Pengambilan opsi dan arsitektur Augmented Generation di AWS

AWS Panduan Preskriptif

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

AWS Panduan Preskriptif: Pengambilan opsi dan arsitektur Augmented Generation di AWS

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Pengantar 1
Audiens yang dituju 1
Tujuan 1
Opsi AI generatif 3
Memahami RAG 4
Komponen-komponen 6
Membandingkan RAG dan fine-tuning 7
Gunakan kasus untuk RAG 10
Opsi RAG yang dikelola sepenuhnya 11
Basis pengetahuan untuk Amazon Bedrock 11
Sumber data 13
Database vektor 15
Amazon Q Bisnis 16
Fitur utama 16
End-user kustomisasi 18
Kanvas Amazon SageMaker AI 18
Arsitektur RAG kustom 21
Penarik Data 21
Amazon Kendra 22
OpenSearch Layanan Amazon 23
Amazon Aurora PostgreSQL dan pgvector 24
Analisis Amazon Neptunus 25
Amazon MemoryDB 25
Amazon DocumentDB 27
Biji pinus 29
MongoDB Atlas 30
Weaviate 31
Generator 32
Amazon Bedrock 32
SageMaker AI JumpStart 33
Memilih opsi RAG 34
Kesimpulan 36
Riwayat dokumen 37
..... xxxviii

Pengambilan opsi dan arsitektur Augmented Generation di AWS

Mithil Shah, Rajeev Muralidhar, dan Benteng Natacha, Amazon Web Services

Oktober 2024 (sejarah dokumen)

Generative AI mengacu pada subset model AI yang dapat membuat konten dan artefak baru, seperti gambar, video, teks, dan audio, dari prompt teks sederhana. Model AI generatif dilatih pada sejumlah besar data yang mencakup berbagai subjek dan tugas. Hal ini memungkinkan mereka untuk menunjukkan keserbagunaan yang luar biasa dalam melakukan berbagai tugas, bahkan yang belum dilatih secara eksplisit. Karena kemampuan model tunggal untuk melakukan banyak tugas, model ini sering disebut sebagai model dasar (FMs).

Salah satu aplikasi penting dari model AI generatif adalah kemahiran mereka dalam menjawab pertanyaan. Namun, ada tantangan khusus yang muncul ketika model ini digunakan untuk menjawab pertanyaan berdasarkan dokumen khusus. Dokumen khusus dapat mencakup informasi kepemilikan, situs web internal, dokumentasi internal, Confluence halaman, SharePoint halaman, dan lainnya. Salah satu opsi adalah menggunakan Retrieval Augmented Generation (RAG). Dengan RAG, model foundation mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya (seperti dokumen kustom Anda) sebelum menghasilkan respons.

Panduan ini menjelaskan opsi AI generatif berbeda yang tersedia untuk menjawab pertanyaan dari dokumentasi khusus, termasuk sistem Retrieval Augmented Generation (RAG). Ini juga memberikan gambaran umum tentang membangun sistem RAG di Amazon Web Services (AWS). Dengan meninjau opsi dan arsitektur RAG, Anda dapat memilih antara layanan yang dikelola sepenuhnya AWS dan arsitektur RAG khusus.

Audiens yang dituju

Audiens yang dituju untuk panduan ini adalah arsitek dan manajer AI generatif yang ingin membangun solusi RAG, untuk meninjau arsitektur yang tersedia, dan untuk memahami manfaat dan kerugian dari setiap opsi.

Tujuan

Panduan ini membantu Anda melakukan hal berikut:

- Memahami opsi AI generatif yang tersedia untuk menjawab pertanyaan dari dokumen khusus
- Tinjau opsi arsitektur untuk sistem RAG AWS
- Pahami kelebihan dan kekurangan masing-masing opsi RAG
- Pilih arsitektur RAG untuk lingkungan Anda AWS

Opsi AI generatif untuk menanyakan dokumen khusus

Organizations sering memiliki berbagai sumber data terstruktur dan tidak terstruktur. Panduan ini berfokus pada bagaimana Anda dapat menggunakan AI generatif untuk menjawab pertanyaan dari data yang tidak terstruktur.

Data yang tidak terstruktur dalam organisasi Anda dapat berasal dari berbagai sumber. Ini mungkin PDF, file teks, wiki internal, dokumen teknis, situs web publik, basis pengetahuan, atau lainnya. Jika Anda menginginkan model dasar yang dapat menjawab pertanyaan tentang data tidak terstruktur, opsi berikut tersedia:

- Latih model yayasan baru dengan menggunakan dokumen khusus Anda dan data pelatihan lainnya
- Fine-tune model dasar yang ada dengan menggunakan data dari dokumen kustom Anda
- Gunakan pembelajaran dalam konteks untuk meneruskan dokumen ke model yayasan saat Anda mengajukan pertanyaan
- Gunakan pendekatan Retrieval Augmented Generation (RAG)

Melatih model fondasi baru dari awal yang menyertakan data kustom Anda adalah usaha yang ambisius. Beberapa perusahaan telah berhasil melakukannya, seperti Bloomberg dengan BloombergGPTmodel mereka. Contoh lain adalah EXAONEmodel multimodal olehLG AI Research, yang dilatih dengan menggunakan 600 miliar karya seni dan 250 juta gambar resolusi tinggi, disertai dengan teks. Menurut The Cost of AI: Sould You Build or Buy Your Foundation Model (LinkedIn), model yang mirip dengan Meta Llama 2 biaya sekitar USD $4,8 juta untuk dilatih. Ada dua prasyarat utama untuk melatih model dari awal: akses ke sumber daya (keuangan, teknis, waktu) dan pengembalian investasi yang jelas. Jika ini tampaknya tidak cocok, maka opsi selanjutnya adalah menyempurnakan model pondasi yang ada.

Fine-tuning model yang ada melibatkan pengambilan model, seperti Amazon Titan, Mistral, atau model Llama, dan kemudian mengadaptasi model ke data kustom Anda. Ada berbagai teknik untuk fine-tuning, yang sebagian besar melibatkan memodifikasi hanya beberapa parameter alih-alih memodifikasi semua parameter dalam model. Ini disebut fine-tuning parameter-efisien. Ada dua metode utama untuk fine-tuning:

- Penyetelan halus yang diawasi menggunakan data berlabel dan membantu Anda melatih model untuk jenis tugas baru. Misalnya, jika Anda ingin membuat laporan berdasarkan formulir PDF,

maka Anda mungkin harus mengajarkan model bagaimana melakukannya dengan memberikan contoh yang cukup.

- Penyetelan halus tanpa pengawasan adalah agnostik tugas dan menyesuaikan model fondasi dengan data Anda sendiri. Ini melatih model untuk memahami konteks dokumen Anda. Model fine-tuned kemudian membuat konten, seperti laporan, dengan menggunakan gaya yang lebih kustom organisasi Anda.

Namun, fine-tuning mungkin tidak ideal untuk kasus penggunaan tanya jawab. Untuk informasi selengkapnya, lihat Membandingkan RAG dan fine-tuning dalam panduan ini.

Ketika Anda mengajukan pertanyaan, Anda dapat meneruskan dokumen model dasar dan menggunakan pembelajaran dalam konteks model untuk mengembalikan jawaban dari dokumen. Opsi ini cocok untuk kueri ad-hoc dari satu dokumen. Namun, solusi ini tidak berfungsi dengan baik untuk menanyakan beberapa dokumen atau untuk sistem kueri dan aplikasi, seperti Microsoft SharePoint atau Atlassian Confluence.

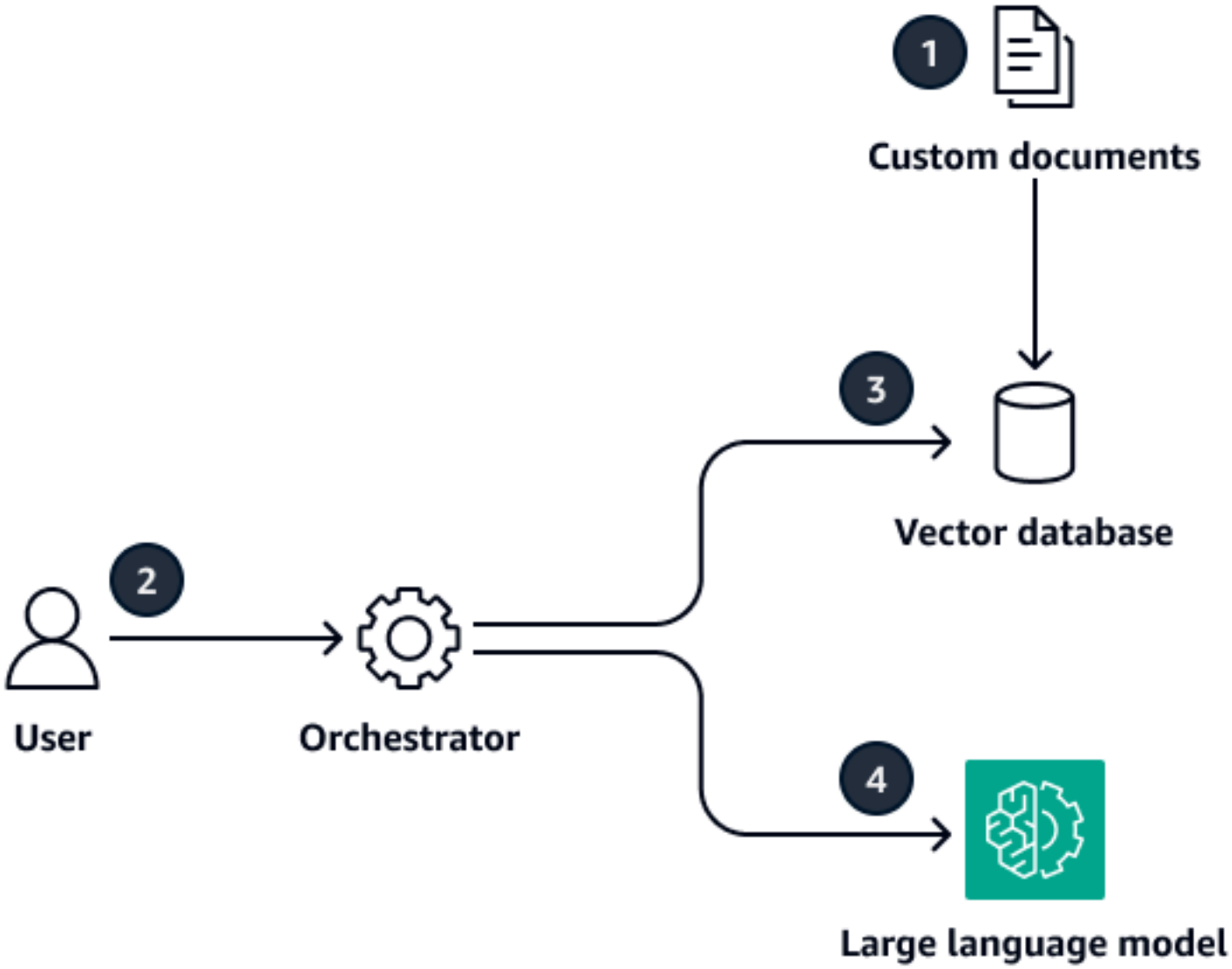
Opsi terakhir adalah menggunakan RAG. Dengan RAG, model foundation mereferensikan dokumen kustom Anda sebelum menghasilkan respons. RAG memperluas kemampuan model ke basis pengetahuan internal organisasi Anda, semua tanpa perlu melatih kembali model. Ini adalah pendekatan hemat biaya untuk meningkatkan output model sehingga tetap relevan, akurat, dan berguna dalam berbagai konteks.

