Panduan Implementasi

Pembuat Aplikasi AI Generatif di AWS

Pembuat Aplikasi AI Generatif di AWS: Panduan Implementasi

Table of Contents

Solusi ini memfasilitasi pengembangan, eksperimen cepat, dan penyebaran aplikasi kecerdasan buatan generatif (AI)

Generative AI Application Builder on AWS memfasilitasi pengembangan, eksperimen cepat, dan penerapan aplikasi kecerdasan buatan (AI) generatif tanpa memerlukan pengalaman mendalam dalam AI. AWS Solution ini mempercepat pengembangan dan menyederhanakan eksperimen dengan membantu Anda:

- Menyerap data dan dokumen spesifik bisnis Anda
- Mengevaluasi dan membandingkan kinerja model bahasa besar (LLMs)
- Jalankan tugas dan alur kerja multi-langkah dengan agen AI
- Bangun aplikasi yang dapat diperluas dengan cepat, dan gunakan aplikasi tersebut dengan arsitektur tingkat perusahaan

Generative AI Application Builder di AWS mencakup integrasi dengan:

- LLMs tersedia di Amazon Bedrock
- LLMs yang telah Anda gunakan di Amazon AI SageMaker
- Basis Pengetahuan Amazon Bedrock untuk Generasi Retrieval-Augmented (RAG)
- Amazon Bedrock Guardrails untuk menerapkan perlindungan dan mengurangi halusinasi
- Amazon Bedrock Agents untuk membangun alur kerja agen yang dapat melakukan orkestrasi dan penyelesaian tugas
- Amazon Bedrock AgentCore untuk membangun, menerapkan, dan mengelola agen AI siap produksi dengan dukungan runtime yang diperpanjang
- Server Model Context Protocol (MCP) untuk data perusahaan dan integrasi alat

Selain itu, solusi ini memungkinkan koneksi ke model pilihan Anda dengan menggunakan LangChain konektor. Konektor ini tersedia dalam fungsi AWS Lambda yang digunakan dengan solusi. Anda dapat memulai dengan panduan penerapan tanpa kode untuk membangun aplikasi AI generatif untuk pencarian percakapan, chatbots yang dihasilkan AI, pembuatan teks, dan ringkasan teks.

Panduan implementasi ini memberikan gambaran umum tentang solusi Generative AI Application Builder on AWS, arsitektur dan komponen referensinya, pertimbangan untuk merencanakan

penerapan, dan langkah-langkah konfigurasi untuk menerapkan solusi ke Amazon Web Services (AWS) Cloud.

Panduan ini ditujukan untuk arsitek solusi, pengambil keputusan bisnis, DevOps insinyur, ilmuwan data, dan profesional cloud yang ingin menerapkan Generative AI Application Builder di AWS di lingkungan mereka.

Gunakan tabel navigasi ini untuk menemukan jawaban atas pertanyaan-pertanyaan ini dengan cepat:

Jika kau mau.	Baca..
Ketahui biaya untuk menjalankan solusi ini. Perkiraan biaya untuk menjalankan solusi ini bervariasi berdasarkan komponen yang Anda gunakan dan jumlah kueri. Biaya untuk menjalankan dasbor Deployment dengan parameter default dan 100 pengguna aktif di Wilayah AS Timur (Virginia) selama satu bulan adalah sekitar $20,12 USD per bulan. Biaya untuk kasus penggunaan Teks yang digunakan tanpa RAG untuk 1 pengguna bisnis yang melakukan 100 kueri per hari dengan LLM adalah sekitar $12,39 USD per bulan. Biaya untuk kasus penggunaan yang mendukung RAG dengan indeks Amazon Kendra yang mendukung 8.000 interaksi per hari adalah sekitar $204,26 USD per bulan, ditambah biaya basis pengetahuan.	Biaya
Memahami pertimbangan keamanan untuk solusi ini.	Keamanan
Ketahui cara merencanakan kuota untuk solusi ini.	Kuota

Jika kau mau.	Baca..
Ketahui Wilayah AWS mana yang mendukung solusi ini.	Wilayah AWS yang Didukung
Lihat atau unduh CloudFormation templat AWS yang disertakan dalam solusi ini untuk secara otomatis menerapkan sumber daya infrastru ktur (“tumpukan”) untuk solusi ini.	CloudFormation Templat AWS
Akses kode sumber dan secara opsional gunakan AWS Cloud Development Kit (AWS CDK) untuk menerapkan solusi.	GitHub repositori

Fitur dan manfaat

Solusi Generative AI Application Builder on AWS menyediakan fitur-fitur berikut:

Eksperimen cepat

Solusi ini memungkinkan pengguna untuk bereksperimen dengan cepat dengan menghapus beban berat yang diperlukan untuk menerapkan beberapa instance dengan konfigurasi yang berbeda dan membandingkan output dan kinerja. Bereksperimenlah dengan berbagai konfigurasi berbagai teknik cepat LLMs, basis pengetahuan perusahaan, pagar pembatas, agen AI, dan parameter lainnya.

Pilihan dan konfigurasi

Dengan konektor pra-bangun ke berbagai LLMs, seperti model yang tersedia melalui Amazon Bedrock, solusi ini memberi Anda fleksibilitas untuk menerapkan model pilihan Anda, serta AWS dan layanan FM terkemuka yang Anda sukai. Anda juga dapat mengaktifkan Amazon Bedrock Agents untuk memenuhi berbagai tugas dan alur kerja.

Agen Builder

Bangun dan terapkan agen AI siap produksi dengan manajemen siklus hidup penuh. Konfigurasikan prompt sistem, integrasikan server Model Context Protocol (MCP) untuk alat perusahaan dan akses data, dan aktifkan kemampuan memori untuk retensi konteks di seluruh percakapan. Agen dikerahkan di Amazon Bedrock AgentCore dengan dukungan runtime yang diperpanjang dan respons streaming waktu nyata.

Pembuat Alur Kerja

Mengatur beberapa agen Agen Builder ke dalam alur kerja yang kompleks menggunakan delegasi hierarkis. Buat agen pengawas yang secara mandiri memilih dan mengoordinasikan agen Agen Pembangun khusus untuk menangani tugas multi-langkah. Konfigurasikan deskripsi agen, strategi delegasi, dan memori tingkat alur kerja saat menggunakan kembali penerapan Agent Builder yang ada.

Siap produksi

Dibangun dengan prinsip desain AWS Well-Architected, solusi ini menawarkan keamanan dan skalabilitas tingkat perusahaan dengan ketersediaan tinggi dan latensi rendah, memastikan integrasi yang mulus ke dalam aplikasi Anda dengan standar kinerja tinggi.

Arsitektur modular yang dapat diperluas

Perluas fungsionalitas solusi ini dengan mengintegrasikan proyek Anda yang ada atau menghubungkan layanan AWS tambahan secara native. Karena ini adalah aplikasi open-source, Anda dapat menggunakan LangChain lapisan orkestrasi yang disertakan atau fungsi Lambda untuk terhubung dengan layanan pilihan Anda.

Integrasi dengan Service Catalog AppRegistry dan Application Manager, kemampuan AWS Systems Manager

Solusi ini mencakup AppRegistry sumber daya Service Catalog untuk mendaftarkan CloudFormation templat solusi dan sumber daya dasarnya sebagai aplikasi di AWS Service Catalog AppRegistry dan AWS Systems Manager Application Manager. Dengan integrasi ini, Anda dapat mengelola sumber daya solusi secara terpusat.

Kasus penggunaan Agen Pembangun vs Agen Batuan Dasar

Solusi ini menyediakan dua pendekatan berbeda untuk bekerja dengan agen AI, masing-masing cocok untuk kasus penggunaan dan persyaratan yang berbeda:

Fitur	Kasus penggunaan Agen Batuan Dasar	Agen Builder
Tujuan	Memanggil Agen Bedrock Amazon yang telah digunakan sebelumnya	Membangun, menyebarkan, dan mengelola agen kustom

Fitur	Kasus penggunaan Agen Batuan Dasar	Agen Builder
Konfigurasi	ID Agen dan ID Alias saja	Konfigurasi agen lengkap: permintaan sistem, model, server MCP, memori
Deployment	Lapisan doa sederhana	Siklus hidup agen lengkap pada Runtime AgentCore
Runtime	Layanan Agen Amazon Bedrock	Amazon Bedrock AgentCore dengan Strands SDK
Integrasi Alat	Dikonfigurasi di konsol Agen Batuan Dasar	Server Model Context Protocol (MCP) dan alat Strands bawaan
Memori	Dikelola oleh Agen Bedrock (hingga 30 hari)	AgentCore Memori dengan retensi jangka pendek dan jangka panjang yang dapat dikonfigurasi
Kustomisasi	Terbatas untuk pengaturan agen yang telah digunakan sebelumnya	Kontrol penuh atas petunjuk, model, alat, dan perilaku
Terbaik untuk	Penyebaran cepat agen yang ada	Pengembangan agen kustom dan penyebaran produksi

> **Note**
>
> Kedua opsi mendukung streaming real-time, riwayat percakapan, dan keamanan tingkat perusahaan.

Pembuat Alur Kerja

Workflow Builder memungkinkan orkestrasi multi-agen dengan membuat agen supervisor yang mendelegasikan pekerjaan ke agen Agen Builder khusus. Setiap alur kerja terdiri dari:

- Agen Supervisor: Agen entrypoint yang menerima permintaan pengguna dan mengoordinasikan agen khusus
- Agen Khusus: Agen Builder menggunakan kasus yang penyelia dapat mendelegasikan tugas
- Agen sebagai Pola Alat: Supervisor mendaftarkan setiap agen Agen Pembangun sebagai alat dan secara mandiri memilih agen mana yang akan digunakan

Fitur	Agen Builder	Pembuat Alur Kerja
Tujuan	Membangun dan menyebarkan agen kustom tunggal	Mengatur beberapa agen Agen Pembangun
Jenis Agen	Agen tunggal dengan alat MCP	Agen supervisor+beberapa agen Agen Builder
Integrasi Alat	Server MCP dan alat Strands	Agen Builder terdaftar sebagai alat
Delegasi	Pemanggilan alat langsung	Pemilihan dan delegasi agen otonom
Kompleksitas	Tugas agen tunggal	Alur kerja multi-langkah dan multi-agen
Penggunaan Kembali Agen	N/A	Menggunakan kembali penerapan Agent Builder yang ada
Terbaik untuk	Tugas domain tunggal yang terfokus	Alur kerja kompleks yang membutuhkan banyak spesialisasi

Note

- Alur kerja memerlukan setidaknya 1 kasus penggunaan Agen Builder sebagai agen agen khusus
- Semua agen khusus harus merupakan kasus penggunaan Agen Builder yang digunakan di GAAB

Kasus penggunaan

Pertanyaan menjawab atas data perusahaan

LLMs dan model dasar lainnya telah dilatih sebelumnya pada kumpulan besar data yang memungkinkan mereka berkinerja baik di banyak tugas pemrosesan bahasa alami (NLP). Tetapi sebagian besar model dasar dan LLMs statis dan telah dilatih sebelumnya, membatasi kemampuan mereka untuk menjawab pertanyaan secara akurat tentang topik yang baru, khusus, atau eksklusif. Dengan menggunakan pembelajaran berbasis prompt, Anda dapat memanfaatkan fitur NLP dan pembuatan teks yang kuat dari LLM untuk memberikan pengalaman pelanggan yang lebih kaya atas data perusahaan Anda.

Prototipe AI generatif cepat

Di luar kotak, solusinya dibundel dengan berbagai penyedia model dan kasus penggunaan. Dengan panduan penerapan yang mudah digunakan, pelanggan dapat menerapkan kasus penggunaan pra-bangun untuk memungkinkan eksperimen cepat berbagai prototipe dan beban kerja AI generatif yang berbeda.

Perbandingan dan eksperimen multi LLM

LLMs tampil berbeda, dan mengingat kebutuhan spesifik aplikasi Anda, Anda mungkin menemukan bahwa satu LLM lebih sesuai dengan aplikasi Anda daripada yang lain. Ini mungkin karena alasan yang berkaitan dengan kinerja, akurasi, biaya, kreativitas, atau banyak faktor lainnya. Solusi ini memungkinkan Anda menerapkan beberapa kasus penggunaan dengan cepat yang memungkinkan Anda bereksperimen dan membandingkan konfigurasi yang berbeda hingga Anda menemukan apa yang memenuhi kebutuhan Anda.

Konsep dan definisi

Bagian ini menjelaskan konsep-konsep kunci dan mendefinisikan terminologi khusus untuk solusi ini:

pengguna admin

Dalam konteks panduan ini, pengguna admin adalah orang yang bertanggung jawab untuk mengelola konten yang terkandung dalam penyebaran. Pengguna ini mendapatkan akses ke UI dasbor Deployment dan terutama bertanggung jawab untuk mengkurasi pengalaman pengguna bisnis. Ini adalah target pelanggan utama kami.

pengguna bisnis

Dalam konteks panduan ini, pengguna bisnis mewakili individu yang digunakan untuk kasus penggunaan. Mereka adalah konsumen dari basis pengetahuan dan pelanggan yang bertanggung jawab untuk mengevaluasi dan bereksperimen dengan. LLMs

Dasbor penyebaran

Dasbor Deployment adalah antarmuka web yang berfungsi sebagai konsol manajemen bagi pengguna admin untuk melihat, mengelola, dan membuat kasus penggunaannya. Dasbor ini memungkinkan pelanggan untuk dengan cepat bereksperimen, mengulangi, dan memproduksi berbagai AI/ML beban kerja dengan memanfaatkan. LLMs

DevOps pengguna

Dalam konteks panduan ini, DevOps pengguna adalah orang yang bertanggung jawab untuk menerapkan solusi dalam akun AWS dan untuk mengelola infrastruktur, memperbarui solusi, memantau kinerja, dan menjaga kesehatan dan siklus hidup solusi secara keseluruhan.

kasus penggunaan

Kasus penggunaan adalah aplikasi terisolasi dari solusi keseluruhan yang terintegrasi dengan LLMs untuk memungkinkan pengalaman pelanggan yang lebih kaya dengan memungkinkan penambahan antarmuka bahasa alami ke aplikasi baru atau yang sudah ada. Kasus penggunaan dapat diterapkan melalui dasbor Deployment atau sendiri.

> **Note**
>
> Untuk referensi umum istilah AWS, lihat Daftar Istilah AWS.

Gambaran umum arsitektur

Bagian ini menyediakan diagram arsitektur implementasi referensi untuk komponen yang digunakan dengan solusi ini.

