헤드	데이터베이스 작업을 관리할 수 있습니다.	작업을 처리 AWS 하려는 경우
비용 구조	직접 데이터베이스 요금을 선호합니다.	통합 Amazon Bedrock 요금을 선호합니다.
출시 시간	사용자 지정 구현 시간이 있습니다.	빠른 배포가 필요합니다.

비용 비교 및 고려 사항

다양한 벡터 데이터베이스 옵션의 비용 구조를 이해하는 것은 정보에 입각한 구현 결정을 내리는 데 필수적입니다. 다음 표에는 개별 데이터베이스와 관리형 서비스를 포함하여 다양한 벡터 데이터베이스 솔루션에 대한 몇 가지 주요 비용 고려 사항이 요약되어 있습니다. 각 옵션에는 pay-as-you-go제 모델부터 인프라 및 운영 비용에 이르기까지 총 소유 비용(TCO)에 영향을 미칠 수 있는 고유한 요금 요소가 있습니다.

벡터 데이터베이스	비용 모델	비용 고려 사항
Amazon Kendra	쿼리를 기반으로 사용한 만큼 결제	비용은 쿼리 수와 인덱싱된 데이터 양에 따라 달라질 수 있습니다. 데이터 저장 및 데이터 전송에는 추가 요금이 적용될 수 있습니다. 자세한 내용은 Amazon Kendra 요금 을 참조하세요.
Amazon OpenSearch Service	인스턴스 시간 및 스토리지를 기준으로 사용한 만큼 결제	비용에는 인스턴스 시간, 스토리지(Amazon EBS 볼륨), 데이터 전송 및 선택적 UltraWarm 스토리지가 포함됩니다. 예약 인스턴스는 최대 30%의 비용을 절감할 수 있습니다. GPU 가속 인스턴스는 벡터 워크로드에 더 나은 가격 대비 성능을 제공합니다. 자세한 내용은 Amazon OpenSearch Service 요금 을 참조하세요.
오픈 소스 OpenSearch	오픈 소스, 직접 비용 없음(소프트웨어를 다운로드하거나 사용하기 위해 비용을 지불할 필요가 없으며 라이선스 비용 없음)	비용에는 인프라(예: 서버 및 스토리지) 및 운영 비용(예: 유지 관리 및 모니터링)이 포함됩니다. 조직은 인력이 인프라를 관리하고 유지 관리할 수 있도록 예산을 책정해야 합니다.

pgvector를 사용하는 Amazon RDS for PostgreSQL	사용량에 따라 사용한 만큼 결제	비용에는 데이터베이스 인스턴스 유형, 스토리지, 데이터 전송 및 백업이 포함됩니다. AWS 프리 티어 이외의 데이터 전송, 인스턴스 유형 및 스토리지에는 추가 요금이 적용될 수 있습니다. 자세한 내용은 Amazon RDS 요금 을 참조하십시오.
Amazon DocumentDB	인스턴스 시간 및 스토리지를 기준으로 사용한 만큼 결제	비용에는 인스턴스 시간, 스토리지(GB-월), I/O 요청, 백업 스토리지 및 데이터 전송이 포함됩니다. 탄력적 클러스터는 동적 조정을 활성화합니다. 비용 최적화에 사용할 수 있는 예약 인스턴스입니다. 자세한 내용은 Amazon DocumentDB 요금 을 참조하세요.
Amazon MemoryDB	노드 시간 및 데이터 스토리지를 기준으로 사용한 만큼 결제	비용에는 노드 시간(노드 유형별), 데이터 스토리지(GB-시간), 스냅샷 스토리지 및 데이터 전송이 포함됩니다. 예약 노드는 최대 55%의 비용을 절감할 수 있습니다. 처리량이 많고 지연 시간이 짧은 워크로드에 최적화되었습니다. 자세한 내용은 Amazon MemoryDB 요금 을 참조하세요.