Topik di bagian ini:

- Memahami Generasi Augmented Retrieval
- Membandingkan Retrieval Augmented Generation dan fine-tuning
- Kasus penggunaan untuk Retrieval Augmented Generation

Memahami Generasi Augmented Retrieval

Retrieval Augmented Generation (RAG) adalah teknik yang digunakan untuk menambah model bahasa besar (LLM) dengan data eksternal, seperti dokumen internal perusahaan. Ini memberikan model dengan konteks yang dibutuhkan untuk menghasilkan output yang akurat dan berguna untuk kasus penggunaan spesifik Anda. RAG adalah pendekatan pragmatis dan efektif untuk menggunakan LLM dalam suatu perusahaan. Diagram berikut menunjukkan gambaran tingkat tinggi tentang cara kerja pendekatan RAG.



Secara garis besar, proses RAG adalah empat langkah. Langkah pertama dilakukan sekali, dan tiga langkah lainnya dilakukan sebanyak yang diperlukan:

1. Anda membuat embeddings untuk menelan dokumen internal ke dalam database vektor. Embeddings adalah representasi numerik teks dalam dokumen yang menangkap makna semantik atau kontekstual dari data. Database vektor pada dasarnya adalah database dari embeddings ini, dan kadang-kadang disebut penyimpanan vektor atau indeks vektor. Langkah ini membutuhkan pembersihan data, pemformatan, dan chunking, tetapi ini adalah aktivitas satu kali di muka.
2. Seorang manusia mengirimkan kueri dalam bahasa alami.
3. Orkestrator melakukan pencarian kesamaan dalam database vektor dan mengambil data yang relevan. Orkestrator menambahkan data yang diambil (juga dikenal sebagai konteks) ke prompt yang berisi kueri.

4. Orkestrator mengirimkan kueri dan konteks ke LLM. LLM menghasilkan respons terhadap kueri dengan menggunakan konteks tambahan.

Dari perspektif pengguna, RAG terlihat seperti berinteraksi dengan LLM apa pun. Namun, sistem tahu lebih banyak tentang konten yang dimaksud dan memberikan jawaban yang disesuaikan dengan basis pengetahuan organisasi.

Untuk informasi selengkapnya tentang cara kerja pendekatan RAG, lihat Apa itu RAG di situs web. AWS

Komponen sistem RAG tingkat produksi

Membangun sistem RAG tingkat produksi membutuhkan pemikiran melalui beberapa aspek yang berbeda dari alur kerja RAG. Secara konseptual, alur kerja RAG tingkat produksi memerlukan kemampuan dan komponen berikut, terlepas dari implementasi spesifiknya:

- Konektor — Ini menghubungkan sumber data perusahaan yang berbeda dengan database vektor. Contoh sumber data terstruktur termasuk database transaksional dan analitis. Contoh sumber data yang tidak terstruktur termasuk penyimpanan objek, basis kode, dan platform perangkat lunak sebagai layanan (SaaS). Setiap sumber data mungkin memerlukan pola konektivitas, lisensi, dan konfigurasi yang berbeda.
- Pemrosesan data — Data datang dalam berbagai bentuk dan bentuk, seperti PDF, gambar yang dipindai, dokumen, presentasi, dan file. Microsoft SharePoint Anda harus menggunakan teknik pemrosesan data untuk mengekstrak, memproses, dan menyiapkan data untuk pengindeksan.
- Embeddings — Untuk melakukan pencarian relevansi, Anda harus mengonversi dokumen dan kueri pengguna Anda ke dalam format yang kompatibel. Dengan menggunakan menyematkan model bahasa, Anda mengonversi dokumen menjadi representasi numerik. Ini pada dasarnya adalah input untuk model pondasi yang mendasarinya.
- Database vektor — Database vektor adalah indeks dari embeddings, teks terkait, dan metadata. Indeks dioptimalkan untuk pencarian dan pengambilan.
- Retriever — Untuk kueri pengguna, retriever mengambil konteks yang relevan dari database vektor dan memberi peringkat tanggapan berdasarkan persyaratan bisnis.
- Model pondasi — Model dasar untuk sistem RAG biasanya LLM. Dengan memproses konteks dan prompt, model pondasi menghasilkan dan memformat respons bagi pengguna.
- Pagar pembatas - Pagar pembatas dirancang untuk memastikan bahwa kueri, konteks yang cepat, diambil, dan respons LLM akurat, bertanggung jawab, etis, dan bebas dari halusinasi dan bias.

- Orkestrator - Orkestrator bertanggung jawab untuk menjadwalkan dan mengelola alur kerja ujung ke ujung.
- Pengalaman pengguna — Biasanya, pengguna berinteraksi dengan antarmuka obrolan percakapan yang memiliki fitur yang kaya, termasuk menampilkan riwayat obrolan dan mengumpulkan umpan balik pengguna tentang tanggapan.
- Identitas dan manajemen pengguna - Sangat penting untuk mengontrol akses pengguna ke aplikasi dengan perincian yang baik. Dalam AWS Cloud, kebijakan, peran, dan izin biasanya dikelola melalui AWS Identity and Access Management (IAM).

Jelas, ada sejumlah besar pekerjaan untuk merencanakan, mengembangkan, merilis, dan mengelola sistem RAG. Layanan yang dikelola sepenuhnya, seperti Amazon Bedrock atau Amazon Q Business, dapat membantu Anda mengelola beberapa angkat berat yang tidak berdiferensiasi. Namun, arsitektur RAG kustom dapat memberikan kontrol lebih besar atas komponen, seperti retriever atau database vektor.

Membandingkan Retrieval Augmented Generation dan fine-tuning

Tabel berikut menjelaskan keuntungan dan kerugian dari fine-tuning dan pendekatan. RAG-based

Pendekatan	Keuntungan	Kekurangan
Fine-tuning	• Jika model yang disetel dengan baik dilatih menggunakan pendekatan tanpa pengawasan, maka ia dapat membuat konten yang lebih cocok dengan gaya organisasi Anda. • Model yang disetel dengan baik yang dilatih tentang data kepemilikan atau peraturan dapat membantu organisasi Anda mengikuti data internal atau khusus industri serta standar kepatuhan.	• Fine-tuning dapat memakan waktu beberapa jam hingga berhari-hari, tergantung pada ukuran model. Oleh karena itu, ini bukan solusi yang baik jika dokumen khusus Anda sering berubah. • Fine-tuning membutuhk an pemahaman tentang teknik, seperti adaptasi peringkat rendah (LoRa) dan fine-tuning hemat parameter (PEFT). Fine-tuni

Pendekatan	Keuntungan	Kekurangan
		ng Mungkin membutuhkan data scientist. • Fine-tuning mungkin tidak tersedia untuk semua model. • Fine-tuned model tidak memberikan referensi ke sumber dalam tanggapan mereka. • Mungkin ada peningkat an risiko halusinasi saat menggunakan model yang disetel dengan baik untuk menjawab pertanyaan.

Pendekatan	Keuntungan	Kekurangan
LAP	• RAG memungkinkan Anda membangun sistem penjawab pertanyaan untuk dokumen kustom Anda tanpa fine-tuning. • RAG dapat memasukkan dokumen terbaru dalam beberapa menit. • AWS menawarkan solusi RAG yang dikelola sepenuhnya. Oleh karena itu, tidak diperlukan ilmuwan data atau pengetahuan khusus tentang pembelajaran mesin. • Dalam tanggapannya, model RAG memberikan referensi ke sumber informasi. • Karena RAG menggunakan konteks dari pencarian vektor sebagai dasar jawaban yang dihasilkan, ada penurunan risiko halusinasi.	• RAG tidak berfungsi dengan baik saat merangkum informasi dari seluruh dokumen.

Jika Anda perlu membangun solusi penjawab pertanyaan yang mereferensikan dokumen kustom Anda, maka kami sarankan Anda memulai dari pendekatan. RAG-based Gunakan fine-tuning jika Anda membutuhkan model untuk melakukan tugas tambahan, seperti meringkas.

Anda dapat menggabungkan pendekatan fine-tuning dan RAG dalam satu model. Dalam kasus ini, arsitektur RAG tidak berubah, tetapi LLM yang menghasilkan jawaban juga disesuaikan dengan dokumen khusus. Ini menggabungkan yang terbaik dari kedua dunia, dan ini mungkin

merupakan solusi optimal untuk kasus penggunaan Anda. Untuk informasi lebih lanjut tentang cara menggabungkan fine-tuning yang diawasi dengan RAG, lihat penelitian RAFT: Adapting Language Model to Domain Specific RAG dari. University of California, Berkeley

Kasus penggunaan untuk Retrieval Augmented Generation

Berikut ini adalah kasus penggunaan umum untuk menggunakan pendekatan RAG:

- Mesin pencari — mesin RAG-enabled pencari dapat memberikan cuplikan fitur yang lebih akurat dan terkini dalam hasil pencarian mereka.
- Question-answering sistem — RAG dapat meningkatkan kualitas tanggapan dalam sistem penjawab pertanyaan. Model berbasis pengambilan menggunakan pencarian kesamaan untuk menemukan bagian atau dokumen yang relevan yang berisi jawabannya. Kemudian, menghasilkan respons yang ringkas dan relevan berdasarkan informasi itu.
- Ritel atau e-commerce — RAG dapat meningkatkan pengalaman pengguna dalam e-commerce dengan memberikan rekomendasi produk yang lebih relevan dan dipersonalisasi. Dengan mengambil dan menggabungkan informasi tentang preferensi pengguna dan detail produk, RAG dapat menghasilkan rekomendasi yang lebih akurat dan bermanfaat bagi pelanggan.
- Industri atau manufaktur - Di bidang manufaktur, RAG membantu Anda mengakses informasi penting dengan cepat, seperti operasi pabrik pabrik. Ini juga dapat membantu proses pengambilan keputusan, pemecahan masalah, dan inovasi organisasi. Untuk produsen yang beroperasi dalam kerangka peraturan yang ketat, RAG dapat dengan cepat mengambil peraturan dan standar kepatuhan yang diperbarui dari sumber internal dan eksternal, seperti dari standar industri atau badan pengatur.
- Perawatan Kesehatan — RAG memiliki potensi dalam industri perawatan kesehatan, di mana akses ke informasi yang akurat dan tepat waktu sangat penting. Dengan mengambil dan menggabungkan pengetahuan medis yang relevan dari sumber eksternal, RAG dapat memberikan tanggapan yang lebih akurat dan sadar konteks dalam aplikasi perawatan kesehatan. Aplikasi semacam itu menambah informasi yang dapat diakses oleh dokter manusia, yang pada akhirnya membuat panggilan dan bukan model.
- Legal — RAG dapat diterapkan secara kuat dalam skenario hukum, seperti merger dan akuisisi, di mana dokumen hukum yang kompleks memberikan konteks untuk pertanyaan. Ini dapat membantu para profesional hukum dengan cepat menavigasi masalah peraturan yang kompleks.

Opsi Retrieval Augmented Generation yang dikelola sepenuhnya pada AWS

Untuk mengelola alur kerja Retrieval Augmented Generation (RAG) AWS, Anda dapat menggunakan pipeline RAG khusus atau menggunakan beberapa kemampuan layanan yang dikelola sepenuhnya yang ditawarkan. AWS Karena mereka mencakup banyak komponen inti dari suatu RAG-based sistem, layanan yang dikelola sepenuhnya dapat membantu Anda mengelola beberapa angkat berat yang tidak berdiferensiasi. Namun, layanan ini memberikan lebih sedikit kesempatan untuk penyesuaian.

Konektor Layanan AWS penggunaan yang dikelola sepenuhnya untuk menyerap data dari sumber data eksternal, seperti situs web, Atlassian Confluence, atau Microsoft. SharePoint Sumber data yang didukung berbeda-beda Layanan AWS.

Bagian ini mengeksplorasi opsi terkelola penuh berikut untuk membangun alur kerja RAG pada: AWS

- Basis pengetahuan untuk Amazon Bedrock
- Amazon Q Bisnis
- Kanvas Amazon SageMaker AI

Untuk informasi selengkapnya tentang cara memilih di antara opsi-opsi ini, lihat Memilih opsi Retrieval Augmented Generation di AWS di panduan ini.