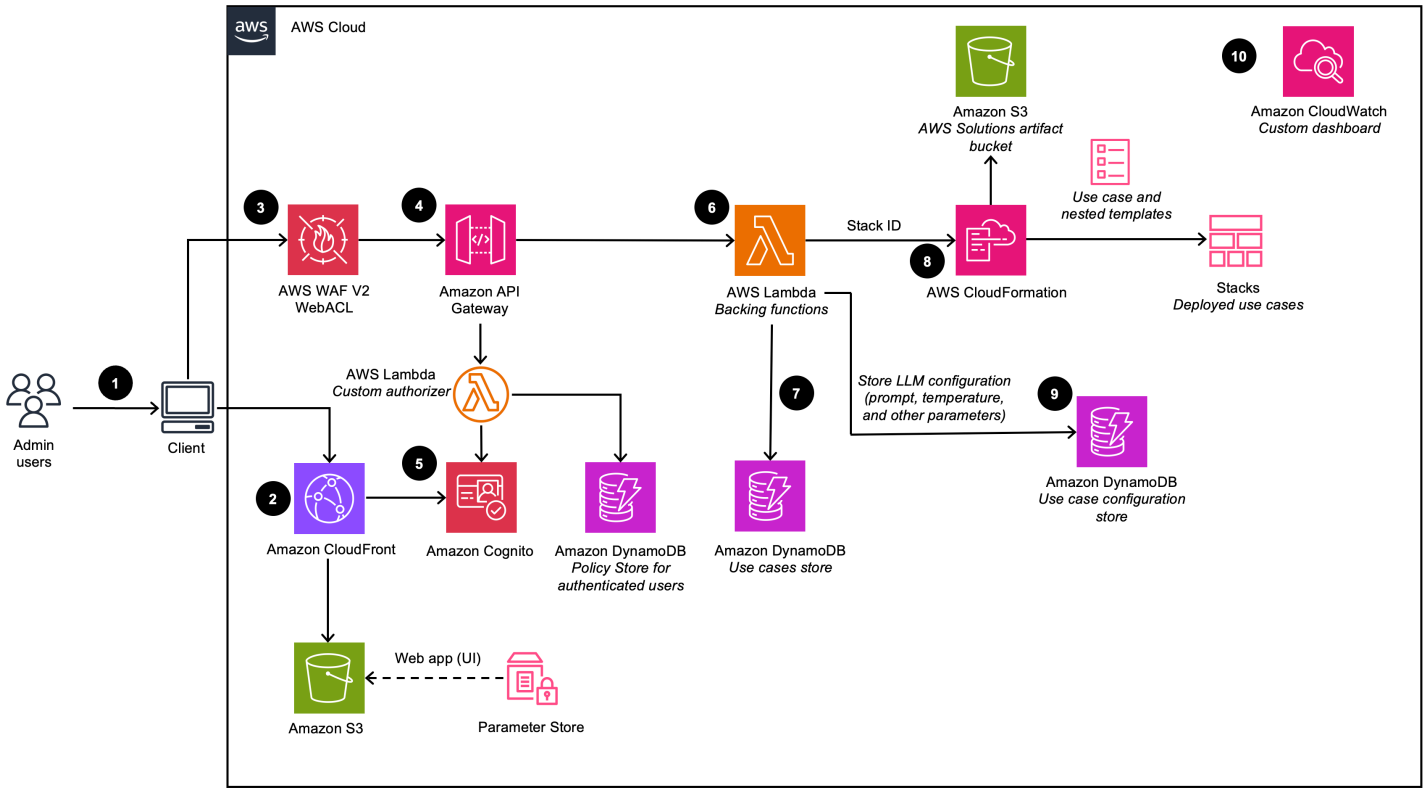
Diagram arsitektur

Untuk mendukung beberapa kasus penggunaan dan kebutuhan bisnis, solusi ini menyediakan enam CloudFormation templat AWS:

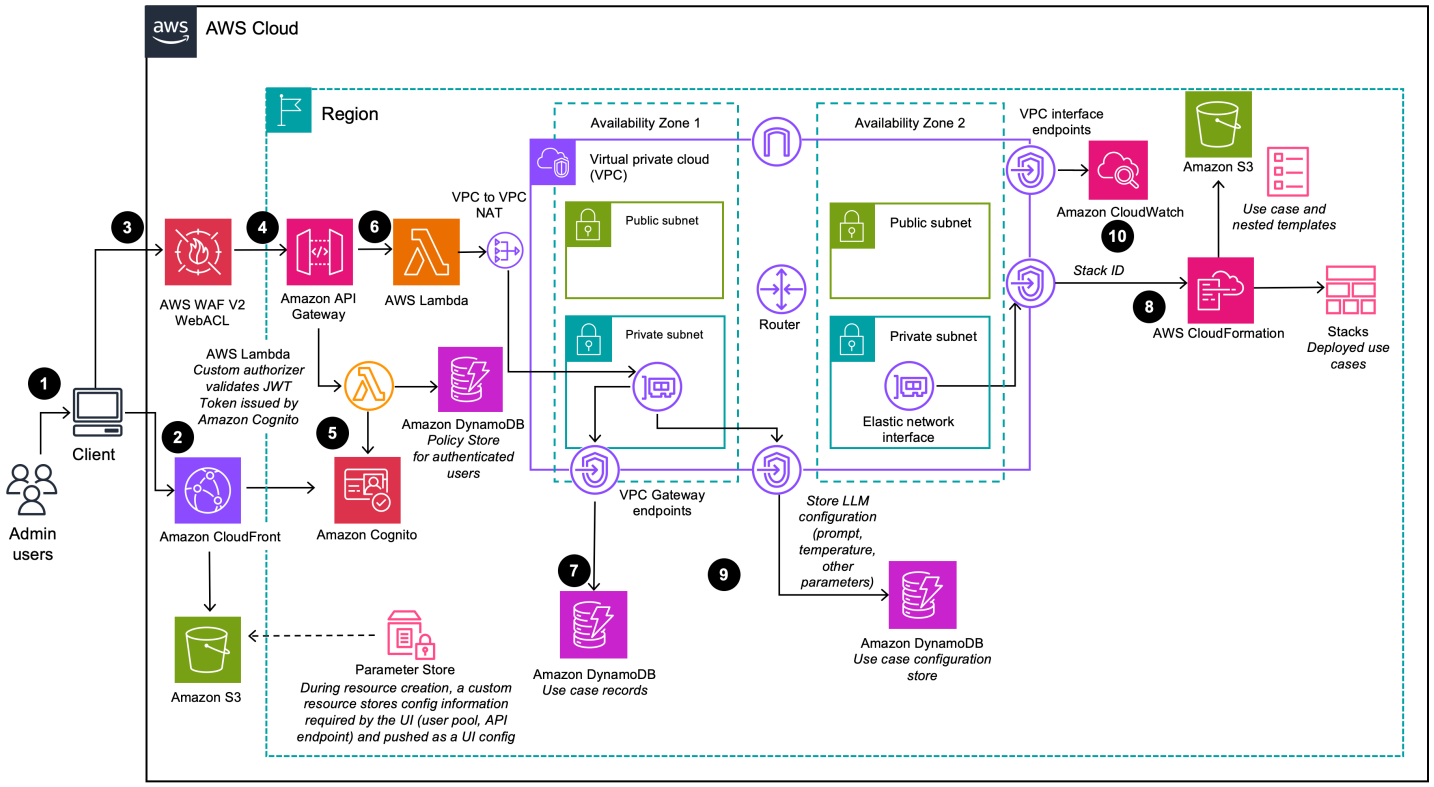
1. Dasbor Deployment - Dasbor Deployment adalah antarmuka web yang berfungsi sebagai konsol manajemen bagi pengguna admin untuk melihat, mengelola, dan membuat kasus penggunaannya. Dasbor ini memungkinkan pelanggan untuk dengan cepat bereksperimen, mengulangi, dan memproduksi berbagai AI/ML beban kerja dengan memanfaatkan. LLMs
2. Kasus penggunaan teks - Kasus penggunaan Teks memungkinkan pengguna untuk mengalami antarmuka bahasa alami menggunakan AI generatif. Kasus penggunaan ini dapat diintegrasikan ke dalam aplikasi baru atau yang sudah ada, dan dapat disebarkan melalui dasbor Deployment atau secara independen melalui URL yang disediakan.
3. Kasus penggunaan Agen Batuan Dasar - Kasus penggunaan Agen Batuan Dasar memungkinkan penggunaan Agen Batuan Dasar yang ada untuk menyelesaikan tugas atau mengotomatiskan alur kerja berulang.
4. MCP Server - Kasus penggunaan MCP Server memungkinkan penyebaran dan pengelolaan server Protokol Konteks Model yang menyediakan akses alat dan sumber daya standar ke aplikasi AI. Mendukung kedua metode gateway untuk membungkus fungsi Lambda yang ada APIs, dan server MCP eksternal, dan metode runtime untuk menyebarkan server MCP kontainer kustom.
5. Agent Builder - Agent Builder memungkinkan pembuatan dan penyebaran agen AI siap produksi di Amazon Bedrock AgentCore dengan kontrol konfigurasi penuh, integrasi server MCP, dan kemampuan manajemen memori.
6. Workflow Builder - Workflow Builder memungkinkan pembuatan agen supervisor yang mengatur beberapa agen Agen Builder menggunakan pola delegasi Agen sebagai Alat untuk alur kerja multi-agen yang kompleks.

Dasbor penyebaran

Menggambarkan arsitektur dasbor Deployment (saat digunakan dengan opsi VPC dinonaktifkan)



Menggambarkan arsitektur dasbor Deployment (saat digunakan dengan opsi VPC diaktifkan)



> **Note**
>
> CloudFormation Sumber daya AWS dibuat dari konstruksi AWS Cloud Development Kit (AWS CDK).

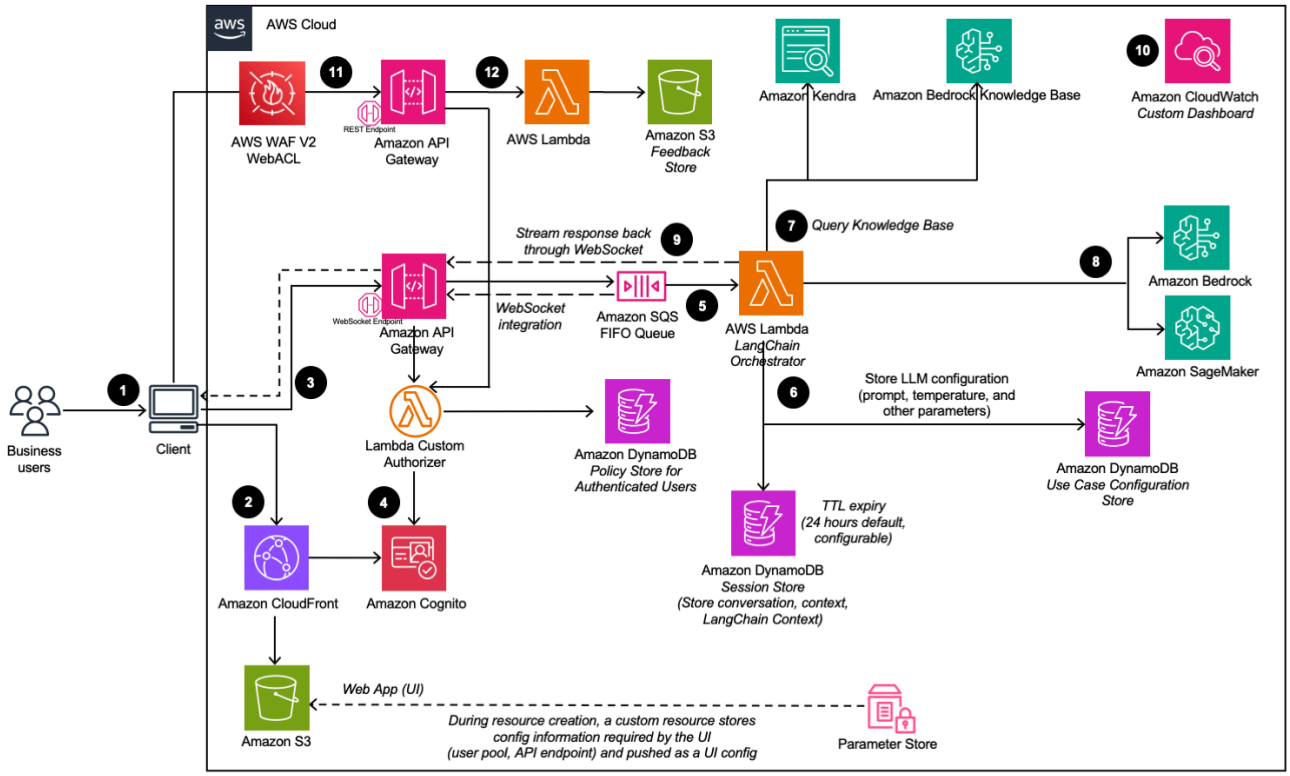
Alur proses tingkat tinggi untuk komponen solusi yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

1. Pengguna admin masuk ke antarmuka pengguna Deployment Dashboard (UI).
2. Amazon CloudFront menghadirkan UI web, yang di-host di bucket Amazon Simple Storage Service (Amazon S3).
3. AWS WAF melindungi APIs dari serangan. Solusi ini mengonfigurasi seperangkat aturan yang disebut daftar kontrol akses web (web ACL) yang memungkinkan, memblokir, atau menghitung permintaan web berdasarkan aturan dan kondisi keamanan web yang dapat dikonfigurasi dan ditentukan pengguna.
4. UI web memanfaatkan satu set REST APIs yang diekspos menggunakan Amazon API Gateway.
5. Amazon Cognito mengautentikasi pengguna dan mendukung UI CloudFront web dan API Gateway.
6. AWS Lambda menyediakan logika bisnis untuk titik akhir REST. Fungsi Lambda pendukung ini mengelola dan membuat sumber daya yang diperlukan untuk melakukan penerapan kasus penggunaan menggunakan AWS. CloudFormation
7. Amazon DynamoDB menyimpan daftar penerapan.
8. Saat kasus penggunaan baru dibuat oleh pengguna admin, fungsi Lambda yang mendukung memulai acara pembuatan tumpukan untuk kasus penggunaan CloudFormation yang diminta.
9. Semua opsi konfigurasi LLM yang disediakan oleh pengguna admin di wizard penerapan disimpan di DynamoDB. Penerapan menggunakan tabel DynamoDB ini untuk mengkonfigurasi LLM saat runtime.
10. Menggunakan Amazon CloudWatch, solusi ini mengumpulkan metrik operasional dari berbagai layanan untuk menghasilkan dasbor khusus yang memungkinkan Anda memantau kinerja solusi dan kesehatan operasional.

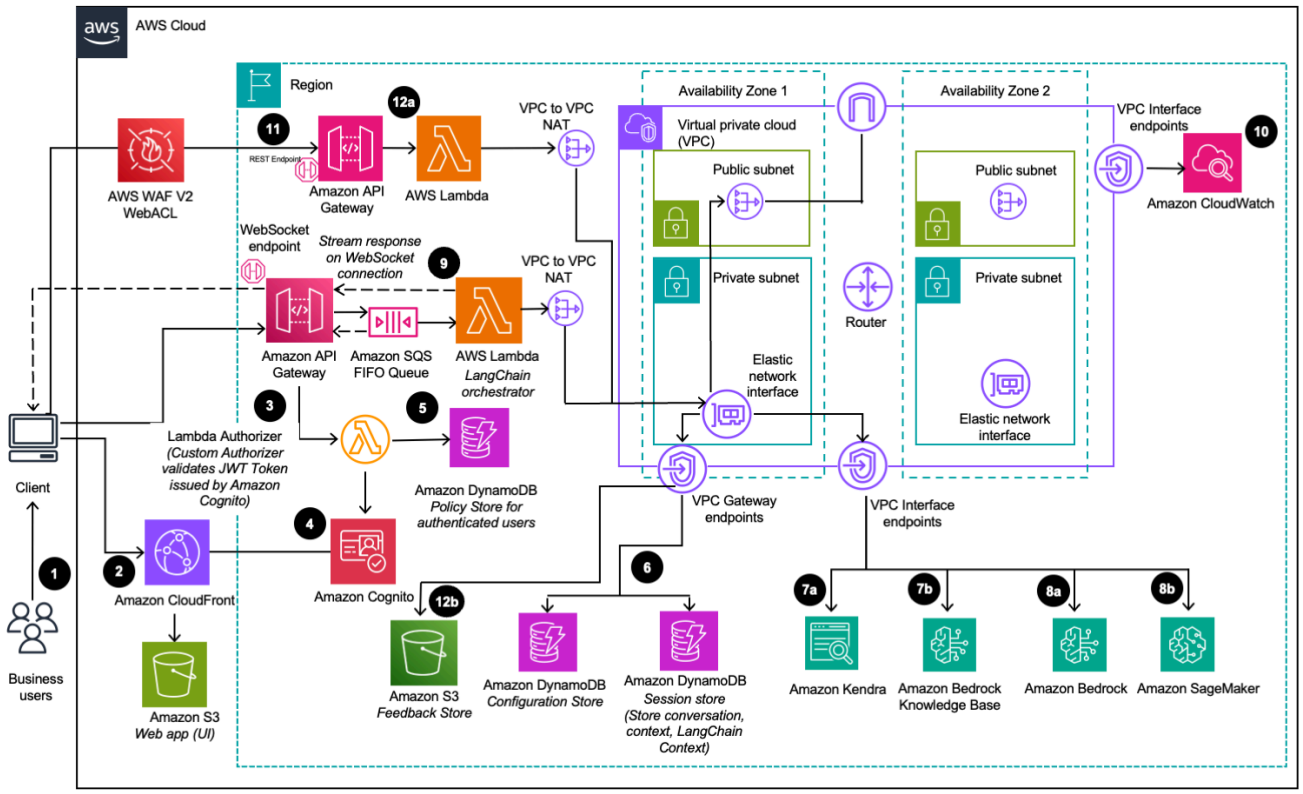
> **Note**
>
> - Jika Anda memilih untuk menerapkan solusi ini di VPC Amazon, data akan dirutekan dalam jaringan pribadi Anda.
> - Meskipun dasbor Deployment dapat diluncurkan di sebagian besar Wilayah AWS, kasus penggunaan yang diterapkan memiliki batasan tertentu berdasarkan ketersediaan layanan. Lihat Wilayah AWS yang Didukung untuk detail selengkapnya.