Amazon Neptune Analytics	용량 단위를 기준으로 사용한 만큼 결제	비용에는 Neptune 용량 단위 (NCUs), 스토리지(GB-월) 및 데이터 전송이 포함됩니다. 선 결제 약정 없이 워크로드를 기반으로 자동 조정. 최소 128개의 NCUs 필요합니다. 자세한 내용은 Amazon Neptune 요금 을 참조하세요.
Amazon S3 Vectors	스토리지 및 요청에 따라 즉시 결제	비용에는 스토리지(GB-월), PUT 및 GET 요청, 벡터 인덱스 관리, 데이터 전송이 포함됩니다. 특수 벡터 데이터베이스에 비해 최대 90%의 비용 절감 효과를 제공합니다. 추가 최적화를 위해 Amazon S3 Intelligent-Tiering 및 수명 주기 정책을 사용할 수 있습니다. 자세한 내용은 Amazon S3 요금 을 참조하세요.
Amazon Bedrock 지식 기반	사용량에 따라 사용한 만큼 결제	비용은 지식 기반 및 Amazon OpenSearch Serverless와 같은 추가 서비스의 사용량에 따라 달라질 수 있습니다. 데이터 스토리지, 데이터 전송 및 추가 기능에 추가 요금이 적용될 수 있습니다. 요금에 대한 자세한 내용은 Amazon OpenSearch Service 요금 을 참조하세요.

벡터 데이터베이스 사용 사례

다음 예제에서는 다양한 벡터 데이터베이스 옵션을 효과적으로 사용하여 지식 관리를 개선하고, 운영 효율성을 개선하고, 더 나은 비즈니스 성과를 제공하는 방법을 강조합니다. 이러한 사용 사례는 이 가이드의 앞부분에서 설명한 벡터 데이터베이스 솔루션의 실제 애플리케이션을 설명하고 실제 성능 및 이점에 대한 인사이트를 제공합니다.

Amazon Kendra를 사용한 지식 관리

고객 문제 - 일본에서 가장 큰 일반 계약업체 중 하나가 숙련된 직원이 감소하고 있었습니다. 회사는 경험 담당자의 지식과 기술을 청년 세대에 효율적으로 이전할 수 있는 방법이 필요했습니다. 복잡한 건설 엔지니어링 지식과 과거 경험을 캡처하고 전파하는 솔루션이 필요했습니다.

AWS 솔루션 -이 문제를 해결하기 위해 고객은 내부 지식 기반을 빠르고 정확하게 처리하고 자연어 쿼리를 허용할 수 있는 AI 솔루션인 Amazon Kendra를 선택했습니다. Amazon Kendra를 사용하면 이제 직원은 필요한 정보를 훨씬 더 빠르게 찾아 생산성을 개선하고 숙련된 직원에서 청년 직원으로의 지식 이전을 촉진할 수 있습니다.

영향 - Amazon Kendra로 구동되는 생성형 AI 챗봇을 구현하여 통합 지식 플랫폼을 만들었습니다. 챗봇을 통해 직원은 기술 지식과 건설 엔지니어링에 대한 과거 경험에 빠르게 액세스할 수 있습니다. 이 솔루션은 조직 내 지식 이전 및 의사 결정 프로세스의 효율성을 크게 개선하여 모든 직원이 귀중한 전문 지식을 보존하고 쉽게 액세스할 수 있도록 했습니다. 이 솔루션의 비용은 사용량 및 구성에 따라 달라질 수 있습니다. 자세한 비용 견적은 단원을 참조하십시오 [AWS Pricing Calculator](#). 벡터 데이터베이스 비용 추정은 이 가이드의 [비용 비교 및 고려 사항](#) 섹션을 참조하거나 [Amazon Kendra 요금](#)을 참조하세요.

다른 고객 사용 사례에 대한 자세한 내용은 [Amazon Kendra 고객](#)을 참조하세요.

OpenSearch Serverless를 사용한 실시간 분석

고객 문제 - 선도적인 금융 서비스 공급자는 엄청난 데이터 에코시스템을 관리하는 문제에 직면했습니다. 연간 3억 개의 권한 부여와 9천만 개의 트랜잭션을 처리하여 약 1.1페타바이트(PB)의 데이터에 누적되었습니다. 6,000개 이상의 보고서에 액세스해야 하는 300,000명의 사용자에게 서비스를 제공하는 기존 시스템은 글로벌 일관성을 제공하고 실시간 의사 결정을 가능하게 하기 위해 현대화가 필요했습니다.

AWS 솔루션 - 솔루션 아키텍처는 자연어 처리를 위해 Amazon Bedrock(Anthropic, Sonnet 3, Sonnet 3.5 및 Haiku 포함)을 통해 사용할 수 있는 파운데이션 모델을 사용했습니다. 고객은 뛰어난 확장성

과 대량 데이터 볼륨을 효율적으로 처리할 수 있는 기능을 위해 벡터 데이터베이스로 OpenSearch Serverless를 선택했습니다. 이 아키텍처를 통해 복잡한 쿼리를 원활하게 처리하고 동적 보고서를 생성할 수 있었습니다.