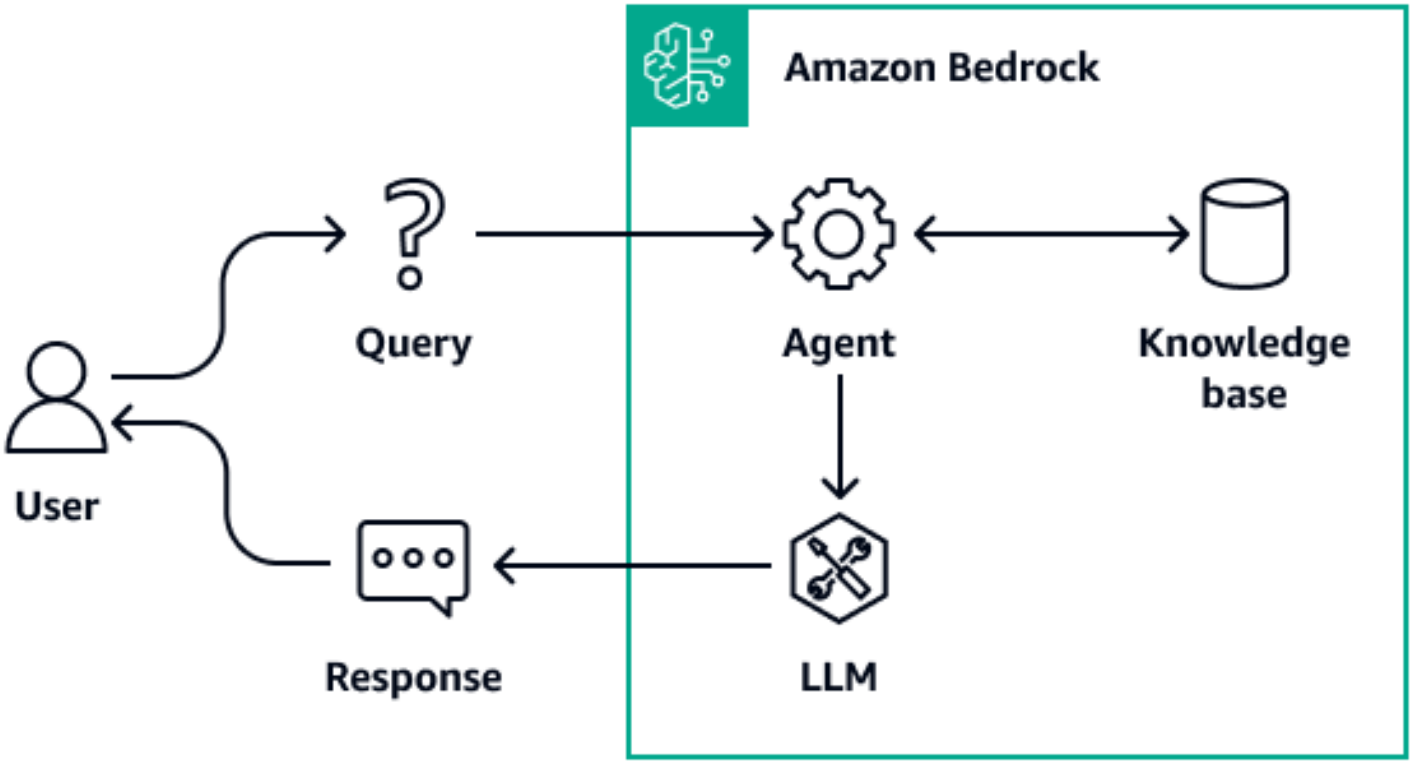
Basis pengetahuan untuk Amazon Bedrock

Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat model fondasi berkinerja tinggi (MM) dari startup AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu. Basis pengetahuan adalah kemampuan Amazon Bedrock yang membantu Anda menerapkan seluruh alur kerja RAG, mulai dari konsumsi hingga pengambilan dan augmentasi cepat. Tidak perlu membangun integrasi kustom ke sumber data atau untuk mengelola aliran data. Manajemen konteks sesi dibangun sehingga aplikasi AI generatif Anda dapat dengan mudah mendukung percakapan multi-putaran.

Setelah Anda menentukan lokasi data Anda, basis pengetahuan untuk Amazon Bedrock secara internal mengambil dokumen, memotongnya menjadi blok teks, mengubah teks menjadi embeddings,

dan kemudian menyimpan embeddings dalam database vektor pilihan Anda. Amazon Bedrock mengelola dan memperbarui penyematan, menjaga database vektor tetap sinkron dengan data. Untuk informasi selengkapnya tentang cara kerja basis pengetahuan, lihat Cara kerja basis pengetahuan Amazon Bedrock.

Jika Anda menambahkan basis pengetahuan ke agen Amazon Bedrock, agen mengidentifikasi basis pengetahuan yang sesuai berdasarkan masukan pengguna. Agen mengambil informasi yang relevan dan menambahkan informasi ke prompt input. Prompt yang diperbarui menyediakan model dengan lebih banyak informasi konteks untuk menghasilkan respons. Untuk meningkatkan transparansi dan meminimalkan halusinasi, informasi yang diambil dari basis pengetahuan dapat dilacak ke sumbernya.



Amazon Bedrock mendukung dua API berikut untuk RAG:

- RetrieveAndGenerate— Anda dapat menggunakan API ini untuk menanyakan basis pengetahuan Anda dan menghasilkan tanggapan dari informasi yang diambilnya. Secara internal, Amazon Bedrock mengubah kueri menjadi embeddings, menanyakan basis pengetahuan, menambah prompt dengan hasil pencarian sebagai informasi konteks, dan mengembalikan respons. LLM-generated Amazon Bedrock juga mengelola memori jangka pendek percakapan untuk memberikan hasil yang lebih kontekstual.

- Ambil - Anda dapat menggunakan API ini untuk menanyakan basis pengetahuan Anda dengan informasi yang diambil langsung dari basis pengetahuan. Anda dapat menggunakan informasi yang dikembalikan dari API ini untuk memproses teks yang diambil, mengevaluasi relevansinya, atau mengembangkan alur kerja terpisah untuk pembuatan respons. Secara internal, Amazon Bedrock mengubah kueri menjadi embeddings, mencari basis pengetahuan, dan mengembalikan hasil yang relevan. Anda dapat membuat alur kerja tambahan di atas hasil penelusuran. Misalnya, Anda dapat menggunakan LangChain`AmazonKnowledgeBasesRetriever`plugin untuk mengintegrasikan alur kerja RAG ke dalam aplikasi AI generatif.

Untuk contoh pola arsitektur dan petunjuk langkah demi langkah untuk menggunakan API, lihat Pangkalan Pengetahuan sekarang memberikan pengalaman RAG yang dikelola sepenuhnya di Amazon Bedrock (AWS posting blog). Untuk informasi selengkapnya tentang cara menggunakan `RetrieveAndGenerate` API untuk membangun alur kerja RAG untuk aplikasi berbasis obrolan cerdas, lihat Membangun aplikasi chatbot kontekstual menggunakan Pangkalan Pengetahuan Amazon Bedrock (posting blog).AWS

Sumber data untuk basis pengetahuan

Anda dapat menghubungkan data kepemilikan Anda ke basis pengetahuan. Setelah mengonfigurasi konektor sumber data, Anda dapat menyinkronkan atau memperbarui data dengan basis pengetahuan Anda dan membuat data Anda tersedia untuk kueri. Basis pengetahuan Amazon Bedrock mendukung koneksi ke sumber data berikut:

- Amazon Simple Storage Service (Amazon S3) - Anda dapat menghubungkan bucket Amazon S3 ke basis pengetahuan Amazon Bedrock dengan menggunakan konsol atau API. Basis pengetahuan mencerna dan mengindeks file dalam ember. Jenis sumber data ini mendukung fitur-fitur berikut:
  - Bidang metadata dokumen — Anda dapat menyertakan file terpisah untuk menentukan metadata file di bucket Amazon S3. Anda kemudian dapat menggunakan bidang metadata ini untuk memfilter dan meningkatkan relevansi tanggapan.
  - Filter penyertaan atau pengecualian — Anda dapat menyertakan atau mengecualikan konten tertentu saat merayapi.
  - Sinkronisasi tambahan — Perubahan konten dilacak, dan hanya konten yang telah berubah sejak sinkronisasi terakhir dirayapi.

- Confluence— Anda dapat menghubungkan Atlassian Confluence instance ke basis pengetahuan Amazon Bedrock dengan menggunakan konsol atau API. Jenis sumber data ini mendukung fitur-fitur berikut:
  - Deteksi otomatis bidang dokumen utama - Bidang metadata secara otomatis terdeteksi dan dirayapi. Anda dapat menggunakan bidang ini untuk pemfilteran.
  - Filter konten penyertaan atau pengecualian — Anda dapat menyertakan atau mengecualikan konten tertentu dengan menggunakan awalan atau pola ekspresi reguler pada spasi, judul halaman, judul blog, komentar, nama lampiran, atau ekstensi.
  - Sinkronisasi tambahan - Perubahan konten dilacak, dan hanya konten yang telah berubah sejak sinkronisasi terakhir dirayapi.
  - Otentikasi OAuth 2.0, otentikasi dengan token Confluence API — Kredensyal otentikasi disimpan di. AWS Secrets Manager
- Microsoft SharePoint— Anda dapat menghubungkan SharePoint instance ke basis pengetahuan dengan menggunakan konsol atau API. Jenis sumber data ini mendukung fitur-fitur berikut:
  - Deteksi otomatis bidang dokumen utama - Bidang metadata secara otomatis terdeteksi dan dirayapi. Anda dapat menggunakan bidang ini untuk pemfilteran.
  - Filter konten penyertaan atau pengecualian — Anda dapat menyertakan atau mengecualikan konten tertentu dengan menggunakan awalan atau pola ekspresi reguler pada judul halaman utama, nama acara, dan nama file (termasuk ekstensinya).
  - Sinkronisasi tambahan - Perubahan konten dilacak, dan hanya konten yang telah berubah sejak sinkronisasi terakhir dirayapi.
  - Otentikasi OAuth 2.0 — Kredensyal otentikasi disimpan di. AWS Secrets Manager
- Salesforce— Anda dapat menghubungkan Salesforce instance ke basis pengetahuan dengan menggunakan konsol atau API. Jenis sumber data ini mendukung fitur-fitur berikut:
  - Deteksi otomatis bidang dokumen utama - Bidang metadata secara otomatis terdeteksi dan dirayapi. Anda dapat menggunakan bidang ini untuk pemfilteran.
  - Filter konten penyertaan atau pengecualian — Anda dapat menyertakan atau mengecualikan konten tertentu dengan menggunakan awalan atau pola ekspresi reguler. Untuk daftar jenis konten yang dapat Anda terapkan filter, lihat Inclusion/exclusion filter di dokumentasi Amazon Bedrock.
  - Sinkronisasi tambahan — Perubahan konten dilacak, dan hanya konten yang telah berubah sejak sinkronisasi terakhir dirayapi.
  - Otentikasi OAuth 2.0 — Kredensyal otentikasi disimpan di. AWS Secrets Manager

- Web Crawler — Perayap Web Amazon Bedrock menghubungkan dan merayapi URL yang Anda berikan. Fitur-fitur berikut didukung:
  - Pilih beberapa URL untuk dirayapi
  - Hormati arahan robots.txt standar, seperti dan `Allow` `Disallow`
  - Kecualikan URL yang cocok dengan pola
  - Batasi tingkat perayapan
  - Di Amazon CloudWatch, lihat status setiap URL yang dirayapi

Untuk informasi selengkapnya tentang sumber data yang dapat Anda sambungkan ke basis pengetahuan Amazon Bedrock, lihat Membuat konektor sumber data untuk basis pengetahuan Anda.