Kasus penggunaan teks

Menggambarkan arsitektur kasus penggunaan Teks (saat digunakan dengan opsi VPC dinonaktifkan)



Menggambarkan arsitektur kasus penggunaan Teks (saat digunakan dengan opsi VPC diaktifkan)



Alur proses tingkat tinggi untuk komponen solusi yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

1. Pengguna admin menerapkan kasus penggunaan menggunakan Dasbor Deployment. Pengguna bisnis masuk ke UI kasus penggunaan.
2. CloudFront memberikan UI web yang di-host di bucket S3.
3. UI web memanfaatkan WebSocket integrasi yang dibangun menggunakan API Gateway. API Gateway didukung oleh fungsi otorisasi Lambda khusus, yang menampilkan kebijakan AWS Identity and Access Management (IAM) yang sesuai berdasarkan grup Amazon Cognito tempat pengguna autentikasi berada. Kebijakan ini disimpan di DynamoDB.
4. Amazon Cognito mengautentikasi pengguna dan mendukung UI CloudFront web dan API Gateway.
5. Permintaan masuk dari pengguna bisnis diteruskan dari API Gateway ke antrean Amazon SQS dan kemudian ke Orchestrator. LangChain LangChain Orchestrator adalah kumpulan fungsi dan lapisan Lambda yang menyediakan logika bisnis untuk memenuhi permintaan yang berasal dari pengguna bisnis. Antrian memungkinkan operasi asinkron dari API Gateway ke integrasi Lambda. Antrian meneruskan informasi koneksi ke fungsi Lambda yang kemudian akan memposting hasil

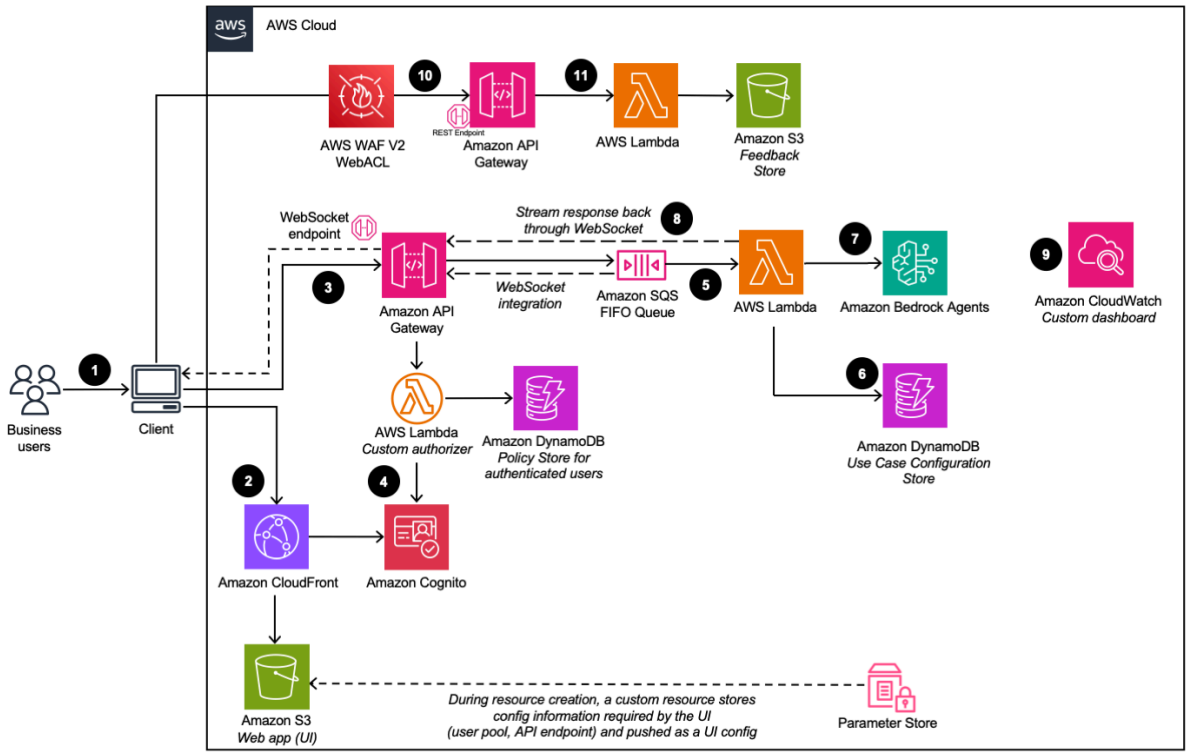
langsung kembali ke koneksi websocket API Gateway untuk mendukung panggilan inferensi yang berjalan lama.

6. LangChain Orchestrator menggunakan Amazon DynamoDB untuk mendapatkan opsi LLM yang dikonfigurasi dan informasi sesi yang diperlukan (seperti riwayat obrolan).
7. Jika penerapan memiliki basis pengetahuan yang diaktifkan, maka LangChain Orchestrator memanfaatkan Amazon Kendra atau Pangkalan Pengetahuan untuk Amazon Bedrock untuk menjalankan kueri penelusuran guna mengambil kutipan dokumen.
8. Menggunakan riwayat obrolan, kueri, dan konteks dari basis pengetahuan, LangChain Orchestrator membuat prompt terakhir dan mengirimkan permintaan ke LLM yang dihosting di Amazon Bedrock atau Amazon AI. SageMaker
9. Ketika respons kembali dari LLM, LangChain Orchestrator mengalirkan respons kembali melalui API Gateway WebSocket untuk dikonsumsi oleh aplikasi klien.
10. Menggunakan Amazon CloudWatch, solusi ini mengumpulkan metrik operasional dari berbagai layanan untuk menghasilkan dasbor khusus yang memungkinkan Anda memantau kinerja penerapan dan kesehatan operasional.
11. Jika pengumpulan umpan balik diaktifkan, titik akhir REST API, memanfaatkan Amazon API Gateway akan tersedia untuk pengumpulan umpan balik pengguna.
12. Umpan balik yang mendukung lambda, menambah umpan balik yang dikirimkan dengan metadata khusus kasus penggunaan tambahan (misalnya model yang digunakan) dan menyimpan data di Amazon S3 untuk analisis dan pelaporan selanjutnya oleh pengguna. DevOps

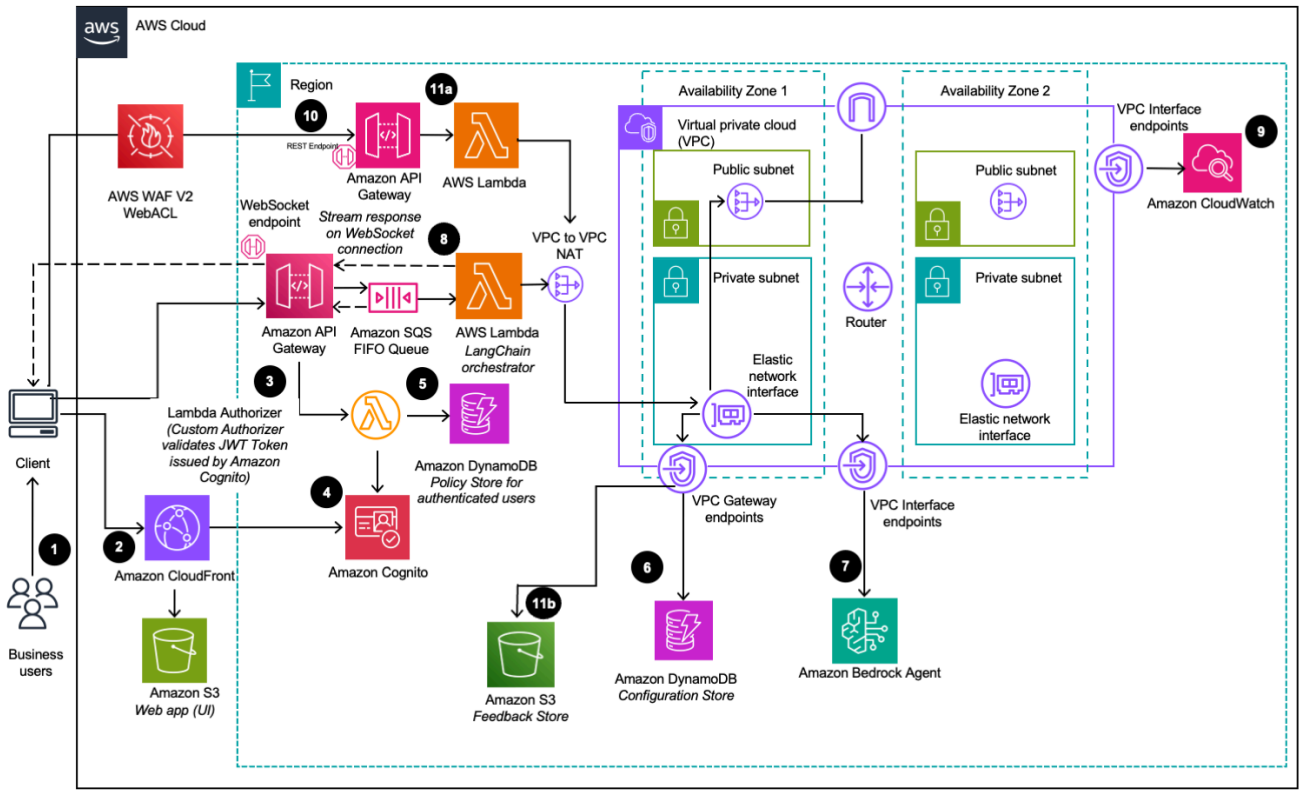
> **Note**
>
> Jika Anda memilih untuk menerapkan solusi ini di VPC Amazon, data akan diarahkan ke jaringan pribadi Anda.

Kasus penggunaan Agen Batuan Dasar

Menggambarkan arsitektur kasus penggunaan Agen Batuan Dasar (saat digunakan dengan opsi VPC dinonaktifkan)



Menggambarkan arsitektur kasus penggunaan Agen Batuan Dasar (saat digunakan dengan opsi VPC diaktifkan)



Alur proses tingkat tinggi untuk komponen solusi yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

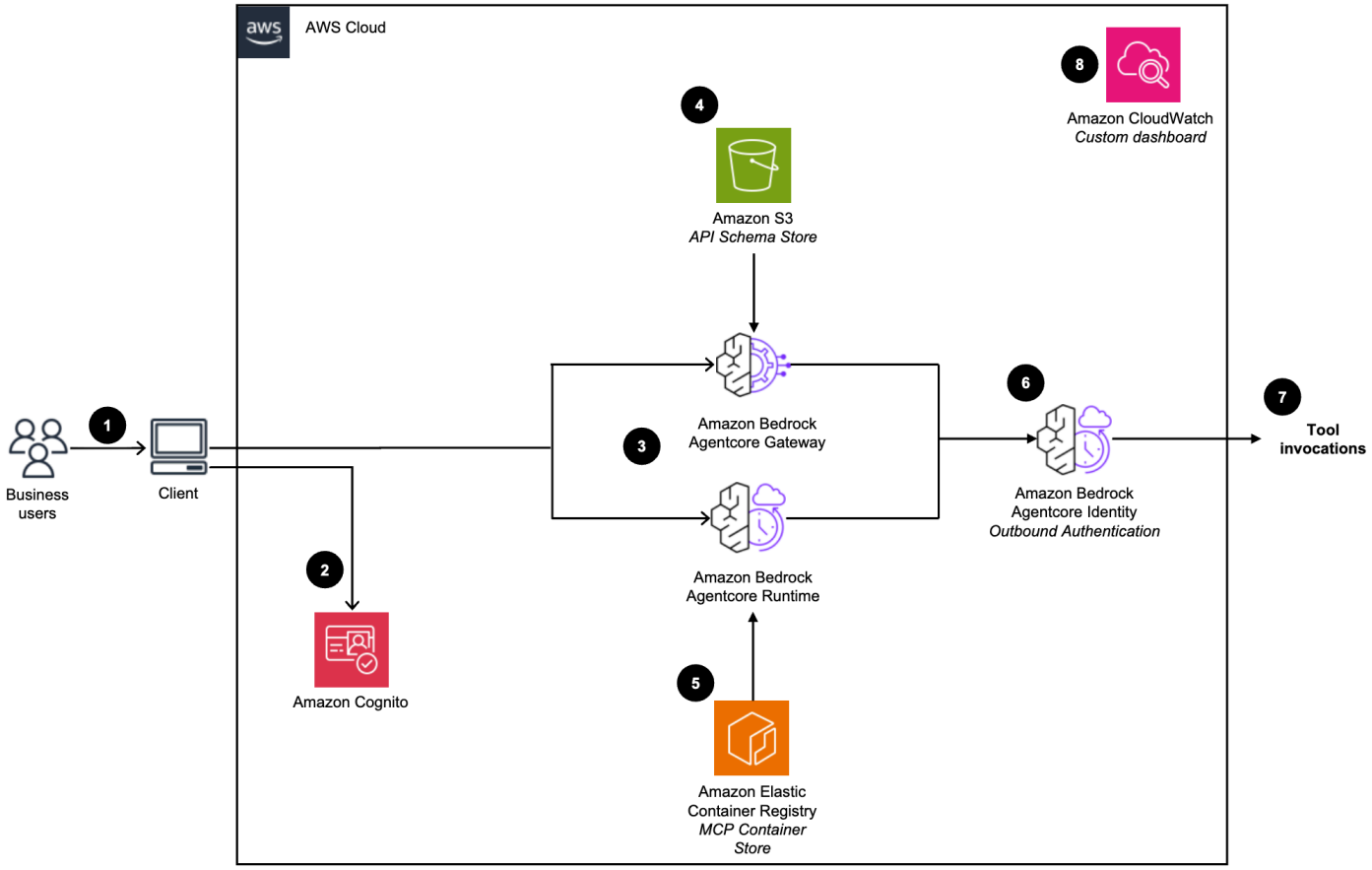
1. Pengguna admin menerapkan kasus penggunaan menggunakan Dasbor Deployment. Pengguna bisnis masuk ke UI kasus penggunaan.
2. CloudFront memberikan UI web yang di-host di bucket S3.
3. UI web memanfaatkan WebSocket integrasi yang dibangun menggunakan API Gateway. API Gateway didukung oleh fungsi otorisasi Lambda khusus, yang menampilkan kebijakan AWS Identity and Access Management (IAM) yang sesuai berdasarkan grup Amazon Cognito tempat pengguna autentikasi berada. Kebijakan ini disimpan di DynamoDB.
4. Amazon Cognito mengautentikasi pengguna dan mendukung UI CloudFront web dan API Gateway.
5. Permintaan masuk dari pengguna bisnis diteruskan dari API Gateway ke antrean Amazon SQS dan kemudian ke fungsi AWS Lambda. Antrian memungkinkan operasi asinkron dari API Gateway ke integrasi Lambda. Antrian meneruskan informasi koneksi ke fungsi Lambda yang kemudian akan memposting hasil langsung kembali ke koneksi websocket API Gateway untuk mendukung panggilan inferensi yang berjalan lama.

6. Fungsi AWS Lambda menggunakan Amazon DynamoDB untuk mendapatkan konfigurasi kasus penggunaan sesuai kebutuhan
7. Menggunakan input pengguna dan konfigurasi kasus penggunaan yang relevan, fungsi AWS Lambda membangun dan mengirimkan payload permintaan ke Agen Bedrock Amazon yang dikonfigurasi untuk memenuhi maksud pengguna.
8. Ketika respons kembali dari Amazon Bedrock Agent, fungsi Lambda mengalirkan respons kembali melalui API WebSocket Gateway untuk dikonsumsi oleh aplikasi klien.
9. Menggunakan Amazon CloudWatch, solusi ini mengumpulkan metrik operasional dari berbagai layanan untuk menghasilkan dasbor khusus yang memungkinkan Anda memantau kinerja penerapan dan kesehatan operasional.
10. Jika pengumpulan umpan balik diaktifkan, titik akhir REST API, memanfaatkan Amazon API Gateway akan tersedia untuk pengumpulan umpan balik pengguna.
11. Umpan balik yang mendukung lambda, menambah umpan balik yang dikirimkan dengan metadata khusus kasus penggunaan tambahan dan menyimpan data di Amazon S3 untuk analisis dan pelaporan selanjutnya oleh pengguna. DevOps

> **Note**
>
> Jika Anda memilih untuk menerapkan solusi ini di VPC Amazon, data akan dirutekan dalam jaringan pribadi Anda.