영향 -이 구현은 100개 이상의 비즈니스 인텔리전스 대시보드를 수동으로 생성할 필요가 없어 생산성이 50% 향상되었습니다. 이제 사용자는 응답 시간이 20~40초인 자연어 쿼리를 통해 보고서를 생성할 수 있습니다. 이 솔루션의 비용은 사용량 및 구성에 따라 달라질 수 있습니다. 자세한 비용 견적은 단원을 참조하십시오 [AWS Pricing Calculator](#). 벡터 데이터베이스 비용 추정치는 이 가이드의 [비용 비교 및 고려 사항](#) 섹션을 참조하거나 [Amazon OpenSearch Service 요금](#)을 참조하세요. 다른 고객 사용 사례에 대한 자세한 내용은 [Amazon OpenSearch Serverless](#)를 참조하세요.

다음 단계 및 리소스

이 가이드를 검토한 후 다음 작업을 고려하여 이해에서 구현으로 이동합니다.

1. 현재 요구 사항을 평가합니다.
 - 기존 데이터베이스 인프라 및 전문 지식을 평가합니다.
 - 특정 벡터 검색 요구 사항을 문서화합니다.
 - 성능, 규모 조정 및 비용 목표를 정의합니다.
2. 다음 옵션 중 하나를 선택하여 벡터 데이터베이스 옵션을 테스트합니다.
 - 옵션 1: 선호하는 벡터 데이터베이스 솔루션을 사용하여 개념 증명을 설정합니다.
 - 옵션 2: Amazon Bedrock 지식 기반에서 샘플 데이터 세트를 실험합니다. Amazon Bedrock 지식 기반에 대한 빠른 생성 환경을 사용해 보세요. 예제는 [Aurora 설명서의 Amazon Bedrock용 Aurora PostgreSQL 지식 기반 빠른 생성을](#) 참조하세요.
3. 추가 [리소스](#)를 검토합니다.
4. 전문가의 도움을 받으세요.
 - 구현 지침은 AWS 계정 팀 또는 AWS 솔루션 아키텍트에 문의하십시오.
 - 벡터 데이터베이스를 전문으로 하는 [AWS 파트너와 협력](#)합니다.
5. 프로덕션 배포를 계획합니다.
 - 기존 데이터베이스에서 이동하는 경우 마이그레이션 전략을 생성합니다.
 - 선택한 솔루션에 대한 조정 계획을 수립합니다.
 - 모니터링 및 유지 관리 절차를 설계합니다.

리소스

다음 리소스는 벡터 데이터베이스를 선택하는 데 도움이 될 수 있습니다.

AWS 블로그 게시물

- [Amazon Bedrock 지식 기반 빠른 생성 및 Amazon Aurora Serverless를 사용하여 생성형 AI 애플리케이션 개발 가속화](#)
- [Amazon OpenSearch Service의 벡터 데이터베이스 기능 설명](#)
- [Amazon Bedrock 지식 기반을 사용하여 벡터 데이터 스토어 심층 분석](#)
- [자연어 처리, 챗봇 및 감정 분석을 위해 pgvector 및 Amazon Aurora PostgreSQL 활용](#)

AWS 서비스 설명서

- [AWS 데이터베이스 서비스 선택](#)
- [Amazon Bedrock 지식 기반 작동 방식](#)
- [Neptune Analytics 설명서](#)
- [Amazon Web Services 개요: 데이터베이스](#)
- [Aurora PostgreSQL을 Amazon Bedrock의 지식 기반으로 사용](#)
- [Amazon Aurora PostgreSQL 작업](#)
- [Amazon DocumentDB](#)
- [Amazon MemoryDB](#)
- [Amazon S3 Vectors](#)

기타 AWS 리소스

- [Amazon Bedrock 지식 기반](#)
- [벡터 데이터베이스 및 임베딩](#)
- [생성형 AI 애플리케이션을 위한 벡터 데이터베이스](#)
- [Machine Learning의 임베딩이란 무엇입니까?](#)

기타 리소스

- [PostgreSQL 정보](#)
- [pgvector 설명서](#)
- [Amazon Bedrock의 지식 기반으로 Pinecone](#)
- [의 Redis Enterprise Cloud AWS](#)

문서 기록

다음 표에서는이 가이드 RAG 사용 사례에 대한 AWS 벡터 데이터베이스 선택에 대한 중요한 변경 사항을 설명합니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
추가됨 AWS 서비스	Amazon DocumentDB, Amazon MemoryDB, Amazon S3 Vectors 및 Amazon Neptune Analytics 사용에 대한 정보가 추가되었습니다.	2026년 3월 30일
최초 게시	—	2025년 3월 6일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.