Database vektor untuk basis pengetahuan

Ketika Anda mengatur koneksi antara basis pengetahuan dan sumber data, Anda harus mengkonfigurasi database vektor, juga dikenal sebagai penyimpanan vektor. Basis data vektor adalah tempat Amazon Bedrock menyimpan, memperbarui, dan mengelola embeddings yang mewakili data Anda. Setiap sumber data mendukung berbagai jenis database vektor. Untuk menentukan database vektor mana yang tersedia untuk sumber data Anda, lihat tipe sumber data.

Jika Anda lebih suka Amazon Bedrock untuk secara otomatis membuat database vektor di Amazon OpenSearch Tanpa Server untuk Anda, Anda dapat memilih opsi ini saat membuat basis pengetahuan. Namun, Anda juga dapat memilih untuk mengatur database vektor Anda sendiri. Jika Anda menyiapkan database vektor Anda sendiri, lihat Prasyarat untuk penyimpanan vektor Anda sendiri untuk basis pengetahuan. Setiap jenis database vektor memiliki prasyaratnya sendiri.

Bergantung pada tipe sumber data Anda, basis pengetahuan Amazon Bedrock mendukung database vektor berikut:

- Amazon Tanpa OpenSearch Server
- Edisi Amazon Aurora PostgreSQL-Compatible
- Pinecone(Pineconedokumentasi)
- Redis Enterprise Cloud(Redisdokumentasi)
- MongoDB Atlas(MongoDBdokumentasi)

Amazon Q Bisnis

Amazon Q Business adalah asisten bertenaga Generatif-AI yang dikelola sepenuhnya yang dapat Anda konfigurasikan untuk menjawab pertanyaan, memberikan ringkasan, menghasilkan konten, dan menyelesaikan tugas berdasarkan data perusahaan Anda. Ini memungkinkan pengguna akhir untuk menerima tanggapan langsung dan sadar izin dari sumber data perusahaan dengan kutipan.

Fitur utama

Kemampuan Amazon Q Business berikut ini dapat membantu Anda membangun aplikasi AI RAG-based generatif tingkat produksi:

- Built-in konektor — Amazon Q Business mendukung lebih dari 40 jenis konektor, seperti konektor untukAdobe Experience Manager (AEM),Salesforce,Jira, danMicrosoft SharePoint. Untuk daftar selengkapnya, lihat Konektor yang didukung. Jika Anda memerlukan konektor yang tidak didukung, Anda dapat menggunakan Amazon AppFlow untuk menarik data dari sumber data Anda ke Amazon Simple Storage Service (Amazon S3) dan kemudian menghubungkan Amazon Q Business ke bucket Amazon S3. Untuk daftar lengkap sumber data yang AppFlow didukung Amazon, lihat Aplikasi yang didukung.
- Built-in jaringan pipa pengindeksan — Amazon Q Business menyediakan pipeline bawaan untuk mengindeks data dalam database vektor. Anda dapat menggunakan AWS Lambda fungsi untuk menambahkan logika preprocessing untuk pipeline pengindeksan Anda.
- Opsi indeks — Anda dapat membuat dan menyediakan indeks asli di Amazon Q Business, dan Anda menggunakan retriever Amazon Q Business untuk menarik data dari indeks tersebut. Atau, Anda dapat menggunakan indeks Amazon Kendra yang telah dikonfigurasi sebelumnya sebagai retriever. Untuk informasi selengkapnya, lihat Membuat retriever untuk aplikasi Amazon Q Business.
- Model Foundation — Amazon Q Business menggunakan model foundation yang didukung di Amazon Bedrock. Untuk daftar lengkapnya, lihat Model foundation yang didukung di Amazon Bedrock.
- Plugin — Amazon Q Business menyediakan kemampuan untuk menggunakan plugin untuk berintegrasi dengan sistem target, seperti cara otomatis untuk meringkas informasi tiket dan pembuatan tiket. Jira Setelah dikonfigurasi, plugin dapat mendukung tindakan baca dan tulis yang dapat membantu Anda meningkatkan produktivitas pengguna akhir. Amazon Q Business mendukung dua jenis plugin: plugin bawaan dan plugin khusus.

- Pagar pembatas - Amazon Q Business mendukung kontrol global dan kontrol tingkat topik. Misalnya, kontrol ini dapat mendeteksi informasi identitas pribadi (PII), penyalahgunaan, atau informasi sensitif dalam permintaan. Untuk informasi selengkapnya, lihat Kontrol admin dan pagar pembatas di Amazon Q Business.
- Manajemen identitas — Dengan Amazon Q Business, Anda dapat mengelola pengguna dan akses mereka ke aplikasi AI RAG-based generatif. Untuk informasi selengkapnya, lihat Manajemen identitas dan akses untuk Amazon Q Business. Selain itu, konektor Amazon Q Business mengindeks informasi daftar kontrol akses (ACL) yang dilampirkan ke dokumen bersama dengan dokumen itu sendiri. Kemudian, Amazon Q Business menyimpan informasi ACL yang diindeksnya di Amazon Q Business User Store untuk membuat pemetaan pengguna dan grup serta memfilter respons obrolan berdasarkan akses pengguna akhir ke dokumen. Untuk informasi selengkapnya, lihat Konsep konektor sumber data.
- Pengayaan dokumen — Fitur pengayaan dokumen membantu Anda mengontrol dokumen dan atribut dokumen apa yang dicerna ke dalam indeks Anda dan juga bagaimana mereka dicerna. Ini dapat dicapai melalui dua pendekatan:
  - Konfigurasikan operasi dasar — Gunakan operasi dasar untuk menambah, memperbarui, atau menghapus atribut dokumen dari data Anda. Misalnya, Anda dapat menggosok data PII dengan memilih untuk menghapus atribut dokumen apa pun yang terkait dengan PII.
  - Konfigurasikan fungsi Lambda — Gunakan fungsi Lambda yang telah dikonfigurasi sebelumnya untuk melakukan logika manipulasi atribut dokumen lanjutan yang lebih disesuaikan ke data Anda. Misalnya, data perusahaan Anda mungkin disimpan sebagai gambar yang dipindai. Dalam hal ini, Anda dapat menggunakan fungsi Lambda untuk menjalankan pengenalan karakter optik (OCR) pada dokumen yang dipindai untuk mengekstrak teks darinya. Kemudian, setiap dokumen yang dipindai diperlakukan sebagai dokumen teks selama konsumsi. Terakhir, selama obrolan, Amazon Q akan memfaktorkan data tekstual yang diekstraksi dari dokumen yang dipindai saat menghasilkan respons.

  Ketika Anda menerapkan solusi Anda, Anda dapat memilih untuk menggabungkan kedua pendekatan pengayaan dokumen. Anda dapat menggunakan operasi dasar untuk melakukan parse pertama data Anda dan kemudian menggunakan fungsi Lambda untuk operasi yang lebih kompleks. Untuk informasi selengkapnya, lihat Pengayaan dokumen di Amazon Q Business.
- Integrasi — Setelah Anda membuat aplikasi Amazon Q Business, Anda dapat mengintegrasikannya ke aplikasi lain, seperti Slack atauMicrosoft Teams. Misalnya, lihat Menerapkan Slack gateway untukAmazon Q Business dan Menerapkan gateway Microsoft Teams untuk Amazon Q Business (AWS posting blog).

End-user kustomisasi

Amazon Q Business mendukung pengunggahan dokumen yang mungkin tidak disimpan dalam sumber data dan indeks organisasi Anda. Dokumen yang diunggah tidak disimpan. Mereka tersedia untuk digunakan hanya untuk percakapan di mana dokumen diunggah. Amazon Q Business mendukung jenis dokumen tertentu untuk diunggah. Untuk informasi selengkapnya, lihat Mengunggah file dan mengobrol di Amazon Q Business.

Amazon Q Business menyertakan fitur atribut pemfilteran berdasarkan dokumen. Baik administrator maupun pengguna akhir dapat menggunakan fitur ini. Administrator dapat menyesuaikan dan mengontrol respons obrolan untuk pengguna akhir dengan menggunakan atribut. Misalnya, jika tipe sumber data adalah atribut yang dilampirkan ke dokumen Anda, Anda dapat menentukan bahwa respons obrolan dibuat hanya dari sumber data tertentu. Atau, Anda dapat mengizinkan pengguna akhir untuk membatasi cakupan respons obrolan dengan menggunakan filter atribut yang telah Anda pilih.

Pengguna akhir dapat membuat Aplikasi Amazon Q yang ringan dan dibuat khusus dalam lingkungan aplikasi Amazon Q Business Anda yang lebih luas. Aplikasi Amazon Q memungkinkan otomatisasi tugas untuk domain tertentu, seperti aplikasi yang dibuat khusus untuk tim pemasaran.

Kanvas Amazon SageMaker AI

Amazon SageMaker AI Canvas membantu Anda menggunakan pembelajaran mesin untuk menghasilkan prediksi tanpa perlu menulis kode apa pun. Ini menyediakan antarmuka visual tanpa kode yang memberdayakan Anda untuk menyiapkan data, membangun, dan menerapkan model ML, merampingkan siklus hidup ML-ke-ujung dalam lingkungan terpadu. Kompleksitas persiapan data, pengembangan model, deteksi bias, penjelasan, dan pemantauan diabstraksikan jauh di belakang antarmuka yang intuitif. Pengguna tidak perlu ahli SageMaker AI atau operasi pembelajaran mesin (MLOP) untuk mengembangkan, mengoperasionalkan, dan memantau model dengan AI Canvas. SageMaker

Dengan SageMaker AI Canvas, fungsionalitas RAG disediakan melalui fitur kueri dokumen tanpa kode. Anda dapat memperkaya pengalaman obrolan di SageMaker AI Canvas dengan menggunakan indeks Amazon Kendra sebagai pencarian perusahaan yang mendasarinya. Untuk informasi selengkapnya, lihat Mengekstrak informasi dari dokumen dengan kueri dokumen.

Menghubungkan SageMaker AI Canvas ke indeks Amazon Kendra memerlukan pengaturan satu kali. Sebagai bagian dari konfigurasi domain, administrator cloud dapat memilih satu atau lebih

indeks Kendra yang dapat ditanyakan pengguna saat berinteraksi dengan Canvas. SageMaker Untuk petunjuk tentang cara mengaktifkan fitur kueri dokumen, lihat Memulai menggunakan Amazon SageMaker AI Canvas.

SageMaker AI Canvas mengelola komunikasi yang mendasari antara Amazon Kendra dan model pondasi yang dipilih. Untuk informasi lebih lanjut tentang model foundation yang didukung SageMaker AI Canvas, lihat Model foundation Generative AI di SageMaker AI Canvas. Diagram berikut menunjukkan cara kerja fitur kueri dokumen setelah administrator cloud menghubungkan SageMaker AI Canvas ke indeks Amazon Kendra.