Kasus penggunaan MCP Server

Menggambarkan arsitektur kasus penggunaan MCP Server



Kasus penggunaan MCP Server memungkinkan penerapan dan pengelolaan server Protokol Konteks Model di Amazon Bedrock. AgentCore Server MCP menyediakan antarmuka standar untuk aplikasi AI untuk mengakses alat, sumber daya, dan sumber data perusahaan.

Solusinya mendukung dua metode penerapan:

- Metode gateway: Membungkus fungsi Lambda yang ada, APIs REST, atau server MCP eksternal sebagai alat MCP, menangani terjemahan protokol secara otomatis
- Metode runtime: Menyebarkan server MCP kontainer khusus dari gambar Amazon ECR

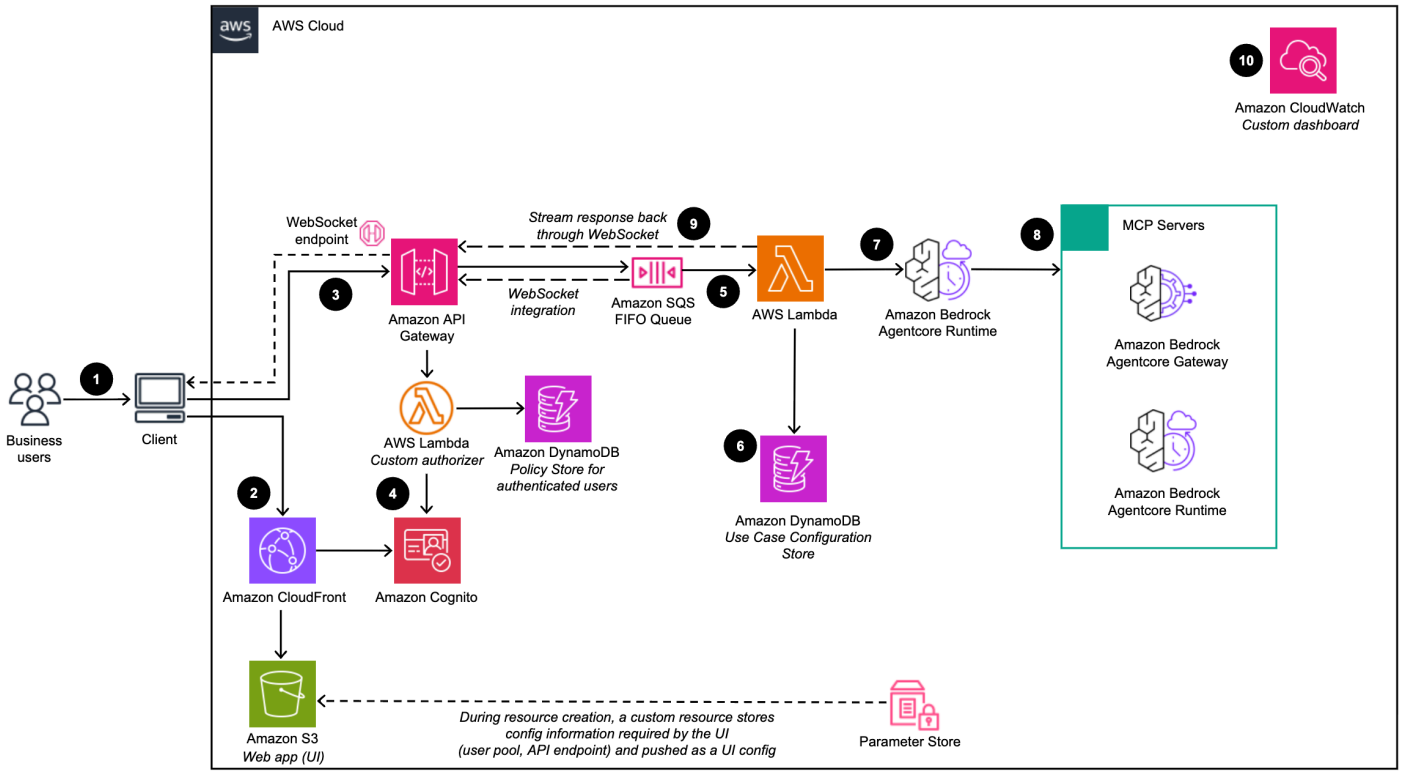
Alur proses tingkat tinggi untuk penyebaran MCP Server adalah sebagai berikut:

1. Pengguna admin menerapkan kasus penggunaan MCP Server menggunakan Dasbor Deployment, memilih metode penerapan Gateway atau Runtime.
2. Tindakan ini diautentikasi dengan Amazon Cognito.

3. Untuk penerapan Gateway, solusinya membuat Amazon Bedrock AgentCore Gateway yang mengubah fungsi Lambda yang ada APIs, atau server MCP eksternal menjadi alat yang sesuai dengan MCP. Untuk penerapan Runtime, solusinya menerapkan server MCP kontainer di Amazon Bedrock AgentCore Runtime menggunakan gambar ECR yang disediakan.
4. Penerapan gateway mengambil API/Lambda/Smithy skema yang diperlukan dari lokasi yang diunggah di Amazon S3, atau terhubung langsung ke titik akhir URL Server MCP.
5. Penerapan runtime mengambil server MCP kontainer yang disediakan oleh pengguna dari Amazon Elastic Container Registry (ECR)
6. Server MCP diinstrumentasi dengan klien Amazon Bedrock AgentCore Identity OAuth
7. MCP Server membuat alat terkait tersedia di endpoint /mcp untuk ditemukan Agen.
8. Amazon CloudWatch mengumpulkan metrik dan log operasional dari penerapan server MCP untuk pemantauan dan pemecahan masalah.

Kasus penggunaan Agent Builder

Menggambarkan arsitektur Agent Builder



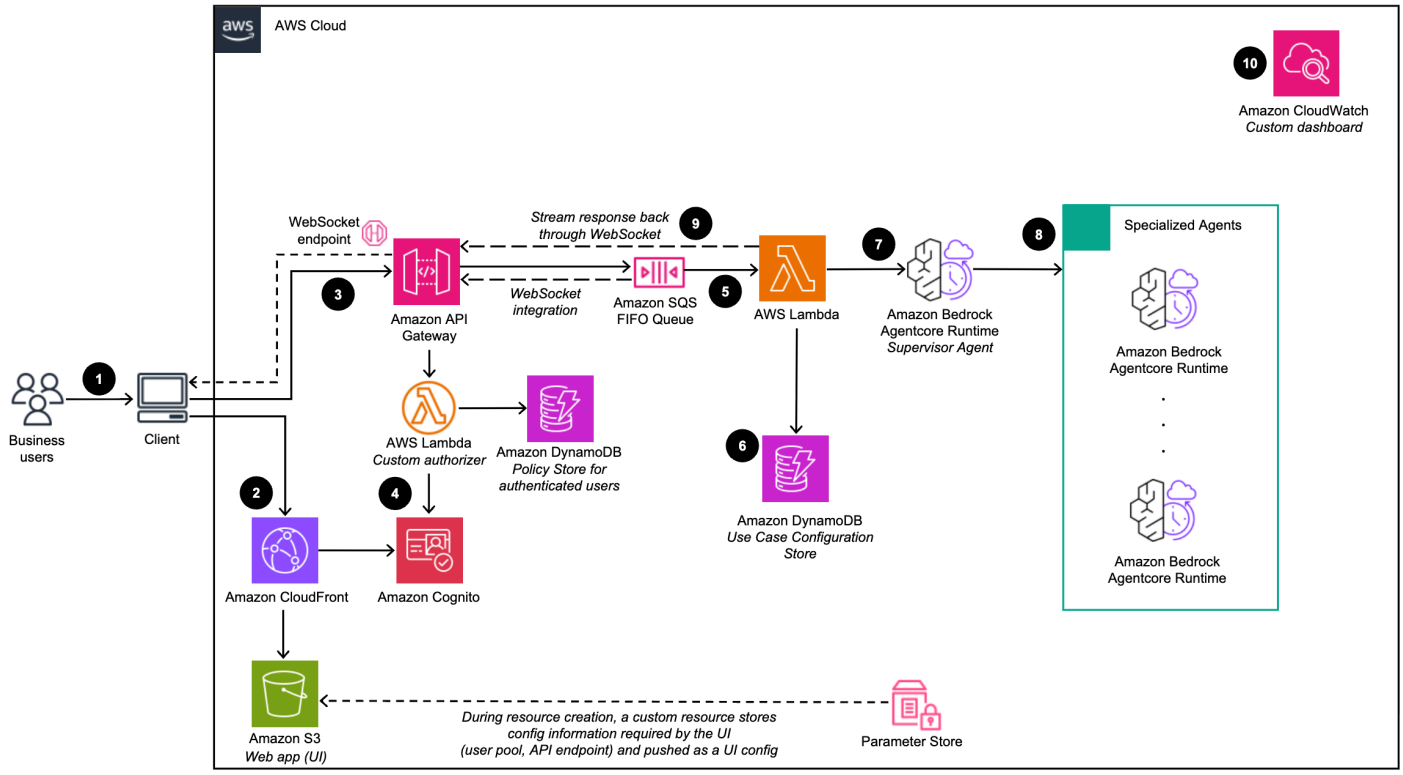
Alur proses tingkat tinggi untuk komponen Agent Builder yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

1. Pengguna admin menerapkan kasus penggunaan menggunakan Dasbor Deployment. Pengguna bisnis masuk ke UI kasus penggunaan.
2. CloudFront memberikan UI web yang di-host di bucket S3.
3. UI web memanfaatkan WebSocket integrasi yang dibangun menggunakan API Gateway. API Gateway didukung oleh fungsi otorisasi Lambda khusus, yang menampilkan kebijakan AWS Identity and Access Management (IAM) yang sesuai berdasarkan grup Amazon Cognito tempat pengguna autentikasi berada. Kebijakan ini disimpan di DynamoDB.
4. Amazon Cognito mengautentikasi pengguna dan mendukung UI CloudFront web dan API Gateway.
5. Permintaan masuk dari pengguna bisnis diteruskan dari API Gateway ke antrean Amazon SQS dan kemudian ke fungsi AWS Lambda. Antrian memungkinkan operasi asinkron dari API Gateway ke integrasi Lambda. Antrian meneruskan informasi koneksi ke fungsi Lambda yang kemudian akan memposting hasil langsung kembali ke koneksi websocket API Gateway untuk mendukung panggilan inferensi yang berjalan lama.
6. Fungsi AWS Lambda mengambil konfigurasi agen dari DynamoDB.
7. Menggunakan input pengguna dan konfigurasi kasus penggunaan yang relevan, fungsi AWS Lambda membangun dan mengirimkan payload permintaan ke agen, berjalan di Amazon Bedrock Runtime. AgentCore
8. Agen terhubung ke server MCP terkait dan mendaftarkan alat ke instance agen untaian. Agen kemudian secara mandiri memilih dan melakukan tindakan berdasarkan deskripsi alat dan persyaratan tugas.
9. Saat respons kembali dari AgentCore runtime Amazon Bedrock, fungsi Lambda mengalirkan respons kembali melalui API Gateway WebSocket untuk dikonsumsi oleh aplikasi klien.

> **Note**
>
> - Pemrosesan agen terbatas pada batas waktu eksekusi Lambda (15 menit).

Kasus penggunaan Workflow Builder

Menggambarkan arsitektur Workflow Builder



Alur proses tingkat tinggi untuk komponen Workflow Builder yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

1. Pengguna admin menerapkan alur kerja menggunakan Dasbor Deployment, memilih agen Pembuat Agen untuk disertakan sebagai agen khusus.
2. CloudFront memberikan UI web yang di-host di bucket S3.
3. UI web memanfaatkan WebSocket integrasi yang dibangun menggunakan API Gateway. API Gateway didukung oleh fungsi otorisasi Lambda khusus, yang menampilkan kebijakan AWS Identity and Access Management (IAM) yang sesuai berdasarkan grup Amazon Cognito tempat pengguna autentikasi berada. Kebijakan ini disimpan di DynamoDB.
4. Amazon Cognito mengautentikasi pengguna dan mendukung UI CloudFront web dan API Gateway.
5. Permintaan masuk dari pengguna bisnis diteruskan dari API Gateway ke antrean Amazon SQS dan kemudian ke fungsi AWS Lambda. Antrian memungkinkan operasi asinkron dari API Gateway ke integrasi Lambda.
6. Fungsi AWS Lambda mengambil konfigurasi alur kerja dari DynamoDB, termasuk daftar agen Pembuat Agen khusus.

7. Menggunakan input pengguna dan konfigurasi alur kerja, Lambda mengirimkan permintaan ke Amazon AgentCore Bedrock Runtime yang menghosting agen supervisor.
8. Agen supervisor membuat instance lokal dari semua agen Agen Builder khusus dalam lingkungan AgentCore Runtime. Agen khusus ini terdaftar sebagai alat menggunakan pola Agen sebagai Alat. Supervisor kemudian secara mandiri memilih dan mendelegasikan bekerja ke agen khusus berdasarkan deskripsi agen dan persyaratan tugas.
9. Agen supervisor mengumpulkan hasil dari agen khusus dan merumuskan respons akhir, mengembalikannya ke Lambda untuk dialirkan kembali ke aplikasi klien melalui Websocket API Gateway.

> **Note**
>
> - Pemrosesan alur kerja terbatas pada batas waktu eksekusi Lambda (15 menit).

Pertimbangan desain AWS Well-Architected

Solusi ini dirancang dengan praktik terbaik dari AWS Well-Architected Framework yang membantu pelanggan merancang dan mengoperasikan beban kerja yang andal, aman, efisien, dan hemat biaya di cloud.

Bagian ini menjelaskan bagaimana prinsip-prinsip desain dan praktik terbaik Kerangka Well-Architected diterapkan saat membangun solusi ini.

Keunggulan operasional

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keunggulan operasional.

- Kami membangun solusi seperti infrastructure-as-code menggunakan Amazon CloudFormation.
- Fungsi Lambda mendorong metrik khusus ke CloudWatch dan CloudWatch dasbor khusus untuk memantau kesehatan solusi.
- Komponen solusi sangat termodulasi, memberikan fleksibilitas untuk memilih komponen mana yang akan digunakan.

Keamanan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keamanan.

- Dasbor Deployment dan semua kasus penggunaan diautentikasi dan diotorisasi dengan Amazon Cognito.
- Semua komunikasi antar-layanan menggunakan peran AWS IAM.
- Semua peran solusi mengikuti akses hak istimewa terkecil; artinya, hanya izin minimum yang diperlukan yang diberikan.
- Semua penyimpanan data termasuk bucket S3, DynamoDB, dan Amazon Kendra memiliki enkripsi saat istirahat.

Keandalan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keandalan.

- Arsitektur berdasarkan paradigma tanpa server.
- Kami membangun arsitektur sesuai permintaan, skalabilitas horizontal, dan pemulihan otomatis dari kegagalan infrastruktur yang mendasarinya.
- Arsitekturnya mencakup permintaan buffering dan throttling agar tidak membanjiri titik akhir yang mendasarinya.

Efisiensi kinerja

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar efisiensi kinerja.

- Solusinya menggunakan DynamoDB, database NoSQL tanpa server yang dikelola sepenuhnya dengan penskalaan sesuai permintaan.
- Solusinya menggunakan Amazon S3 untuk penyimpanan objek dan meng-host situs web (melalui CloudFront) untuk memberikan biaya rendah, skalabel, dengan daya tahan 11 9s.

Optimalisasi biaya

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar pengoptimalan biaya.

- Jika memungkinkan, kami membangun solusi untuk menggunakan arsitektur tanpa server; jadi Anda hanya membayar untuk apa yang Anda gunakan.

Keberlanjutan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik pilar keberlanjutan.

- Arsitektur modular dan komponen solusi memberikan fleksibilitas untuk menyesuaikan sumber daya yang akan disediakan untuk kasus penggunaan individual.
- Arsitektur menggunakan komputasi dan penyimpanan tanpa server, yang mengoptimalkan pemanfaatan sumber daya.
- Sebagai solusi berbasis cloud, solusi ini mendapat manfaat dari sumber daya bersama, jaringan, pendinginan daya, dan fasilitas fisik.

Detail arsitektur

Bagian ini menjelaskan komponen dan layanan AWS yang membentuk solusi ini dan detail arsitektur tentang cara komponen ini bekerja sama.

Layanan AWS dalam solusi ini

AWS service	Deskripsi
Amazon API Gateway	Inti. Layanan ini menyediakan REST APIs untuk dasbor Deployment dan WebSocket API untuk kasus penggunaan.
AWS CloudFormation	Inti. Solusi ini didistribusikan sebagai CloudFormation templat, dan CloudForm ation menyebarkan sumber daya AWS untuk solusinya.
Amazon CloudFront	Inti. CloudFront menyajikan konten web yang dihosting di Amazon S3.
Amazon Cognito	Inti. Layanan ini menangani manajemen pengguna dan otentikasi untuk API.
Amazon DynamoDB	Inti. DynamoDB menyimpan informasi penyebaran dan detail konfigurasi untuk dasbor Deployment. Ini menyimpan riwayat obrolan dan percakapan IDs dalam kasus penggunaa n Teks untuk mengaktifkan riwayat percakapan dan disambiguasi kueri.
AWS Lambda	Inti. Solusinya menggunakan fungsi Lambda untuk: * Kembali titik akhir REST dan WebSocket API * Menangani logika inti dari setiap orkestrator

AWS service	Deskripsi
	kasus penggunaan * Menerapkan sumber daya khusus selama penerapan CloudFormation
Amazon S3	Inti. Amazon S3 meng-host konten web statis.
Amazon CloudWatch	Mendukung. Solusi ini menerbitkan log dari sumber daya solusi ke CloudWatch Log, dan menerbitkan metrik ke metrik. CloudWatch Solusinya juga membuat CloudWatch dasbor untuk melihat data ini.
AWS Systems Manager	Mendukung. Systems Manager menyediak an pemantauan sumber daya tingkat aplikasi dan visualisasi operasi sumber daya dan data biaya. Juga digunakan untuk menyimpan data konfigurasi di Parameter Store.
AWS WAF	Mendukung. AWS WAF diterapkan di depan penerapan API Gateway untuk melindunginya.
Amazon Bedrock	Opsional. Solusi ini memanfaatkan Amazon Bedrock untuk mengakses fondasi atau model yang disesuaikan, Amazon Bedrock Agents, Amazon Bedrock Knowledge Bases. Amazon Bedrock adalah integrasi yang disarankan untuk menjaga agar data Anda tidak keluar dari jaringan AWS.
Amazon Bedrock AgentCore	Opsional Solusinya memanfaatkan Amazon Bedrock AgentCore untuk menjalankan dan mendukung koneksi MCP Server serta Kasus Penggunaan Pembuat Agen dan Alur Kerja.

AWS service	Deskripsi
Amazon Elastic Container Registry (Amazon ECR)	Opsional. Untuk penerapan Agent Builder, ECR menyimpan dan mendistribusikan gambar kontainer agen. Solusinya menggunakan ECR Pull-Through Cache untuk secara otomatis mengambil gambar agen pra-bangun dari repositori ECR publik tim GAAB.
AWS Distro untuk OpenTelemetry (ADOT)	Opsional. Untuk penerapan Agent Builder, ADOT menyediakan instrumentasi otomatis untuk observabilitas agen, memungkinkan penelusuran terdistribusi dan pencatatan terstruktur untuk operasi agen.
Amazon Kendra	Opsional. Dalam kasus penggunaan Teks, pengguna admin secara opsional dapat memutuskan untuk menghubungkan indeks Amazon Kendra untuk digunakan sebagai basis pengetahuan untuk percakapan dengan LLM. Ini dapat digunakan untuk menyuntik kan informasi baru ke LLM memberikan kemampuan untuk menggunakan informasi itu dalam tanggapannya.
Amazon SageMaker AI	Opsional. Solusinya dapat diintegrasikan dengan titik akhir inferensi Amazon SageMaker AI untuk mengakses FMs yang dihosting dalam akun AWS dan Wilayah Anda dan merupakan integrasi pilihan untuk menjaga data Anda agar tidak keluar dari jaringan AWS. **Note** Anda harus menerapkan solusi di Wilayah yang sama di mana titik akhir inferensi tersedia.

AWS service	Deskripsi
Amazon Virtual Private Cloud	Opsional. Solusinya menyediakan opsi untuk menyebarkan komponen dengan konfigura si berkemampuan VPC. Saat menerapkan solusi dengan konfigurasi berkemampuan VPC, Anda memiliki opsi untuk membiarkan solusi membuat VPC untuk Anda, atau menggunak an VPC yang ada di akun dan Wilayah yang sama tempat solusi akan digunakan (Bawa VPC Anda Sendiri). Jika solusi menciptak an VPC, itu menciptakan komponen jaringan yang diperlukan yang mencakup, subnet, grup keamanan dan aturannya, tabel rute, jaringan, Gateway NAT, Gateway Internet ACLs, titik akhir VPC, dan kebijakannya.

Dasbor penyebaran

Otorisasi kustom API Gateway

Di bawah permukaan, otorisasi kustom Lambda untuk API Gateway digunakan untuk semua panggilan API (RESTful baik WebSocket dan berbasis) untuk memvalidasi jika pengguna tertentu memiliki izin untuk melakukan tindakan berdasarkan grup tempat mereka berada. Authorizer kustom ini didukung oleh tabel DynamoDB yang berisi kebijakan untuk setiap grup. Saat memanggil API, API Gateway memanggil fungsi Lambda otorisasi kustom, yang menerjemahkan token akses Amazon Cognito yang disediakan untuk menentukan grup pengguna mana yang dimiliki pengguna. Tabel kebijakan kemudian ditanyakan berdasarkan nama grup untuk menampilkan kebijakan yang relevan untuk grup tersebut.

Pada setiap penerapan kasus penggunaan baru, kebijakan admin diperbarui untuk menyimpan pernyataan baru yang memungkinkan tindakan execute-api:invoke pada API kasus penggunaan tersebut. Ketika kasus penggunaan dihapus, pernyataan yang sesuai dihapus dari kebijakan.

Untuk grup yang dibuat untuk kasus penggunaan individual, hanya satu pernyataan yang ada dalam kebijakan, yang memungkinkan tindakan execute-API:invoke hanya pada API kasus penggunaan tersebut.

Karena struktur ini, setiap pengguna yang termasuk dalam grup kasus penggunaan dapat mengakses API kasus penggunaan tersebut. Satu pengguna juga dapat ditambahkan secara manual ke beberapa grup untuk memungkinkan pengguna tersebut menggunakan beberapa kasus penggunaan.

> **Warning**
>
> Anda juga dapat mengedit kebijakan untuk grup tertentu secara manual dalam tabel kebijakan jika Anda ingin memberikan akses ke kasus penggunaan baru ke grup pengguna yang sudah ada. Grup kasus penggunaan dihapus ketika kasus penggunaan dihapus (bahkan jika Anda telah melakukan pengeditan manual), jadi lanjutkan dengan hati-hati saat menghapus kasus penggunaan.

Dalam kasus di mana tumpukan kasus penggunaan diterapkan secara mandiri (tanpa menggunakan dasbor Deployment), kumpulan pengguna Amazon Cognito dibuat untuk penerapan yang berisi satu pengguna dengan akses ke API kasus penggunaan tersebut. Kumpulan pengguna ini hanya milik kasus penggunaan ini dan tidak dibagikan di seluruh penerapan mandiri lainnya.

Kasus penggunaan teks

Dukungan streaming

Dalam aplikasi obrolan, latensi adalah metrik penting untuk memungkinkan pengalaman pengguna yang responsif. Potensi kesimpulan LLM untuk mengambil dari detik ke menit, memberikan tantangan dalam cara terbaik untuk melayani konten kepada pelanggan. Untuk alasan ini, beberapa penyedia LLM mengizinkan respons streaming kembali ke penelepon. Alih-alih menunggu seluruh inferensi selesai sebelum mengembalikan respons, setiap token dapat dikembalikan ketika tersedia.

Untuk mendukung penggunaan fitur ini, kasus penggunaan Teks telah dirancang untuk menggunakan WebSocket API untuk mendukung pengalaman obrolan. Ini WebSocket diterapkan melalui API Gateway. Penggunaan WebSocket API memungkinkan koneksi dibuat di awal sesi obrolan dan agar respons dialirkan melalui soket itu. Hal ini memungkinkan aplikasi frontend untuk memberikan pengalaman pengguna yang lebih baik.

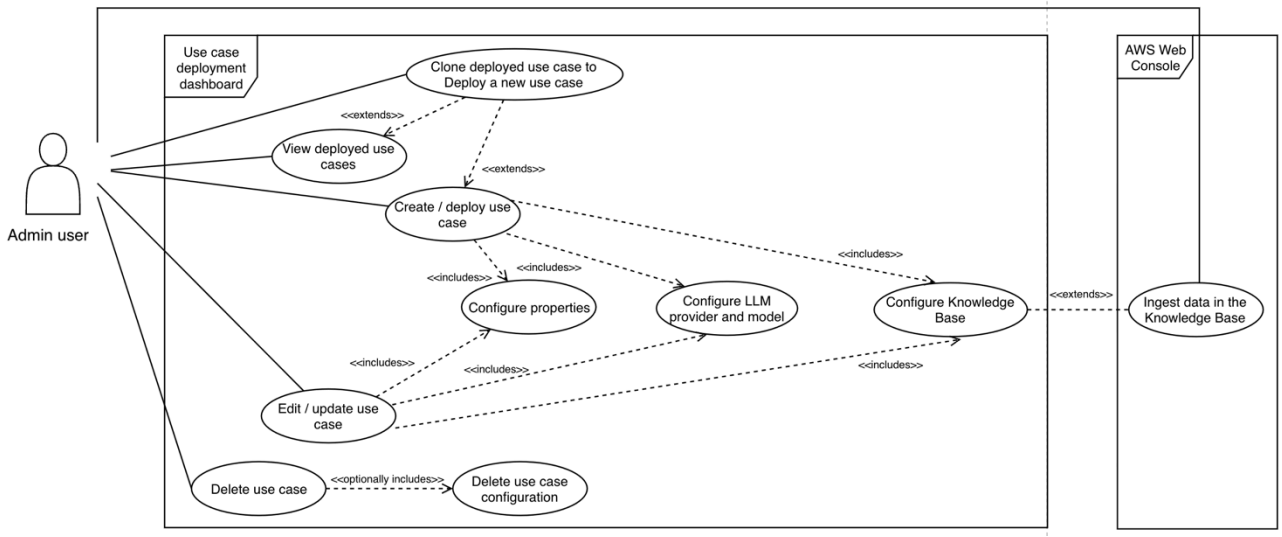
> **Note**
>
> Bahkan jika model menyediakan dukungan streaming, ini tidak berarti bahwa solusi tersebut akan dapat mengalirkan respons kembali melalui WebSocket API. Ada kebutuhan akan solusi untuk mengaktifkan logika khusus untuk mendukung streaming untuk setiap penyedia model. Jika streaming tersedia, pengguna admin akan dapat menggunakan fitur enable/disable ini pada waktu penerapan.

Cara kerja Generative AI Application Builder pada solusi AWS

Pengguna admin terutama berinteraksi dengan dasbor Deployment untuk melihat, membuat, dan mengelola penerapan kasus penggunaan baru dan yang sudah ada. Melalui dasbor ini, pengguna admin memiliki akses ke tindakan berikut:

- Lihat daftar penerapan
- Buat penerapan baru
- Edit penerapan yang ada
- Mengkloning konfigurasi penerapan untuk membuat penerapan baru
- Menghapus penerapan (menghentikan penyediaan sumber daya melalui penghapusan) CloudFormation
- Hapus detail konfigurasi penerapan secara permanen

Menggambarkan diagram Use case untuk pengguna admin dasbor Deployment

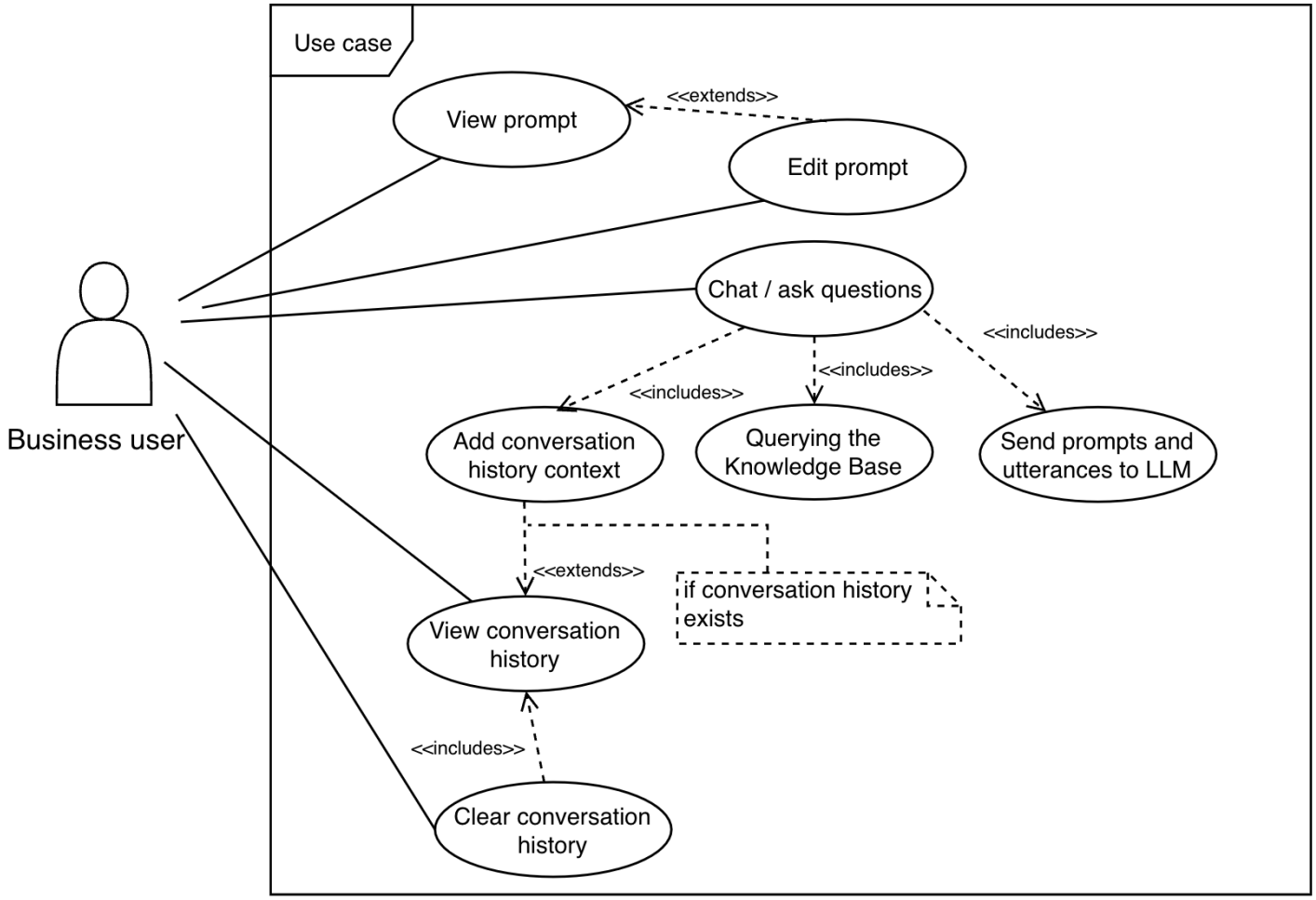


> **Note**
>
> Pengguna admin mungkin tidak memiliki akses langsung ke konsol AWS. Dalam hal ini, pengguna admin harus bekerja dengan DevOps pengguna untuk mendukung tindakan seperti menelan data ke dalam basis pengetahuan Kendra.