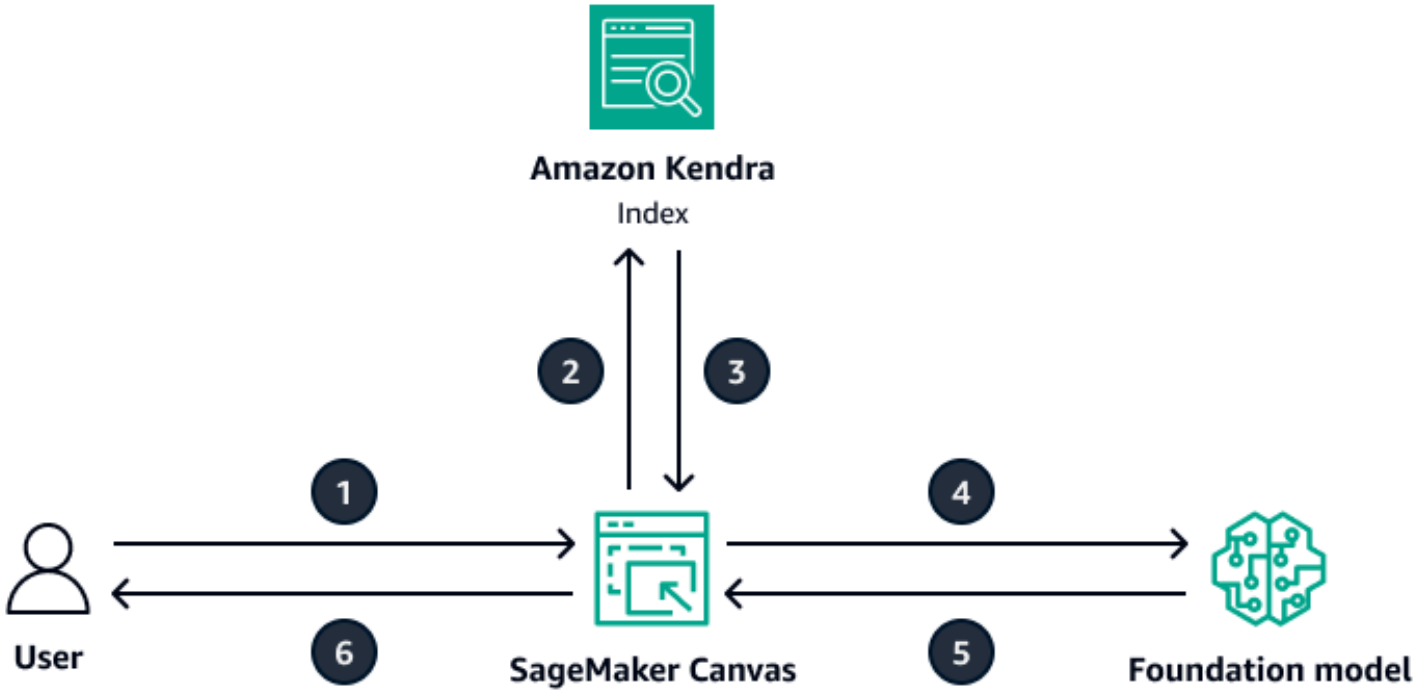


Diagram menunjukkan alur kerja berikut:

1. Pengguna memulai obrolan baru di SageMaker AI Canvas, mengaktifkan dokumen Kueri, memilih indeks target, dan kemudian mengirimkan pertanyaan.
2. SageMaker AI Canvas menggunakan kueri untuk mencari indeks Amazon Kendra untuk data yang relevan.
3. SageMaker AI Canvas mengambil data dan sumbernya dari indeks Amazon Kendra.
4. SageMaker AI Canvas memperbarui prompt untuk menyertakan konteks yang diambil dari indeks Amazon Kendra dan mengirimkan prompt ke model yayasan.
5. Model dasar menggunakan pertanyaan asli dan konteks yang diambil untuk menghasilkan jawaban.

6. SageMaker AI Canvas memberikan jawaban yang dihasilkan kepada pengguna. Ini termasuk referensi ke sumber data, seperti dokumen, yang digunakan untuk menghasilkan respons.

Arsitektur Generasi Augmented Retrieval Kustom di AWS

Bagian sebelumnya menjelaskan cara menggunakan Retrieval Augmented Generation (RAG) yang dikelola Layanan AWS sepenuhnya. Namun, beberapa kasus penggunaan memerlukan kontrol lebih besar atas komponen sistem, seperti retriever atau LLM (juga disebut generator). Misalnya, Anda mungkin memerlukan fleksibilitas untuk memilih database vektor Anda sendiri atau mengakses sumber data yang tidak didukung. Untuk kasus penggunaan ini, Anda dapat membangun arsitektur RAG kustom.

Bagian ini berisi topik berikut:

- Retriever untuk alur kerja RAG
- Generator untuk alur kerja RAG

Untuk informasi selengkapnya tentang cara memilih antara opsi retriever dan generator di bagian ini, lihat Memilih opsi Retrieval Augmented Generation di AWS di panduan ini.

Retriever untuk alur kerja RAG

Bagian ini menjelaskan cara membuat retriever. Anda dapat menggunakan solusi pencarian semantik yang dikelola sepenuhnya, seperti Amazon Kendra, atau Anda dapat membuat pencarian semantik kustom dengan menggunakan database vektor. AWS

Sebelum Anda meninjau opsi retriever, pastikan Anda memahami tiga langkah proses pencarian vektor:

1. Anda memisahkan dokumen yang perlu diindeks menjadi bagian-bagian yang lebih kecil. Ini disebut chunking.
2. Anda menggunakan proses yang disebut embedding untuk mengubah setiap potongan menjadi vektor matematika. Kemudian, Anda mengindeks setiap vektor dalam database vektor. Pendekatan yang Anda gunakan untuk mengindeks dokumen memengaruhi kecepatan dan akurasi pencarian. Pendekatan pengindeksan tergantung pada database vektor dan opsi konfigurasi yang disediakannya.
3. Anda mengonversi kueri pengguna menjadi vektor dengan menggunakan proses yang sama. Retriever mencari database vektor untuk vektor yang mirip dengan vektor kueri pengguna. Kesamaan dihitung dengan menggunakan metrik seperti jarak Euclidean, jarak kosinus, atau produk titik.

Panduan ini menjelaskan cara menggunakan layanan berikut Layanan AWS atau pihak ketiga untuk membuat lapisan pengambilan kustom pada AWS:

- Amazon Kendra
- OpenSearch Layanan Amazon
- Amazon Aurora PostgreSQL dan pgvector
- Analisis Amazon Neptunus
- Amazon MemoryDB
- Amazon DocumentDB
- Biji pinus
- MongoDB Atlas
- Weaviate

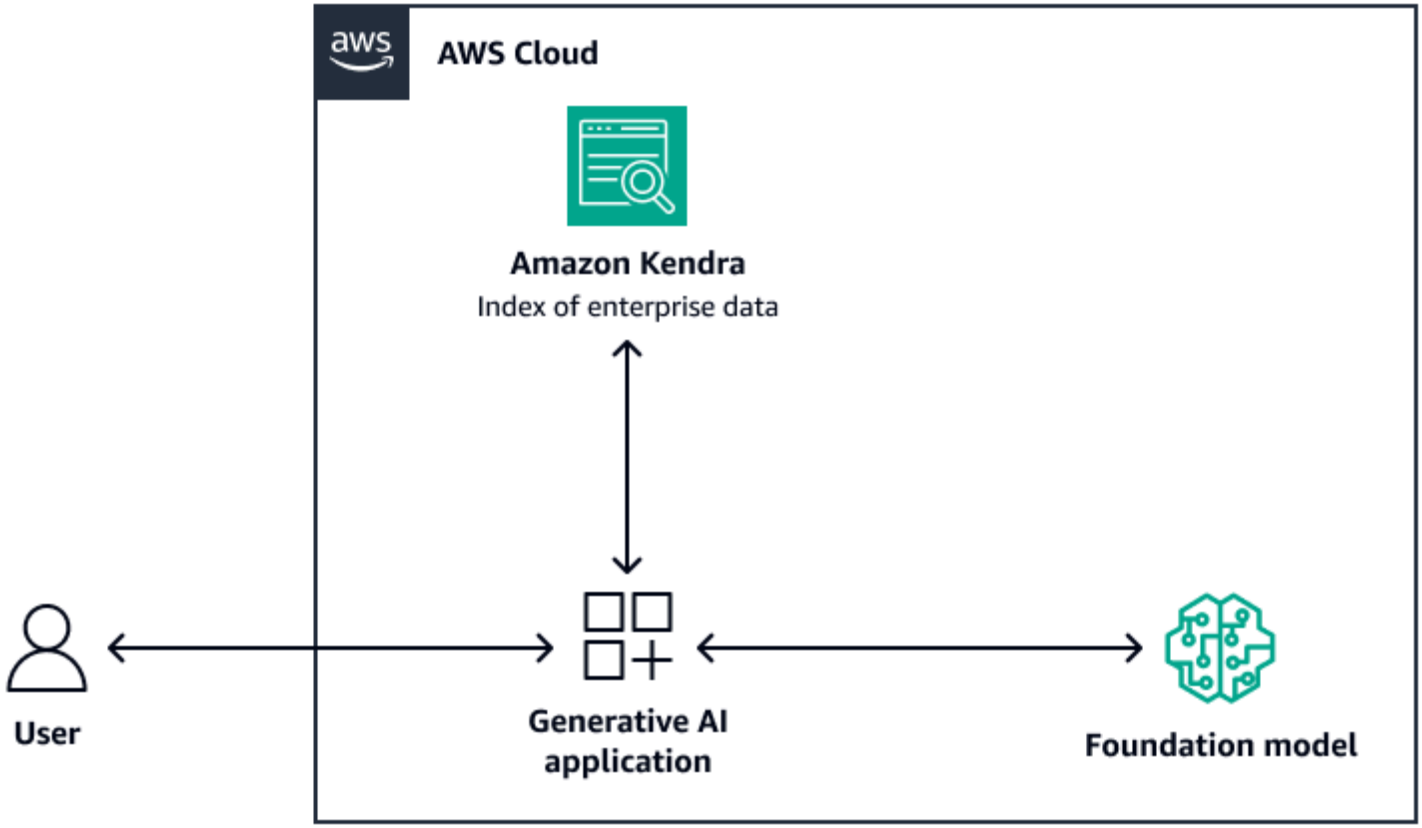
Amazon Kendra

Amazon Kendra adalah layanan pencarian cerdas yang dikelola sepenuhnya yang menggunakan pemrosesan bahasa alami dan algoritme pembelajaran mesin canggih untuk mengembalikan jawaban spesifik atas pertanyaan penelusuran dari data Anda. Amazon Kendra membantu Anda secara langsung menyerap dokumen dari berbagai sumber dan menanyakan dokumen setelah berhasil disinkronkan. Proses sinkronisasi menciptakan infrastruktur yang diperlukan untuk membuat pencarian vektor pada dokumen yang dicerna. Oleh karena itu, Amazon Kendra tidak memerlukan tiga langkah tradisional dari proses pencarian vektor. Setelah sinkronisasi awal, Anda dapat menggunakan jadwal yang ditentukan untuk menangani konsumsi yang sedang berlangsung.

Berikut ini adalah keuntungan menggunakan Amazon Kendra untuk RAG:

- Anda tidak perlu memelihara database vektor karena Amazon Kendra menangani seluruh proses pencarian vektor.
- Amazon Kendra berisi konektor bawaan untuk sumber data populer, seperti database, perayap situs web, bucket Amazon S3, instans, dan instance. Microsoft SharePoint Atlassian Confluence Konektor yang dikembangkan oleh AWS Mitra tersedia, seperti konektor untuk Box danGitLab.
- Amazon Kendra menyediakan pemfilteran daftar kontrol akses (ACL) yang hanya mengembalikan dokumen yang dapat diakses pengguna akhir.
- Amazon Kendra dapat meningkatkan respons berdasarkan metadata, seperti tanggal atau repositori sumber.

Gambar berikut menunjukkan contoh arsitektur yang menggunakan Amazon Kendra sebagai lapisan pengambilan sistem RAG. Untuk informasi selengkapnya, lihat Membangun aplikasi AI Generatif dengan akurasi tinggi dengan cepat pada data perusahaan menggunakan Amazon KendraLangChain,, dan model bahasa besarAWS (posting blog).



Untuk model foundation, Anda dapat menggunakan Amazon Bedrock atau LLM yang digunakan melalui Amazon AI. SageMaker JumpStart Anda dapat menggunakan AWS Lambda with LangChainuntuk mengatur aliran antara pengguna, Amazon Kendra, dan LLM. Untuk membangun sistem RAG yang menggunakan Amazon KendraLangChain,, dan berbagai LLM, lihat repositori Ekstensi Amazon Kendra. LangChain GitHub

OpenSearch Layanan Amazon

Amazon OpenSearch Service menyediakan algoritme HTML bawaan untuk penelusuran k-terdekat (k-NN) untuk melakukan pencarian vektor. OpenSearch Layanan juga menyediakan mesin vektor untuk Amazon EMR Tanpa Server. Anda dapat menggunakan mesin vektor ini untuk membangun sistem RAG yang memiliki kemampuan penyimpanan dan pencarian vektor yang dapat diskalakan dan berkinerja tinggi. Untuk informasi selengkapnya tentang cara membangun sistem RAG menggunakan OpenSearch Tanpa Server, lihat Membangun alur kerja RAG yang dapat diskalakan

dan tanpa server dengan mesin vektor untuk model Amazon OpenSearch Tanpa Server dan Amazon Bedrock Claude (posting blog).AWS

Berikut ini adalah keuntungan menggunakan OpenSearch Service untuk pencarian vektor:

- Ini memberikan kontrol penuh atas database vektor, termasuk membangun pencarian vektor terukur dengan menggunakan OpenSearch Serverless.
- Ini memberikan kontrol atas strategi chunking.
- Ini menggunakan algoritma perkiraan tetangga terdekat (ANN) dari perpustakaan Non-Metric Space Library (NMSLIB), Faiss, dan Apache Lucene untuk menyalakan pencarian K-nn. Anda dapat mengubah algoritme berdasarkan kasus penggunaan. Untuk informasi selengkapnya tentang opsi untuk menyesuaikan pencarian vektor melalui OpenSearch Layanan, lihat kemampuan database vektor Amazon OpenSearch Service dijelaskan (posting AWS blog).
- OpenSearch Tanpa server terintegrasi dengan basis pengetahuan Amazon Bedrock sebagai indeks vektor.

Amazon Aurora PostgreSQL dan pgvector

Amazon Aurora PostgreSQL-Compatible Edition adalah mesin database relasional terkelola penuh yang membantu Anda mengatur, mengoperasikan, dan menskalakan penerapan PostgreSQL. pgvector adalah ekstensi PostgreSQL open-source yang menyediakan kemampuan pencarian kesamaan vektor. Ekstensi ini tersedia untuk Aurora PostgreSQL-Compatible dan Amazon Relational Database Service (Amazon RDS) untuk PostgreSQL. Untuk informasi selengkapnya tentang cara membangun RAG-based sistem yang menggunakan Aurora PostgreSQL-Compatible dan pgvector, lihat posting blog berikut: AWS

- Membangun AI-powered pencarian di PostgreSQL menggunakan Amazon AI dan pgvector SageMaker
- Manfaatkan pgvector dan Amazon Aurora PostgreSQL untuk Pemrosesan Bahasa Alami, Chatbots, dan Analisis Sentimen

Berikut ini adalah keuntungan menggunakan pgvector dan Aurora: PostgreSQL-Compatible

- Ini mendukung pencarian tetangga terdekat yang tepat dan perkiraan. Ini juga mendukung metrik kesamaan berikut: jarak L2, produk dalam, dan jarak kosinus.

- Ini mendukung pengindeksan File Terbalik dengan Kompresi Datar (iVFFlat) dan Hierarchical Navigable Small Worlds (HNSW).
- Anda dapat menggabungkan pencarian vektor dengan kueri atas data spesifik domain yang tersedia dalam instance PostgreSQL yang sama.
- Aurora PostgreSQL-Compatible dioptimalkan untuk I/O dan menyediakan caching berjenjang. Untuk beban kerja yang melebihi memori instans yang tersedia, pgvector dapat meningkatkan kueri per detik untuk pencarian vektor hingga 8 kali.

Analisis Amazon Neptunus

Amazon Neptune Analytics adalah mesin database grafik yang dioptimalkan untuk memori untuk analitik. Ini mendukung perpustakaan algoritma analitik grafik yang dioptimalkan, kueri grafik latensi rendah, dan kemampuan pencarian vektor dalam traversal grafik. Ini juga memiliki pencarian kesamaan vektor bawaan. Ini menyediakan satu titik akhir untuk membuat grafik, memuat data, memanggil kueri, dan melakukan pencarian kesamaan vektor. Untuk informasi selengkapnya tentang cara membangun RAG-based sistem yang menggunakan Neptunus Analytics, lihat Menggunakan grafik pengetahuan untuk membangun aplikasi GraphRag dengan Amazon Bedrock dan Amazon Neptunus (posting blog).AWS

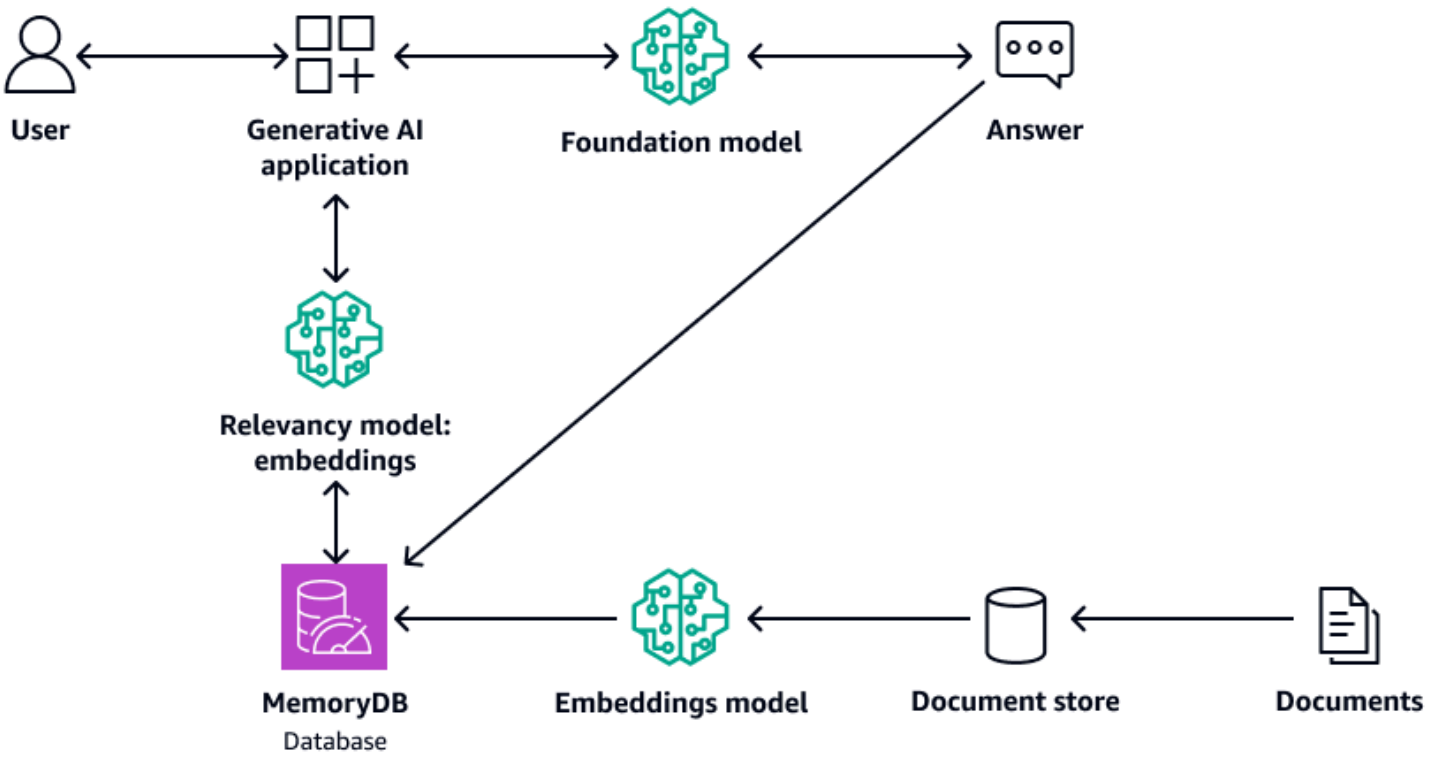
Berikut ini adalah keuntungan menggunakan Neptune Analytics:

- Anda dapat menyimpan dan mencari embeddings dalam kueri grafik.
- Jika Anda mengintegrasikan Neptunus Analytics LangChain dengan, arsitektur ini mendukung kueri grafik bahasa alami.
- Arsitektur ini menyimpan dataset grafik besar dalam memori.

Amazon MemoryDB

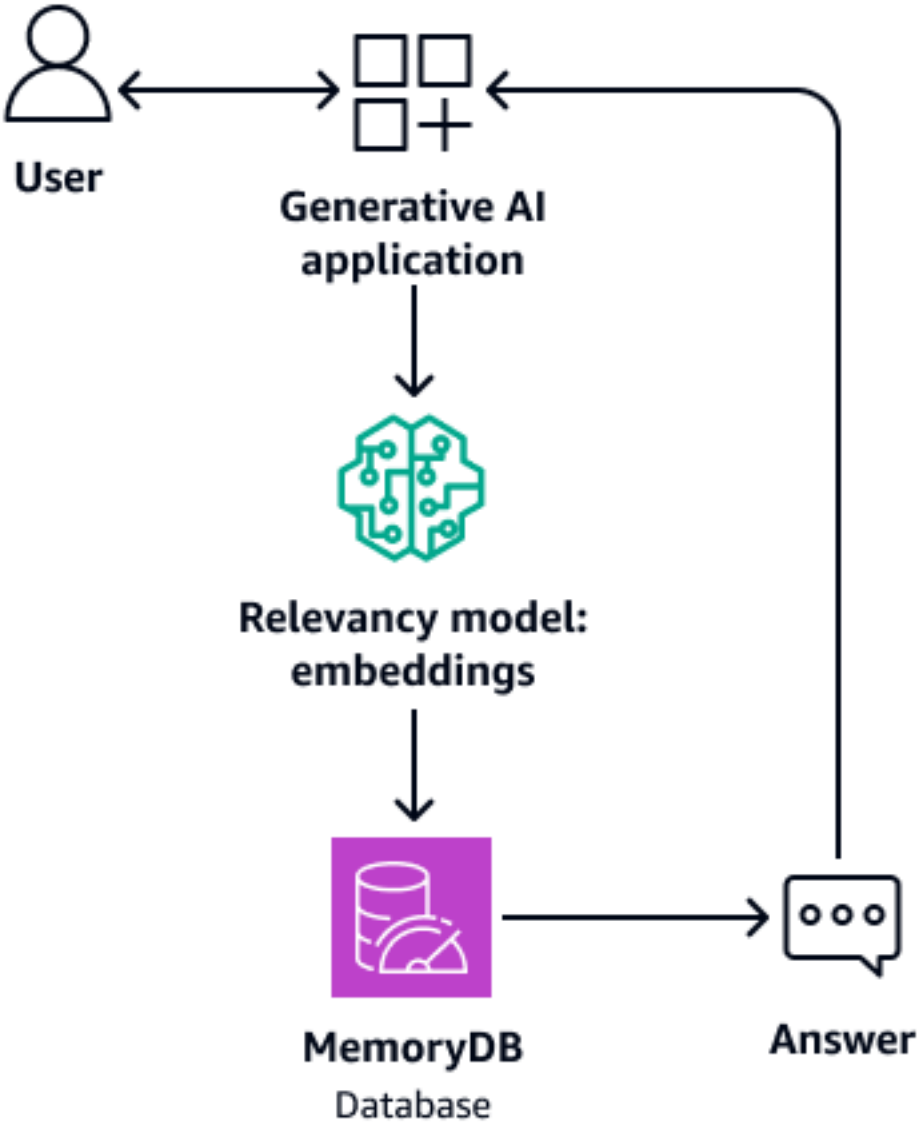
Amazon MemoryDB adalah layanan database dalam memori yang tahan lama yang memberikan kinerja sangat cepat. Semua data Anda disimpan dalam memori, yang mendukung pembacaan mikrodetik, latensi tulis milidetik satu digit, dan throughput tinggi. Pencarian vektor untuk MemoryDB memperluas fungsionalitas MemoryDB dan dapat digunakan bersama dengan fungsionalitas MemoryDB yang ada. Untuk informasi lebih lanjut, lihat Pertanyaan menjawab dengan LLM dan repositori RAG di. GitHub

Diagram berikut menunjukkan arsitektur sampel yang menggunakan MemoryDB sebagai database vektor.