Untuk kasus penggunaan Teks, pengguna bisnis mendapatkan akses ke antarmuka pengguna yang memungkinkan mereka untuk mengobrol dengan LLM. Spesifikasi konfigurasi ini dikendalikan oleh pengaturan penerapan yang dikonfigurasi oleh pengguna admin. Dalam kasus penggunaan Teks, pengguna bisnis memiliki akses ke tindakan berikut:

- Kirim pesan melalui antarmuka obrolan
- Lihat riwayat percakapan
- Hapus riwayat percakapan
- Lihat prompt
- Edit prompt

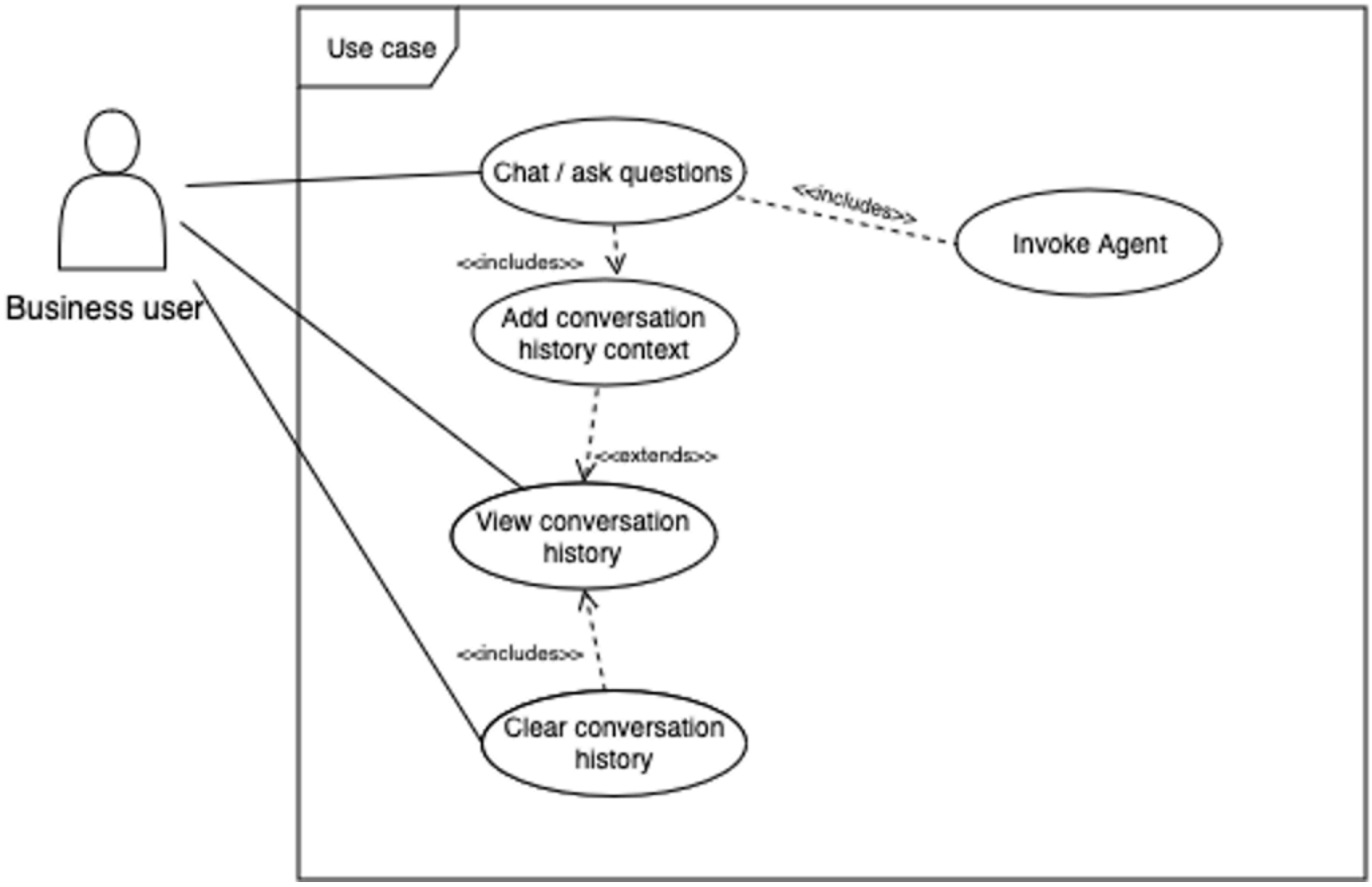
Menggambarkan Diagram kasus penggunaan untuk pengguna bisnis kasus penggunaan Teks



Dengan kasus penggunaan Agen Bedrock, pengguna bisnis dapat mengakses UI untuk mengobrol dengan Agen Amazon Bedrock yang dikonfigurasi. Pengguna admin dapat mengonfigurasi spesifikasi ini dalam pengaturan penerapan. Dalam kasus penggunaan Agen Bedrock, pengguna bisnis memiliki akses ke tindakan berikut:

- Kirim pesan melalui antarmuka obrolan
- Lihat riwayat percakapan
- Hapus riwayat percakapan

Menggambarkan Diagram kasus penggunaan untuk pengguna bisnis kasus penggunaan Agen Batuan Dasar



Agen Builder

Agen Builder menyediakan platform untuk membuat, menyebarkan, dan mengelola agen AI siap produksi di Amazon Bedrock. AgentCore Bagian ini menjelaskan komponen teknis dan detail implementasi.

AgentCore integrasi

Agent Builder menggunakan pendekatan penerapan berbasis konfigurasi dengan image agen yang dibuat sebelumnya untuk memungkinkan penerapan agen yang cepat, aman, dan dapat diskalakan.

Gambar agen pra-dibangun

Gambar kontainer agen dibuat oleh tim GAAB selama CI/CD pipeline dan dipublikasikan ke repositori ECR publik. Setiap versi gambar terkait dengan versi solusi GAAB (misalnya, v4.0.0 →:v4.0.0). gaab-strands-agent Gambar didasarkan pada Strands SDK dan termasuk:

- Lingkungan runtime agen
- Integrasi klien MCP
- Kemampuan manajemen memori
- OpenTelemetry instrumentasi

Cache Pull-Through ECR

Solusinya menggunakan ECR Pull-Through Cache untuk secara otomatis mendistribusikan gambar agen dari repositori ECR publik ke ECR pribadi pelanggan. Layanan yang dikelola AWS ini:

- Cache gambar pada tarikan pertama (penundaan 2-5 menit)
- Menghilangkan logika penyalinan gambar kustom
- Menyediakan ketersediaan gambar lokal untuk penerapan berikutnya
- Membuat aturan cache unik per penerapan untuk menghindari konflik

Penyimpanan konfigurasi

Konfigurasi agen disimpan di DynamoDB bersama konfigurasi kasus penggunaan yang ada. Setiap konfigurasi meliputi:

- Templat prompt sistem
- Penyedia model dan ID model
- Parameter model (suhu, max_tokens)
- Referensi dan titik akhir server MCP
- Pengaturan memori (toggle memori jangka panjang)
- Metadata penyebaran

Registri versi gambar

Tabel DynamoDB melacak versi gambar agen yang tersedia dan URIs cache mereka, memungkinkan manajemen versi dan kompatibilitas mundur.

Konfigurasi agen

Permintaan sistem

Permintaan sistem mendefinisikan perilaku agen, kepribadian, dan kemampuan. Pengguna admin dapat:

- Mengedit template default melalui UI Agent Builder
- Sertakan instruksi untuk penggunaan alat dan pemformatan respons
- Setel ulang ke templat default kapan saja

Pemilihan model

Agent Builder mendukung model Amazon Bedrock di v4.0.0:

- Penyedia model: Amazon Bedrock (hanya opsi di v4.0.0)
- Pemilihan model: Claude, Nova, dan model Bedrock lainnya
- Parameter model: Suhu, max_tokens, top_p, dan pengaturan khusus model

Integrasi server MCP

Server Protokol Konteks Model memberi agen akses ke alat dan data perusahaan:

- Penemuan server melalui titik akhir GET /mcp API
- Konfigurasi dinamis tanpa perubahan kode
- Otentikasi dan manajemen titik akhir
- Kemampuan alat paparan agen

Streaming dan pemrosesan

Streaming waktu nyata

Agent Builder menggunakan Server-Sent Events (SSE) dari AgentCore dijembatani hingga WebSocket streaming respons waktu nyata:

- Fungsi Lambda membuat koneksi SSE ke Runtime AgentCore
- Stream dijembatani ke API Gateway WebSocket

- Memungkinkan pengiriman token-by-token respons ke klien
- Menjaga koneksi untuk permintaan yang berjalan lama

Kendala pemrosesan

Pemrosesan agen di v4.0.0 terbatas pada batas waktu eksekusi Lambda:

- Waktu pemrosesan maksimum: 15 menit
- Model pemrosesan sinkron
- Cocokkan untuk agen percakapan dan alur kerja moderat
- Dukungan asinkron yang diperluas direncanakan untuk v4.1 +

Manajemen memori

Memori jangka pendek

Diaktifkan secara default untuk semua agen yang menggunakan kustom MemoryHookProvider:

- Menangkap peristiwa percakapan melalui handler callback Strands
- Mengatur berdasarkan ActorID dan SessionID untuk isolasi konteks
- Mempertahankan konteks percakapan dalam sesi
- Integrasi otomatis dengan AgentCore Memori

Memori jangka panjang

Fitur opsional menggunakan AgentCore Memory Tool dari strands_tools:

- Beralih sederhana di UI Agent Builder
- Strategi memori semantik dengan pengaturan default
- Akses yang dikendalikan agen melalui pemanggilan alat alami
- Menyimpan wawasan yang diekstraksi di seluruh sesi
- Menggunakan ConversationId sebagai SessionId

Observabilitas

AWS OpenTelemetry Distro (ADOT)

Agen secara otomatis diinstrumentasi selama pembuatan kontainer:

- Pembuatan jejak otomatis untuk operasi agen
- Penelusuran terdistribusi melintasi batas layanan
- Pencatatan terstruktur dengan korelasi IDs
- Integrasi dengan Penelusuran CloudWatch Transaksi

Aliran otentikasi

Pengguna mengautentikasi melalui Amazon Cognito dengan token JWT yang divalidasi oleh otorisasi Lambda khusus yang mengambil kebijakan IAM dari DynamoDB berdasarkan grup pengguna.

Pembuat Alur Kerja

Workflow Builder memungkinkan orkestrasi multi-agen dengan membuat agen supervisor yang mengoordinasikan beberapa agen Pembuat Agen menggunakan pola delegasi Agen sebagai Alat.

Arsitektur alur kerja

Komponen kunci

- Agen Supervisor: Agen Entrypoint yang menerima permintaan pengguna dan delegasi ke agen khusus
- Agen Khusus: Kasus penggunaan Agen Builder terdaftar sebagai alat untuk supervisor
- Agent Registry: tabel DynamoDB menyimpan konfigurasi agen dan metadata
- Lapisan Orkestrasi: Implementasi SDK Strands of Agents as Tools pattern

Instantiasi agen

Pembuatan agen lokal

Semua agen khusus dipakai secara lokal dalam Runtime yang sama: AgentCore

1. Mengambil konfigurasi agen dari DynamoDB
2. Membuat instance lokal dari setiap agen Agen Builder
3. Setiap agen mempertahankan koneksi server MCP sendiri
4. Agen pengawas mendaftarkan agen khusus sebagai alat

5. Strands SDK mengelola pemilihan dan delegasi agen

Rencanakan penyebaran Anda

Bagian ini menjelaskan pertimbangan biaya, keamanan, Wilayah, dan kuota untuk merencanakan penyebaran Anda.

> **Important**
>
> Solusi ini memanfaatkan Amazon Bedrock sebagai layanan utama untuk mengakses model yang dihasilkan AI. Anda harus terlebih dahulu meminta akses ke model sebelum tersedia untuk digunakan dalam solusi. Untuk detailnya, lihat Akses model di Panduan Pengguna Amazon Bedrock.

Wilayah AWS yang Didukung

> **Important**
>
> Solusi ini secara opsional menggunakan layanan Amazon Bedrock dan Amazon Kendra, yang saat ini tidak tersedia di semua Wilayah AWS. Anda harus meluncurkan solusi ini di Wilayah AWS tempat layanan ini tersedia. Untuk ketersediaan terbaru layanan AWS menurut Wilayah, lihat Daftar Layanan Regional AWS.

Pembuat Aplikasi AI Generatif di AWS didukung di Wilayah AWS berikut:

Nama wilayah	
AS Timur (Ohio)	(Canada (Central)
US East (N. Virginia)	Eropa (Frankfurt)
AS Barat (California Utara)	Eropa (Irlandia)
AS Barat (Oregon)	Eropa (London)
Asia Pasifik (Mumbai)	Europe (Milan)
Asia Pasifik (Seoul)	Eropa (Paris)

Nama wilayah	
Asia Pasifik (Singapura)	Eropa (Stockholm)
Asia Pasifik (Sydney)	Timur Tengah (Bahrain)
Asia Pasifik (Tokyo)	Amerika Selatan (Sao Paulo)

> **Note**
>
> Jika menggunakan model dasar yang diakses di luar AWS dalam penerapan Anda, tanyakan kepada penyedia model di Wilayah mana mereka APIs tersedia. Jika mereka hanya APIs tersedia di Wilayah tertentu, Anda mungkin mengalami ketidakstabilan dalam bentuk latensi tinggi atau bahkan time out. Penting juga untuk memeriksa dengan tim hukum dan kepatuhan organisasi Anda untuk mengevaluasi pertimbangan data yang melintasi batas-batas regional.

Biaya

Dengan AWS Solution ini, Anda hanya membayar sumber daya yang Anda gunakan dan tidak ada biaya minimum atau biaya penyiapan. Pengguna membayar dasbor yang digunakan untuk meluncurkan kasus penggunaan AI Generatif dan, dan untuk kasus penggunaan apa pun yang digunakan. Biaya kasus penggunaan yang diterapkan tergantung pada konfigurasi. Contoh konfigurasi:

1. Dasbor Deployment sederhana yang harganya sekitar $20 USD per bulan.
2. Kasus penggunaan chatbot siap produksi sederhana yang digunakan dengan pengaturan default yang berjalan di US East (Virginia N.), didukung oleh Amazon Bedrock tanpa akses ke dokumen, yang juga berharga sekitar $200 USD per bulan.
3. Sistem skala dalam kasus penggunaan VPC Amazon yang mendukung 8.000 kueri per hari selama puluhan ribu dokumen, yang biayanya sekitar $1.500 USD per bulan. Biaya kasus penggunaan akan bervariasi tergantung pada konfigurasi, seperti kasus penggunaan Teks dengan penyedia model yang berbeda, dengan atau tanpa Retrieval Augmented Generation (RAG) diaktifkan, dan sebagainya.

Deskripsi beban kerja	Perkiraan biaya (USD/bulan)
Biaya sampel untuk dasbor Deployment	$20/bulan
Biaya sampel untuk bukti konsep berbasis teks (termasuk dasbor Deployment dan 1 kasus penggunaan Teks, ~ 100 interaksi per hari)	$40/bulan
Biaya sampel untuk mesin kueri AI generatif yang sangat skalabel (Termasuk dasbor Deployment, 1 kasus penggunaan Teks, dan Indeks Amazon Kendra untuk RAG hingga 100K dokumen dengan ~8K kueri per hari, dengan VPC diaktifkan	$1,500/bulan
Biaya sampel untuk bukti konsep berbasis agen (Termasuk dasbor Deployment, 1 kasus penggunaan Agen Batuan Dasar dengan Pangkalan Pengetahuan Amazon Bedrock dan Amazon Bedrock Guardrails diaktifkan, ~100 interaksi per hari)	$840/bulan
Biaya sampel untuk MCP Server (Termasuk dasbor Deployment, 1 kasus penggunaan Server MCP dengan metode Gateway untuk integrasi Lambda, ~ 100 pemanggilan alat per hari)	$22/bulan
Biaya sampel untuk Agent Builder (Termasuk dasbor Deployment, 1 kasus penggunaan Agent Builder dengan integrasi MCP dan memori jangka panjang diaktifkan, ~ 100 interaksi per hari)	$55/bulan
Biaya sampel untuk Workflow Builder	$109/bulan

Deskripsi beban kerja	Perkiraan biaya (USD/bulan)
(Termasuk dasbor Deployment, 1 Workflow dengan 3 agen Agent Builder, ~ 100 interaksi per hari)	

Important

Contoh-contoh ini hanya dimaksudkan untuk membantu Anda memperkirakan biaya untuk beban kerja spesifik Anda. Penggunaan berbagai konfigurasi LLMs, atau layanan AWS dapat mengubah biaya Anda (misalnya, serverless/on-demand billing vs. provisioned/time -billed). Untuk mengelola biaya, sebaiknya buat anggaran melalui AWS Cost Explorer. Harga dapat berubah sewaktu-waktu. Untuk detail selengkapnya, lihat halaman web harga untuk setiap layanan AWS yang digunakan dalam solusi ini.

Contoh biaya untuk menjalankan dasbor Deployment

Tabel berikut memberikan rincian biaya untuk dasbor Deployment dengan parameter default dan 100 pengguna aktif di Wilayah AS Timur (Virginia N.) selama satu bulan, yang akan menelan biaya sekitar $20/bulan.