Berikut ini adalah keuntungan menggunakan MemoryDB:

- Ini mendukung algoritma pengindeksan Flat dan HNSW. Untuk informasi lebih lanjut, lihat Pencarian vektor untuk Amazon MemoryDB sekarang tersedia secara umum di Blog Berita AWS
- Ini juga dapat bertindak sebagai memori penyangga untuk model pondasi. Ini berarti bahwa pertanyaan yang dijawab sebelumnya diambil dari buffer alih-alih melalui proses pengambilan dan pembuatan lagi. Diagram berikut menunjukkan proses ini.



- Karena menggunakan database dalam memori, arsitektur ini menyediakan waktu kueri milidetik satu digit untuk pencarian semantik.
- Ini menyediakan hingga 33.000 kueri per detik pada penarikan 95-99% dan 26.500 kueri per detik dengan penarikan lebih dari 99%. Untuk informasi lebih lanjut, lihat AWS re:Invent 2023 - pencarian Ultra-low vektor latensi untuk video Amazon MemoryDB di. YouTube

Amazon DocumentDB

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan database yang cepat, andal, dan terkelola sepenuhnya. Ini membuatnya mudah untuk mengatur, mengoperasikan, dan menskalakan basis data MongoDB yang kompatibel di cloud. Pencarian vektor untuk Amazon

DocumentDB menggabungkan fleksibilitas dan kemampuan query yang kaya dari database dokumen JSON-based dengan kekuatan pencarian vektor. Untuk informasi lebih lanjut, lihat Pertanyaan menjawab dengan LLM dan repositori RAG di. GitHub

Diagram berikut menunjukkan contoh arsitektur yang menggunakan Amazon DocumentDB sebagai database vektor.

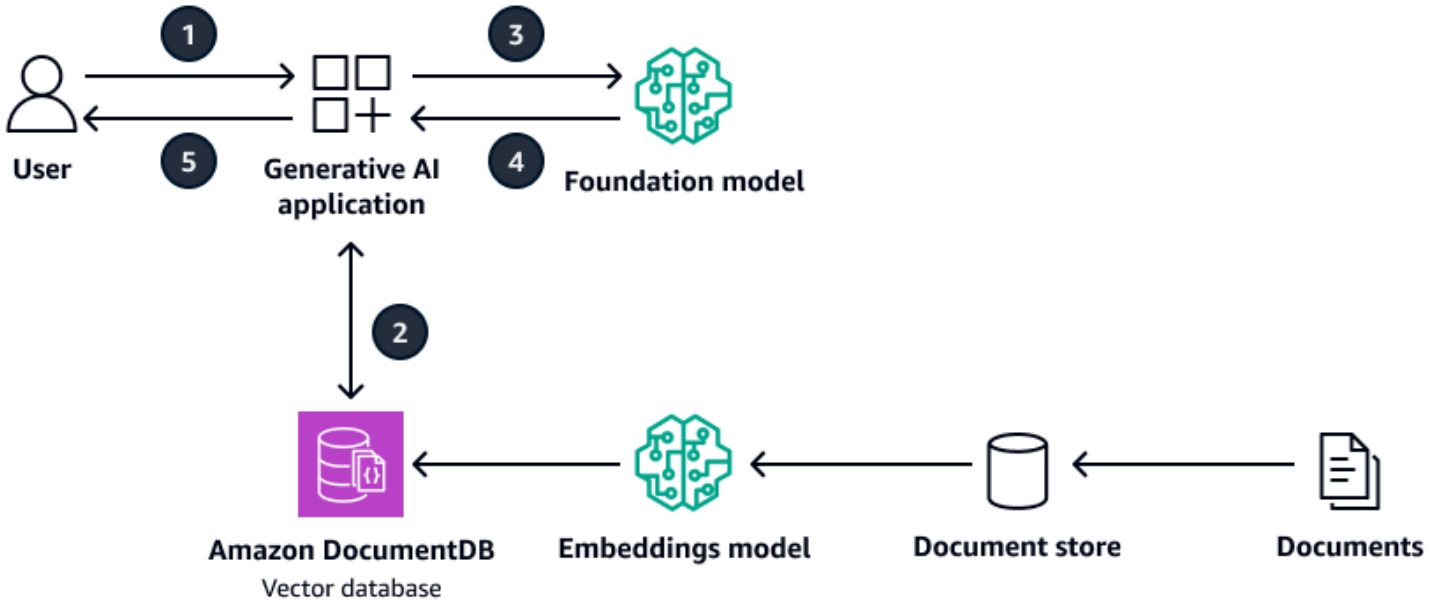


Diagram menunjukkan alur kerja berikut:

1. Pengguna mengirimkan kueri ke aplikasi AI generatif.
2. Aplikasi AI generatif melakukan pencarian kesamaan di database vektor Amazon DocumentDB dan mengambil ekstrak dokumen yang relevan.
3. Aplikasi AI generatif memperbarui kueri pengguna dengan konteks yang diambil dan mengirimkan prompt ke model pondasi target.
4. Model dasar menggunakan konteks untuk menghasilkan respons terhadap pertanyaan pengguna dan mengembalikan respons.
5. Aplikasi AI generatif mengembalikan respons kepada pengguna.

Berikut ini adalah keuntungan menggunakan Amazon DocumentDB:

- Ini mendukung metode pengindeksan HNSW dan iVFFlat.
- Ini mendukung hingga 2.000 dimensi dalam data vektor dan mendukung metrik jarak produk Euclidean, cosinus, dan titik.

- Ini memberikan waktu respons milidetik.

Biji pinus

Pineconeadalah database vektor yang dikelola sepenuhnya yang membantu Anda menambahkan pencarian vektor ke aplikasi produksi. Ini tersedia melalui AWS Marketplace. Penagihan didasarkan pada penggunaan, dan biaya dihitung dengan mengalikan harga pod dengan jumlah pod. Untuk informasi selengkapnya tentang cara membangun RAG-based sistem yang menggunakanPinecone, lihat posting AWS blog berikut:

- Mengurangi halusinasi melalui RAG menggunakan Pinecone database vektor & dari Amazon AI Llama-2 SageMaker JumpStart
- Gunakan Amazon SageMaker AI Studio untuk membuat solusi penjawab pertanyaan RAG dengan Llama 2LangChain, dan Pinecone untuk eksperimen cepat

Diagram berikut menunjukkan arsitektur sampel yang menggunakan Pinecone sebagai database vektor.

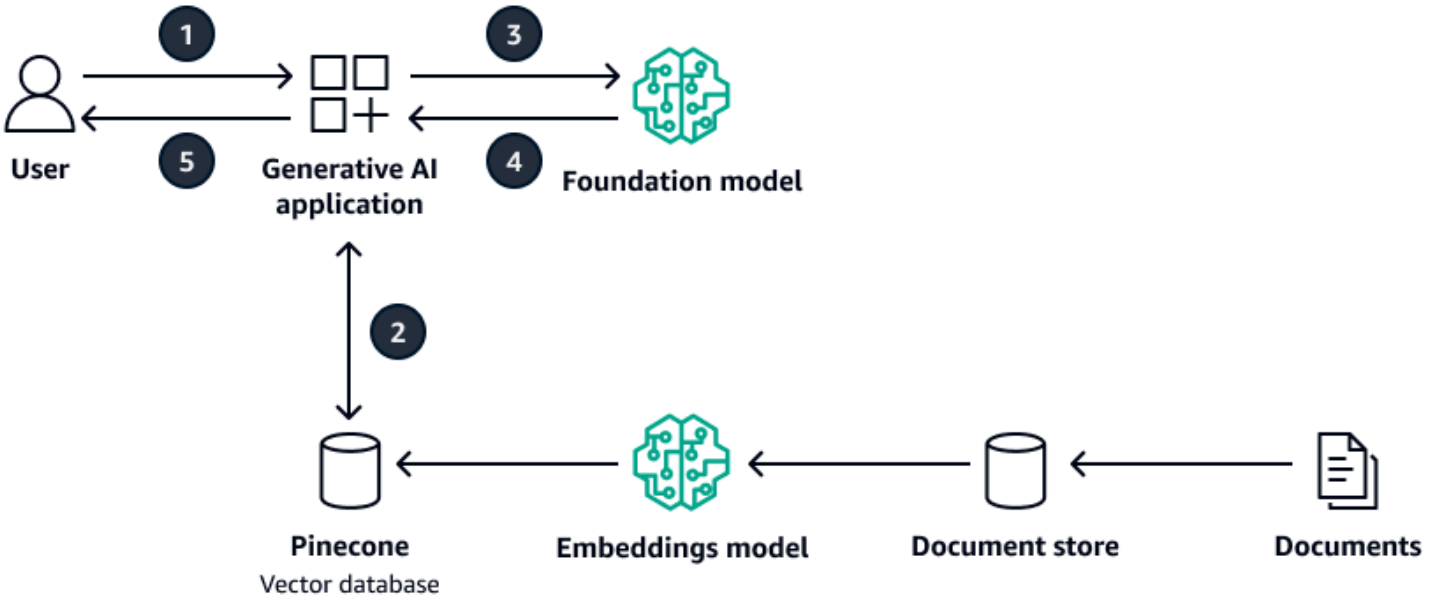


Diagram menunjukkan alur kerja berikut:

1. Pengguna mengirimkan kueri ke aplikasi AI generatif.
2. Aplikasi AI generatif melakukan pencarian kesamaan dalam database Pinecone vektor dan mengambil ekstrak dokumen yang relevan.

3. Aplikasi AI generatif memperbarui kueri pengguna dengan konteks yang diambil dan mengirimkan prompt ke model pondasi target.
4. Model dasar menggunakan konteks untuk menghasilkan respons terhadap pertanyaan pengguna dan mengembalikan respons.
5. Aplikasi AI generatif mengembalikan respons kepada pengguna.

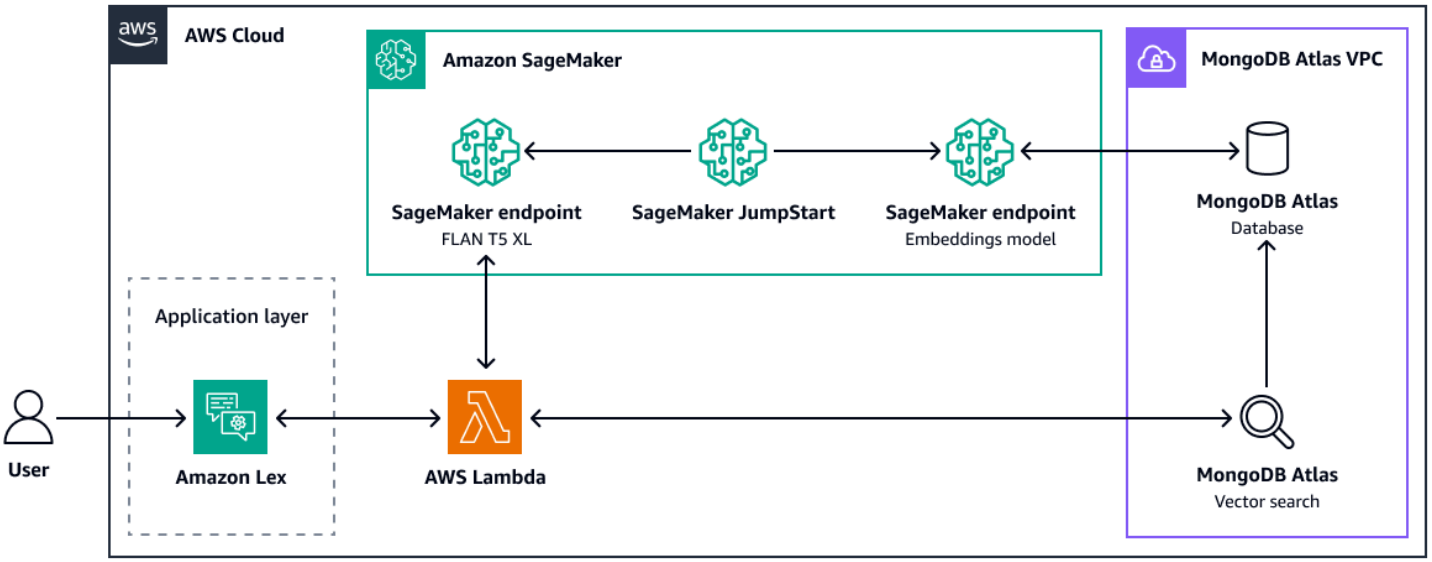
Berikut ini adalah keuntungan menggunakanPinecone:

- Ini adalah database vektor yang dikelola sepenuhnya dan menghilangkan biaya pengelolaan infrastruktur Anda sendiri.
- Ini menyediakan fitur tambahan penyaringan, pembaruan indeks langsung, dan peningkatan kata kunci (pencarian hibrida).