AWS service	Dimensi	Biaya [USD]
API Gateway, DynamoDB, CloudFront Amazon S3, Lambda, Toko Parameter Systems Manager	5.000 512 KB REST API panggilan per bulan tanpa caching diaktifkan	$1,97
Amazon Cognito	100 pengguna aktif per bulan dengan fitur keamanan canggih diaktifkan dan tidak ada pengguna yang masuk melalui federasi SAMP atau OIDC	$5,55

AWS service	Dimensi	Biaya [USD]
AWS WAF	10.000 permintaan web di 1 web ACL dan 7 aturan yang ditetapkan tanpa grup aturan	$12,60
Total biaya dasbor Deployment		$20.12

Biaya sampel untuk bukti konsep berbasis teks

Dasbor Deployment dapat memiliki banyak kasus penggunaan yang diterapkan pada waktu tertentu. Tabel berikut menunjukkan rincian biaya kasus penggunaan yang digunakan tanpa RAG untuk 1 pengguna bisnis yang melakukan 100 kueri per hari dengan LLM. Kueri dikirim sebagai pesan teks pada WebSocket dan respon dialirkan kembali sebagai token dengan asumsi bahwa streaming diaktifkan. Menggunakan model Amazon Bedrock Nova Pro, biaya menjalankan kasus penggunaan ini adalah sekitar $20/bulan.

AWS service	Dimensi	Biaya [USD]
API Gateway (WebSocket), CloudFront, Lambda, Amazon S3, AWS Systems Manager Parameter Store	100 interaksi obrolan per hari. Ukuran pesan rata-rata 32 KB per pesan dan 5 menit per koneksi.	$0,61
CloudWatch	CloudWatch Log 1,5 GB dengan mode verbose aktif untuk eksperimen	$7,23
Amazon DynamoDB	Tabel riwayat percakapan, penyimpanan 1 GB Tabel konfigurasi LLM, penyimpanan 1 GB	$3,05

AWS service	Dimensi	Biaya [USD]
Subtotal biaya kasus penggunaan (tidak termasuk LLMs)		$10,89
Batuan Dasar Amazon (Nova Pro)	Asumsi untuk 100 interaksi per hari: * Biaya bulanan untuk token input 190K per hari = $0.152 × 30 * Biaya bulanan untuk token keluaran 16K per hari = $0.0512 × 30	$6,10
Total biaya aplikasi dengan Amazon Bedrock (Nova Pro)	$10.89 (Biaya Kasus Penggunaan) +$6,10 (biaya Amazon Bedrock)	$17,00

> **Note**
>
> Biaya panggilan inferensi yang dilakukan ke layanan di luar jaringan AWS tidak termasuk dalam perkiraan ini. Lihat panduan harga penyedia LLM Anda jika Anda tidak menggunakan penyedia model AWS.
> Panduan harga untuk layanan AWS dapat ditemukan di: Harga Amazon Bedrock dan harga Amazon SageMaker AI.

Biaya sampel untuk mesin kueri AI generatif yang sangat skalabel

Tabel berikut memberikan rincian biaya kasus penggunaan yang mendukung RAG dengan model Nova Pro Amazon Bedrock sebagai LLM. Ketika Basis Pengetahuan Batuan Dasar ditambahkan, kasus penggunaan ini berharga sekitar $1300/bulan

AWS service	Dimensi	Biaya [USD]
API Gateway (WebSocket)	8000 interaksi obrolan per hari. Ukuran pesan rata-rata 32 KB per pesan dan 5 menit per koneksi.	$38,89
CloudFront	240.000 permintaan per bulan dengan data 100 GB ditransfer ke internet dan data 1 GB ditransfer ke asal	$8,76
Batuan Dasar Amazon (Nova Pro)	Asumsi: Token masukan = PromptTemplate (400) + konteks (400) +Chathistory (1080) +kueri Token masukan (20) = 1.900 Token keluaran = 160 (rata-rata) Dengan 8.000 transaksi per hari, Biaya Token Input Harian (1.900 x 8.000 = 15.200.000 token x harga 0,0008/1000 per token) Biaya Token Output Harian (160 x 8.000 = 1.280.000 token x 0,0032/1000 harga per token) Biaya bulanan (($12,16 + $4,10) x 30)	$487,80

AWS service	Dimensi	Biaya [USD]
CloudWatch	24 metrik menggunakan data 5 GB yang dicerna untuk log dan 1 dasbor	$9,72
DynamoDB	Tabel DynamoDB untuk melacak riwayat percakapan dengan setiap catatan hingga 1 KB data, 8.000 baca dan tulis per hari	$11,70
Lambda	Ukuran wadah - 128 MB, 512 MB fana penyimpanan, 2 fungsi Lambda yang digunakan untuk otorisasi Ukuran wadah - 256 MB, penyimpanan singkat 512 MB, 5 permintaan per detik dengan waktu komputasi rata-rata 20 detik	$20,89
Total biaya kasus penggunaan		$577.76/bulan+biaya dasar pengetahuan (lihat di bawah)

> **Note**
>
> Biaya panggilan API yang dilakukan ke layanan apa pun di luar jaringan AWS tidak termasuk dalam perkiraan ini. Lihat panduan harga penyedia LLM Anda jika tidak menggunakan Amazon Bedrock.

Biaya untuk menambahkan basis pengetahuan

Biaya dasar pengetahuan akan bervariasi berdasarkan jenis basis pengetahuan yang digunakan, dan (dalam kasus Bedrock) penyimpanan vektor pendukung yang digunakan oleh basis pengetahuan. Penyediaan dan pengelolaan basis pengetahuan berada di luar ruang lingkup solusi.

Basis Pengetahuan Amazon Bedrock

Solusi ini tidak mengelola atau menyediakan sumber daya apa pun yang terkait dengan Pangkalan Pengetahuan Amazon Bedrock. Amazon Bedrock tidak dikenakan biaya untuk menggunakan fitur basis pengetahuan itu sendiri, namun Anda akan dikenakan biaya untuk penggunaan model penyematan yang digunakan oleh kasus penggunaan Anda pada setiap kueri. Selain itu, penyimpanan vektor dukungan untuk basis pengetahuan Anda (misalnya, indeks di Amazon OpenSearch Service, atau database di dalam Amazon Relational Database Service) akan memiliki biaya terkait yang tidak dapat disediakan atau dihitung di sini.

Untuk skenario mesin kueri AI generatif yang sangat skalabel di atas, biaya yang dikeluarkan oleh layanan ini untuk memanggil model penyematan Amazon Bedrock adalah sebagai berikut:

AWS service	Dimensi	Biaya [USD]
Amazon Bedrock (Embeddings Teks Amazon Titan V2)	8.000 kueri sehari dengan 1.900 token input per kueri = 15.200.000 token = $0,30 USD per hari. Biaya harian x 30 hari = $9.00 USD biaya bulanan	$9.00
Penggunaan OpenSearch Sampel Layanan Amazon (Tanpa Server)	Konfigurasi dasar tanpa server dengan 4 x OpenSearch Compute Unit (OCU) (minimum yang dapat ditagih) = $23.04 USD per hari Biaya harian x 30 hari = $691.20 USD	$691,20

AWS service	Dimensi	Biaya [USD]
	Note Ini memberikan perkiraan kasar, karena beberapa beban kerja akan membutuhkan lebih banyak OCUs, sementara pelanggan dengan OpenSearch sumber daya yang disediakan yang ada akan dikenakan biaya lebih sedikit di sini.	
Total biaya tambahan		$700.20

Amazon Kendra

Solusinya dapat menyediakan indeks Kendra untuk Anda, atau Anda dapat membawa sendiri. Biaya untuk menjalankan konfigurasi yang sesuai dengan mesin kueri AI generatif yang sangat skalabel di atas adalah sebagai berikut:

AWS service	Dimensi	Biaya [USD]
Amazon Kendra	0-8.000 kueri sehari dan hingga 100.000 dokumen dengan Amazon Kendra Enterprise Edition dengan 0-50 sumber data	$1,008.00

> **Note**
>
> Anda dapat membagikan indeks Amazon Kendra di antara kasus penggunaan, tetapi ini dapat meningkatkan jumlah kueri per indeks. Jika ini berada di luar edisi Amazon Kendra Enterprise, biaya tambahan akan berlaku.

Biaya tambahan untuk mengaktifkan Amazon VPC untuk kasus penggunaan

Tabel berikut memberikan rincian biaya untuk mengaktifkan Amazon VPC untuk kasus penggunaan yang digunakan dalam dua. AZs

AWS service	Dimensi	Biaya [USD]
Gerbang NAT Amazon	Asumsi: 2 penyebaran AZ, dengan NAT Gateway di setiap AZ. 100 GB data diproses melalui NAT Gateway 730 jam, data 100 GB diproses per bulan	$74,70
AWS PrivateLink (Titik Akhir VPC)	Asumsi: 2 penyebaran AZ, dengan 1 subnet pribadi di setiap AZ dan 1 VPC Endpoint dengan 2 antarmuka jaringan elastis (). ENIs 6 titik akhir VPC, 2 per titik akhir ENIs VPC, 730 jam dengan data 1.024 GB diproses dalam sebulan	$97,84
IPv4 Alamat publik	Asumsi: 2 penyebaran AZ, 1 subnet publik di setiap AZ dengan NAT Gateway di setiap subnet publik. Setiap	$7,30

AWS service	Dimensi	Biaya [USD]
	NAT Gateway dikonfigurasi dengan 1 publik IPv4 aktif. 2 IPv4 alamat publik aktif x 730 jam dalam sebulan x $0,005 biaya per jam = $7,3 USD	
Biaya tambahan (untuk Amazon VPC)		$179,93

Implikasi biaya saat menggunakan Provisioned Throughput

Biaya throughput yang disediakan akan bervariasi berdasarkan jenis model yang telah Anda sediakan dan periode komitmen Anda serta Unit Model yang dipilih untuk periode komitmen. Ada biaya tambahan yang terkait dengan penggunaan Provisioned Throughput.

Untuk informasi lebih lanjut dan up-to-date harga terbanyak, Anda dapat merujuk ke Harga Batuan Dasar.

Biaya untuk menggunakan inferensi lintas wilayah

Tidak ada biaya tambahan untuk perutean atau transfer data untuk menggunakan inferensi lintas wilayah. Anda membayar harga yang sama per token untuk model seperti di sumber atau Wilayah utama Anda.

Biaya sampel untuk bukti konsep berbasis agen

Saat Anda menggunakan Agen Bedrock Amazon, Anda dikenakan biaya berdasarkan komponen yang terdiri dari agen, seperti model dukungan dan basis pengetahuan (jika RAG diaktifkan), bersama dengan kemampuan tambahan yang Anda tambahkan. Tabel berikut menunjukkan rincian biaya kasus penggunaan Agen Batuan Dasar yang dikonfigurasi dengan model Claude 3.5 Sonnet sesuai permintaan, Pangkalan Pengetahuan Amazon Bedrock, dan Amazon Bedrock Guardrails.

Mirip dengan biaya untuk menambahkan Pangkalan Pengetahuan Amazon Bedrock, solusi ini tidak mengelola atau menyediakan sumber daya yang terkait dengan Agen Bedrock Amazon. Solusinya

juga tidak mengeluarkan biaya untuk menggunakan Pangkalan Pengetahuan Amazon Bedrock, tetapi mengeluarkan biaya untuk:

- Menggunakan model penyematan untuk setiap kueri yang dikirim ke sana
- Penyimpanan vektor pendukung untuk basis pengetahuan Anda (misalnya, indeks di Amazon OpenSearch Service, atau database di dalam Amazon RDS)

Tabel berikut mengasumsikan 100 interaksi per hari dengan 1.900 token input dan 160 token output per kueri.

> **Note**
>
> Untuk contoh kasus penggunaan Agen Batuan Dasar ini, jika ada grup tindakan yang dikonfigurasi untuk menggunakan API eksternal, biaya tersebut akan menjadi tambahan. Mereka berada di luar lingkup perhitungan dalam tabel ini.

AWS service	Dimensi	Biaya [USD]
API Gateway (WebSocket), CloudFront, Lambda, Amazon S3, Toko Parameter Systems Manager	100 interaksi obrolan per hari, ukuran pesan rata-rata 32 KB per pesan, 5 menit per koneksi	$0,61
CloudWatch	1.5 GB CloudWatch Log dengan mode verbose aktif untuk eksperimen	$7,23
DynamoDB	Tabel konfigurasi LLM untuk ukuran rekaman 1KB dan penyimpanan 1 GB	$0,25
Subtotal biaya (tidak termasuk LLMs)		$8,09
Antropik Claude 3.5 Soneta	* Biaya harian untuk token input 190K per hari (0,003/1. 000 token) = $0,57 +	$24,30

AWS service	Dimensi	Biaya [USD]
	Biaya harian × 30 hari = $17,10 * Biaya harian untuk token keluaran 16K per hari (0,015/1.000 token) = $0,24 + Biaya harian × 30 hari = $7.20	
Amazon Bedrock (Amazon Titan Text Embeddings V2) untuk Basis Pengetahuan Amazon Bedrock	Biaya harian untuk token input 190K per hari (0,00002/1000 token) = 0,004 Biaya harian × 30 hari = $0,12	$0,12
Penggunaan OpenSearch sampel Layanan Amazon (Tanpa Server)	Konfigurasi tanpa server dasar dengan 4 × Unit OpenSearch Komputasi (OCU) (minimum yang dapat ditagih) = $23,04 per hari Biaya harian × 30 hari = $691.20	$691,20

AWS service	Dimensi	Biaya [USD]
Pagar Batuan Dasar Amazon	Token 190K kira-kira setara dengan 760K (190.000 × 4) karakter dan 3.800 unit teks (760K karakter/200) Pertimbangkan pagar pembatas yang dikonfigurasi dengan filter konten, filter informasi identifikasi pribadi (PII), filter informasi sensitif (ekspresi reguler) dan filter kata Biaya filter konten harian (0,75/1000 unit teks) +Biaya filter PII ($0,1/1000 unit teks) +filter informasi sensitif (regex) +filter kata = $2,85 + $0,38 + $0 + $0 Biaya bulanan = Biaya harian × 30 hari = $96,90	$96,90
Total biaya aplikasi untuk agen yang didukung oleh Anthropic Claude 3.5 Sonnet	$8,09 (biaya kasus penggunaan) +$812.52 (konfigurasi agen lainnya)	$820,61

> **Note**
>
> Lihat panduan harga penyedia LLM Anda jika Anda tidak menggunakan penyedia model AWS. Panduan harga untuk layanan AWS dapat ditemukan di: Harga Amazon Bedrock dan harga Amazon SageMaker AI.

Biaya sampel untuk MCP Server

Kasus penggunaan MCP Server memungkinkan penerapan dan pengelolaan server Protokol Konteks Model di Amazon Bedrock. AgentCore Tabel berikut menunjukkan rincian biaya kasus penggunaan Server MCP menggunakan metode Gateway untuk membungkus fungsi Lambda yang ada.

Solusinya mengelola penerapan dan konfigurasi AgentCore Gateway. Anda dikenakan biaya untuk:

- Biaya infrastruktur (API Gateway, Lambda, DynamoDB,, S3) CloudWatch
- AgentCore Konsumsi gateway (per pemanggilan alat)
- Biaya eksekusi fungsi Lambda (untuk metode Gateway dengan target Lambda)
- Biaya API eksternal (untuk metode Gateway dengan target API atau MCP Server, jika berlaku)

Item	Perhitungan	Biaya
Amazon API Gateway (REST API)	100 pemanggilan alat per hari × 30 hari = 3.000 permintaan per bulan	$0,05
AWS Lambda (orkestrasi)	100 doa per hari × 30 hari × 1 detik rata-rata × 512 MB = 3.000 GB-detik per bulan	$0,05
Amazon DynamoDB	3.000 read/write permintaan per bulan+penyimpanan 1 GB	$0,15
Amazon CloudWatch	Pemantauan dan pencatatan standar untuk 3.000 pemanggilan	$1,00
Amazon S3	Penyimpanan konfigurasi dan log (penggunaan minimal)	$0,25
Gerbang Dasar Dasar Amazon AgentCore	3.000 pemanggilan alat per bulan	$0,05

Item	Perhitungan	Biaya
Fungsi Lambda Target	100 doa per hari × 30 hari × 0,5 detik × 128 MB = 1.500 GB-detik per bulan	$0,25
Total biaya bulanan	$1,75 (infrastruktur) +$0,05 (Gateway) AgentCore	$1,80

> **Note**
>
> Biaya bervariasi berdasarkan metode penerapan (Gateway vs Runtime), jenis target, dan pola penggunaan. Penerapan metode runtime dikenakan biaya AgentCore Runtime alih-alih biaya Gateway. Biaya API eksternal dan biaya hosting kontainer khusus adalah tambahan.