MongoDB Atlas

MongoDB Atlasadalah database cloud yang dikelola sepenuhnya yang menangani semua kompleksitas penerapan dan pengelolaan penerapan Anda. AWS Anda dapat menggunakan pencarian Vektor MongoDB Atlas untuk menyimpan embeddings vektor dalam database Anda. MongoDB Basis pengetahuan Amazon Bedrock mendukung MongoDB Atlas penyimpanan vektor. Untuk informasi selengkapnya, lihat Memulai Integrasi Basis Pengetahuan Amazon Bedrock dalam MongoDB dokumentasi.

Untuk informasi selengkapnya tentang cara menggunakan pencarian MongoDB Atlas vektor untuk RAG, lihat Retrieval-Augmented Generasi denganLangChain, Amazon SageMaker AI JumpStart, dan Pencarian MongoDB Atlas Semantik (AWS posting blog). Diagram berikut menunjukkan arsitektur solusi rinci dalam posting blog ini.



Berikut ini adalah keuntungan menggunakan pencarian MongoDB Atlas vektor:

- Anda dapat menggunakan implementasi yang ada MongoDB Atlas untuk menyimpan dan mencari embeddings vektor.
- Anda dapat menggunakan MongoDBQuery API untuk menanyakan embeddings vektor.
- Anda dapat secara mandiri menskalakan pencarian vektor dan database.
- Penyematan vektor disimpan di dekat data sumber (dokumen), yang meningkatkan kinerja pengindeksan.

Weaviate

Weaviateadalah sumber terbuka populer, database vektor latensi rendah yang mendukung jenis media multimodal, seperti teks dan gambar. Basis data menyimpan objek dan vektor, yang menggabungkan pencarian vektor dengan penyaringan terstruktur. Untuk informasi selengkapnya tentang penggunaan Weaviate dan Amazon Bedrock untuk membangun alur kerja RAG, lihat Membangun solusi AI generatif siap perusahaan dengan model dasar Cohere di Amazon Bedrock dan Weaviate database vektor di (posting blog). AWS MarketplaceAWS

Berikut ini adalah keuntungan menggunakanWeaviate:

- Ini adalah open source dan didukung oleh komunitas yang kuat.
- Itu dibangun untuk pencarian hibrida (baik vektor dan kata kunci).

- Anda dapat menerapkannya sebagai penawaran perangkat lunak terkelola AWS sebagai layanan (SaaS) atau sebagai klaster Kubernetes.

Generator untuk alur kerja RAG

Model bahasa besar (LLM) adalah model pembelajaran mendalam yang sangat besar yang dilatih sebelumnya pada sejumlah besar data. Mereka sangat fleksibel. LLM dapat melakukan berbagai tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan bahasa, dan menyelesaikan kalimat. Mereka memiliki potensi untuk mengganggu pembuatan konten dan cara orang menggunakan mesin pencari dan asisten virtual. Meskipun tidak sempurna, LLM menunjukkan kemampuan luar biasa untuk membuat prediksi berdasarkan prompt atau jumlah input yang relatif kecil.

LLM adalah komponen penting dari solusi RAG. Untuk arsitektur RAG kustom, ada dua Layanan AWS yang berfungsi sebagai opsi utama:

- Amazon Bedrock adalah layanan yang dikelola sepenuhnya yang membuat LLM dari perusahaan AI terkemuka dan Amazon tersedia untuk Anda gunakan melalui API terpadu.
- Amazon SageMaker AI JumpStart adalah hub ML yang menawarkan model dasar, algoritme bawaan, dan solusi HTML bawaan. Dengan SageMaker AI JumpStart, Anda dapat mengakses model yang telah dilatih sebelumnya, termasuk model pondasi. Anda juga dapat menggunakan data Anda sendiri untuk menyempurnakan model yang telah dilatih sebelumnya.

Amazon Bedrock

Amazon Bedrock menawarkan model industri terkemuka dariAnthropic,,,,Stability AI,Meta, CohereAI21 Labs, Mistral AI dan Amazon. Untuk daftar lengkapnya, lihat Model foundation yang didukung di Amazon Bedrock. Amazon Bedrock juga memungkinkan Anda untuk menyesuaikan model dengan data Anda sendiri.

Anda dapat mengevaluasi kinerja model untuk menentukan mana yang paling cocok untuk kasus penggunaan RAG Anda. Anda dapat menguji model terbaru dan juga menguji untuk melihat kemampuan dan fitur mana yang memberikan hasil terbaik dan dengan harga terbaik. Model Anthropic Claude Sonnet adalah pilihan umum untuk aplikasi RAG karena unggul dalam berbagai tugas dan memberikan tingkat keandalan dan prediktabilitas yang tinggi.

SageMaker AI JumpStart

SageMaker AI JumpStart menyediakan model open source yang sudah terlatih untuk berbagai jenis masalah. Anda dapat melatih dan menyempurnakan model-model ini secara bertahap sebelum penerapan. Anda dapat mengakses model, templat solusi, dan contoh yang telah dilatih sebelumnya melalui halaman JumpStart arahan SageMaker AI di Amazon SageMaker AI Studio atau menggunakan AI Python SageMaker SDK.

SageMaker AI JumpStart menawarkan model dasar canggih untuk kasus penggunaan seperti penulisan konten, pembuatan kode, penjawab pertanyaan, copywriting, ringkasan, klasifikasi, pengambilan informasi, dan banyak lagi. Gunakan model JumpStart foundation untuk membuat solusi AI generatif Anda sendiri dan mengintegrasikan solusi khusus dengan fitur SageMaker AI tambahan. Untuk informasi selengkapnya, lihat Memulai dengan Amazon SageMaker AI JumpStart.

SageMaker AI melakukan JumpStart onboard dan memelihara model foundation yang tersedia untuk umum agar Anda dapat mengakses, menyesuaikan, dan mengintegrasikan ke dalam siklus hidup ML Anda. Untuk informasi selengkapnya, lihat Model foundation yang tersedia untuk umum. SageMaker AI JumpStart juga mencakup model yayasan berpemilik dari penyedia pihak ketiga. Untuk informasi selengkapnya, lihat Model foundation proprietary.

Memilih opsi Retrieval Augmented Generation di AWS

Opsi RAG yang dikelola sepenuhnya dan bagian arsitektur RAG Kustom dari panduan ini menjelaskan berbagai pendekatan untuk membangun solusi pencarian berbasis RAG. AWS Bagian ini menjelaskan cara memilih di antara opsi-opsi ini berdasarkan kasus penggunaan Anda. Dalam beberapa situasi, lebih dari satu opsi mungkin berhasil. Dalam skenario itu, pilihan tergantung pada kemudahan implementasi, keterampilan yang tersedia di organisasi Anda, dan kebijakan dan standar perusahaan Anda.

Kami menyarankan Anda mempertimbangkan opsi RAG yang dikelola sepenuhnya dan kustom dalam urutan berikut dan memilih opsi pertama yang sesuai dengan kasus penggunaan Anda:

1. Gunakan Amazon Q Business kecuali:
   - Layanan ini tidak tersedia di Anda Wilayah AWS, dan data Anda tidak dapat dipindahkan ke Wilayah yang tersedia
   - Anda memiliki alasan khusus untuk menyesuaikan alur kerja RAG
   - Anda ingin menggunakan database vektor yang ada atau LLM tertentu
2. Gunakan basis pengetahuan untuk Amazon Bedrock kecuali:
   - Anda memiliki database vektor yang tidak didukung
   - Anda memiliki alasan khusus untuk menyesuaikan alur kerja RAG
3. Gabungkan Amazon Kendra dengan generator pilihan Anda kecuali:
   - Anda ingin memilih database vektor Anda sendiri
   - Anda ingin menyesuaikan strategi chunking
4. Jika Anda ingin lebih banyak kontrol atas retriever dan ingin memilih database vektor Anda sendiri:
   - Jika Anda tidak memiliki database vektor yang ada dan tidak memerlukan latensi rendah atau kueri grafik, pertimbangkan untuk menggunakan Amazon OpenSearch Service.
   - Jika Anda memiliki database PostgreSQL vektor yang ada, pertimbangkan untuk menggunakan Amazon Aurora PostgreSQL dan opsi. pgvector
   - Jika Anda membutuhkan latensi rendah, pertimbangkan opsi dalam memori, seperti Amazon MemoryDB atau Amazon DocumentDB.
   - Jika Anda ingin menggabungkan pencarian vektor dengan kueri grafik, pertimbangkan Amazon Neptunus Analytics.
   - Jika Anda sudah menggunakan database vektor pihak ketiga atau menemukan manfaat tertentu darinya, pertimbangkan PineconeMongoDB Atlas, dan Weaviate.

5. Jika Anda ingin memilih LLM:
   - Jika Anda menggunakan Amazon Q Business, Anda tidak dapat memilih LLM.
   - Jika Anda menggunakan Amazon Bedrock, Anda dapat memilih salah satu model pondasi yang didukung.
   - Jika Anda menggunakan Amazon Kendra atau database vektor kustom, Anda dapat menggunakan salah satu generator yang dijelaskan dalam panduan ini atau menggunakan LLM kustom.

   > **Note**
   >
   > Anda juga dapat menggunakan dokumen kustom Anda untuk menyempurnakan LLM yang ada untuk meningkatkan akurasi tanggapannya. Untuk informasi selengkapnya, lihat Membandingkan RAG dan fine-tuning dalam panduan ini.

6. Jika Anda memiliki implementasi Amazon SageMaker AI Canvas yang ingin Anda gunakan atau jika Anda ingin membandingkan respons RAG dari yang berbeda LLMs, pertimbangkan Amazon SageMaker AI Canvas.

Kesimpulan

Panduan ini menjelaskan berbagai opsi untuk membangun sistem Retrieval Augmented Generation (RAG). AWS Anda dapat memulai dengan layanan yang dikelola sepenuhnya, seperti Amazon Q Business dan basis pengetahuan Amazon Bedrock. Jika Anda ingin lebih banyak kontrol atas alur kerja RAG, Anda dapat memilih retriever kustom. Untuk generator, Anda dapat menggunakan API untuk memanggil LLM yang didukung di Amazon Bedrock, atau Anda dapat menerapkan LLM Anda sendiri dengan menggunakan Amazon AI. SageMaker JumpStart Tinjau rekomendasi dalam Memilih opsi RAG untuk menentukan opsi mana yang paling cocok untuk kasus penggunaan Anda. Setelah Anda memilih opsi terbaik untuk kasus penggunaan Anda, gunakan referensi yang disediakan dalam panduan ini untuk mulai membangun aplikasi berbasis RAG Anda.

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	Oktober 28, 2024

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