Biaya sampel untuk Agent Builder

Agent Builder memungkinkan Anda untuk membuat dan menyebarkan agen kustom di Amazon Bedrock AgentCore. Tabel berikut menunjukkan rincian biaya kasus penggunaan Agen Builder yang dikonfigurasi dengan Claude 3.5 Sonnet, integrasi server MCP, dan memori jangka panjang diaktifkan.

Solusinya mengelola penerapan dan AgentCore konfigurasi Runtime. Anda dikenakan biaya untuk:

- Biaya infrastruktur (API Gateway, Lambda, DynamoDB,, S3) CloudWatch
- AgentCore Konsumsi runtime (CPU dan jam memori berdasarkan waktu eksekusi agen aktual)
- Inferensi model pondasi (token input dan output)
- AgentCore Memori (peristiwa jangka pendek dan penyimpanan/pengambilan jangka panjang)

Tabel berikut mengasumsikan 100 interaksi per hari dengan 1.900 token input dan 160 token output per kueri, dengan waktu eksekusi agen rata-rata 5 detik per interaksi.

AWS service	Dimensi	Biaya [USD]
API Gateway (WebSocket), CloudFront, Lambda, Amazon S3, Toko Parameter Systems Manager	100 interaksi obrolan per hari, ukuran pesan rata-rata 32 KB per pesan, 5 menit per koneksi	$0,61
CloudWatch	1.5 GB CloudWatch Log dengan mode verbose aktif untuk eksperimen	$7,23
DynamoDB	Tabel konfigurasi LLM untuk ukuran rekaman 1KB dan penyimpanan 1 GB	$0,25
Subtotal biaya infrastruktur		$8,09
Runtime Amazon Bedrock AgentCore	* CPU: 1 vCPU × 5 detik × 100 interaksi = 125 vCPU-seconds/day = 0.140 vCPU-hours/day + Biaya harian: 0.140 × $0.0895 = $0.013+Biaya bulanan: $0.013 × 30 = $0.38 * Memori: 512 MB (0,5 GB) × 5 detik × 100 interaksi = 250 GB- seconds/day = 0.069 GB-hours/day + Biaya harian: 0,069 × $0,00945 = $0,0007 + Biaya bulanan: $0,0007 × 30 = $0,02	$0,40
Antropik Claude 3.5 Soneta	* Biaya harian untuk token input 190K per hari (0,003/1.000 token) = $0,57+Biaya harian × 30 hari = $17,10	$24,30

AWS service	Dimensi	Biaya [USD]
	* Biaya harian untuk token keluaran 16K per hari (0,015/1.000 token) = $0,24+Biaya harian × 30 hari = $7,20	
Memori Batuan Dasar AgentCore Amazon	* Memori jangka pendek: 100 baru events/day × $0,25/1.000 acara = $0,025/hari+Biaya bulanan: $0,025 × 30 = $0,75 * Penyimpanan memori jangka panjang (strategi bawaan): 100 catatan × $0,75/1.000 = $0,075/bulan records/month * Pengambilan memori jangka panjang: 100 retrievals/day × $0,50/1.000 pengambilan = $0,05/hari+Biaya bulanan: $0,05 × 30 = $1,50	$2,33
Total biaya aplikasi untuk Agent Builder dengan Claude 3.5 Sonnet	$8,09 (infrastruktur) +$0,40 (AgentCore Runtime) +$24,30 (model) +$2,33 (memori)	$35,12

Note

AgentCore Harga runtime berbasis konsumsi. Biaya aktual tergantung pada:

- Waktu eksekusi agen (CPU dan penggunaan memori selama pemrosesan aktif)
- Jumlah interaksi dan kompleksitasnya
- Penggunaan alat MCP (tambahan CPU/memory untuk eksekusi alat)
- Konfigurasi memori (memori jangka pendek vs jangka panjang diaktifkan)

Untuk AgentCore harga terperinci, lihat harga Amazon Bedrock.

> **Note**
>
> Jika menggunakan server MCP yang memanggil eksternal APIs atau layanan, biaya tersebut tambahan dan di luar cakupan perhitungan ini. Demikian pula, jika menggunakan alat AgentCore Browser atau Penerjemah Kode, biaya berbasis konsumsi berlaku sebesar $0,0895 per jam VCPU dan $0,00945 per GB-jam.

Biaya sampel untuk Workflow Builder

Workflow Builder menciptakan agen supervisor yang mengatur beberapa agen Agent Builder. Tabel berikut menunjukkan rincian biaya untuk alur kerja dengan 1 agen pengawas dan 3 agen Agen Builder khusus, semuanya dikonfigurasi dengan Claude 3.5 Sonnet dan memori jangka panjang diaktifkan.

Asumsi: 100 interaksi per hari, rata-rata 2 delegasi agen per interaksi, waktu eksekusi 5 detik per agen.

AWS service	Dimensi	Biaya [USD]
API Gateway (WebSocket), CloudFront, Lambda, Amazon S3, Toko Parameter Systems Manager	100 interaksi obrolan per hari, ukuran pesan rata-rata 32 KB per pesan, 5 menit per koneksi	$0,61
CloudWatch	1.5 GB CloudWatch Log dengan mode verbose aktif untuk eksperimen	$7,23
DynamoDB	Tabel konfigurasi LLM untuk ukuran rekaman 1KB dan penyimpanan 1 GB	$0,25
Subtotal biaya infrastruktur		$8,09

AWS service	Dimensi	Biaya [USD]
AgentCore Runtime Amazon Bedrock (Agen Pengawas)	* CPU: 1 vCPU × 5 detik × 100 interaksi = 0.140 vCPU hours/day × 30 = $0.38 * Memory: 0.5 GB × 5 seconds × 100 interactions = 0.069 GB-hours/day - × 30 = $0.02	$0,40
Amazon Bedrock AgentCore Runtime (3 Agen Khusus)	* Rata-rata 2 delegasi per interaksi = 200 agen executions/day * CPU: 1 vCPU × 5 seconds × 200 = 0.278 vCPU-hours/day × 30 = $0.75 * Memory: 0.5 GB × 5 seconds × 200 = 0.139 GB-hours/day × 30 = $0,04	$0,79
Antropik Claude 3.5 Soneta (Agen Pengawas)	* Masukan: 190K tokens/day × $0.003/1K = $0.57/hari × 30 = $17.10 * Keluaran: 16K × $0.015/1K = $0.24/hari × 30 = $7.20 tokens/day	$24,30
Anthropic Claude 3.5 Soneta (Agen Khusus)	* Rata-rata 2 delegasi per interaksi * Masukan: 380K tokens/day × $0,003/1K = $1,14/hari × 30 = $34,20 * Keluaran: 32K × $0,015/1K = $0,48/hari × 30 = $14,40 tokens/day	$48,60

AWS service	Dimensi	Biaya [USD]
Amazon Bedrock AgentCore Memory (Agen Supervisor)	* Jangka pendek: 100 events/day × $0,25/1K × 30 = $0,75 * Penyimpanan jangka panjang: 100 catatan × $0,75/1K = $0,08 * Pengambilan jangka panjang: 100 × $0,50/1K × 30 = $1,50 retrievals/day	$2,33
Amazon Bedrock AgentCore Memory (Agen Khusus)	* Jangka pendek: 200 events/day × $0,25/1K × 30 = $1,50 * Penyimpanan jangka panjang: 200 catatan × $0,75/1K = $0,15 * Pengambilan jangka panjang: 200 × $0,50/1K × 30 = $3,00 retrievals/day	$4,65
Total biaya aplikasi untuk Workflow Builder dengan 3 agen	$8,09 (infrastruktur) +$1,19 (AgentCore Runtime) +$72,90 (model) +$6,98 (memori)	$89,16

Note

- Tingkat delegasi yang lebih tinggi meningkatkan konsumsi token secara proporsional

Untuk AgentCore harga terperinci, lihat harga Amazon Bedrock.

Keamanan

Saat Anda membangun sistem pada infrastruktur AWS, tanggung jawab keamanan dibagi antara Anda dan AWS. Model tanggung jawab bersama ini mengurangi beban operasional Anda karena AWS mengoperasikan, mengelola, dan mengontrol komponen termasuk sistem operasi host, lapisan virtualisasi, dan keamanan fisik fasilitas tempat layanan beroperasi. Untuk informasi selengkapnya tentang keamanan AWS, kunjungi AWS Cloud Security.

Menggunakan model pondasi di Amazon Bedrock

Amazon Bedrock menyelenggarakan koleksi model dari model Amazon Nova hingga model pondasi terkemuka lainnya (FMs). Saat menggunakan Amazon Bedrock, semua model di-host dalam infrastruktur AWS. Ini berarti bahwa saat menggunakan Amazon Bedrock sebagai penyedia LLM, semua permintaan inferensi Anda akan tetap berada dalam jaringan AWS dan lalu lintas jaringan tidak akan meninggalkan Wilayah Anda.

> **Note**
>
> Semua model foundation (FMs) yang tersedia melalui Amazon Bedrock di-host langsung di infrastruktur AWS yang dikelola dan dimiliki oleh AWS. Penyedia model tidak memiliki akses ke data pelanggan seperti petunjuk dan kelanjutan, atau log layanan Amazon Bedrock. Untuk informasi tambahan tentang postur keamanan Amazon Bedrock, lihat Perlindungan data di Amazon Bedrock di Panduan Pengguna Amazon Bedrock.

Peran IAM

Peran IAM memungkinkan pelanggan untuk menetapkan kebijakan akses terperinci dan izin ke layanan dan pengguna di AWS Cloud. Solusi ini menciptakan peran IAM yang memberikan akses fungsi Lambda solusi untuk membuat sumber daya Regional.

CloudWatch Log

Anda dapat mengaktifkan mode verbose saat menerapkan kasus penggunaan menggunakan halaman pemilihan model Dasbor Deployment, di bawah Pengaturan Tambahan. Mode verbose memungkinkan CloudWatch log terperinci yang dapat membantu untuk debugging dan eksperimen cepat.

> **Note**
>
> Ketika mode verbose diaktifkan, dokumen yang diambil dari basis pengetahuan (jika RAG diaktifkan) dan prompt juga akan dicatat, yang mungkin berisi informasi sensitif.

VPC

Solusinya menyediakan dua opsi untuk konfigurasi Amazon VPC:

1. Biarkan solusinya membangun VPC Amazon untuk Anda.
2. Mengelola dan membawa VPC Amazon Anda sendiri untuk digunakan dalam solusi.

Biarkan solusinya membangun VPC Amazon untuk Anda

Jika Anda memilih opsi untuk membiarkan solusi membangun VPC Amazon, itu akan diterapkan sebagai arsitektur 2-AZ secara default dengan rentang CIDR 10.10.0.0/20. Anda memiliki opsi untuk menggunakan Amazon VPC IP Address Manager (IPAM), dengan 1 subnet publik dan 1 subnet pribadi di setiap AZ. Solusinya menciptakan Gateway NAT di setiap subnet publik, dan mengonfigurasi fungsi Lambda untuk membuat subnet pribadi. ENIs Selain itu, konfigurasi ini membuat tabel rute dan entri, grup keamanan dan aturannya, jaringan ACLs, titik akhir VPC (gateway dan titik akhir antarmuka).

Mengelola VPC Amazon Anda sendiri

Saat menerapkan solusi dengan VPC Amazon, Anda memiliki opsi untuk menggunakan VPC Amazon yang ada di akun dan Wilayah AWS Anda. Kami menyarankan agar VPC Anda tersedia di setidaknya dua zona ketersediaan untuk memastikan ketersediaan yang tinggi. VPC Anda juga harus memiliki titik akhir VPC berikut dan kebijakan IAM terkait untuk konfigurasi VPC dan tabel rute Anda.

Untuk dasbor Deployment Amazon VPC

1. Titik akhir Gateway untuk DynamoDB.
2. Titik akhir gateway untuk S3.
3. Titik akhir antarmuka untuk CloudWatch.
4. Titik akhir antarmuka untuk AWS CloudFormation.

Untuk kasus penggunaan Amazon VPC

1. Titik akhir Gateway untuk DynamoDB.
2. Titik akhir gateway untuk S3.
3. Titik akhir antarmuka untuk CloudWatch.
4. Endpoint antarmuka untuk Systems Manager Parameter Store.

> **Note**
>
> Solusinya hanya membutuhkan`com.amazonaws.region.ssm`.

5. Titik akhir antarmuka untuk Amazon Bedrock (bedrock-runtime, agent-runtime,). bedrock-agent-runtime
6. Opsional: Jika penerapan akan menggunakan Amazon Kendra sebagai basis pengetahuan, maka titik akhir antarmuka untuk Amazon Kendra diperlukan.
7. Opsional: jika penerapan akan menggunakan LLM apa pun di bawah Amazon Bedrock, maka titik akhir antarmuka untuk Amazon Bedrock diperlukan.

> **Note**
>
> Solusinya hanya membutuhkan`com.amazonaws.region.bedrock-runtime`.

8. Opsional: Jika penerapan akan menggunakan Amazon SageMaker AI untuk LLM, maka titik akhir antarmuka untuk Amazon SageMaker AI diperlukan.

> **Note**
>
> Solusinya tidak akan menghapus atau memodifikasi konfigurasi VPC saat menggunakan opsi Bawa penyebaran VPC Anda sendiri. Namun, itu akan menghapus apa pun VPCs yang dibuat oleh solusi di opsi Buat VPC untuk saya. Untuk alasan ini, Anda harus berhati-hati saat membagikan VPC yang dikelola solusi di seluruh tumpukan/penerapan.
> Misalnya, penerapan A menggunakan opsi Buat VPC untuk saya. Deployment B menggunakan Bring My Own VPC menggunakan VPC yang dibuat oleh deployment A. Jika deployment A dihapus sebelum deployment B, maka deployment B tidak akan berfungsi lagi karena VPC telah dihapus. Juga karena penerapan B menggunakan fungsi yang ENIs dibuat oleh Lambda, menghapus penerapan A mungkin memiliki kesalahan dan retensi sumber daya sisa.

Amazon CloudFront

Solusi ini menerapkan konsol web yang dihosting di bucket Amazon S3. Untuk membantu mengurangi latensi dan meningkatkan keamanan, solusi ini mencakup CloudFront distribusi dengan

identitas akses asal, yaitu CloudFront pengguna yang menyediakan akses publik ke konten bucket situs web solusi. Untuk informasi selengkapnya, lihat Membatasi Akses ke Konten Amazon S3 dengan Menggunakan Identitas Akses Asal di Panduan Pengembang CloudFront Amazon.

> **Note**
>
> CloudFront memiliki batas kuota lunak tingkat akun dari 20 kebijakan header respons. Solusi ini membuat kebijakan header respons khusus untuk tujuan keamanan. Jika Anda memiliki lebih dari 20 penerapan Generative AI Application Builder di AWS atau kasus penggunaannya, penerapan baru mungkin gagal karena mencapai batas kuota.

Untuk mengatasi masalah ini, Anda dapat meminta peningkatan kuota untuk kuota Kebijakan Header Respons di konsol AWS Service Quotas dengan mengikuti langkah-langkah berikut:

1. Buka konsol AWS Service Quotas.
2. Di panel navigasi, pilih Layanan AWS.
3. Cari dan pilih Amazon CloudFront.
4. Gulir ke kuota Kebijakan Header Respons dan pilih Permintaan peningkatan kuota.
5. Ikuti petunjuk untuk meminta peningkatan batas kuota untuk akun AWS Anda.

Dengan meningkatkan kuota Kebijakan Header Respons, Anda dapat memastikan bahwa penerapan baru Generative AI Application Builder di AWS atau kasus penggunaannya tidak gagal karena batas kuota.

Kuota

Service quotas, juga disebut batasan, adalah jumlah maksimum sumber daya layanan atau operasi untuk akun AWS Anda.

Kuota untuk layanan AWS dalam solusi ini

Pastikan Anda memiliki kuota yang cukup untuk setiap layanan yang diterapkan dalam solusi ini. Untuk informasi selengkapnya, lihat kuota layanan AWS.

Gunakan tautan berikut untuk membuka halaman untuk layanan itu. Untuk melihat kuota layanan untuk semua layanan AWS dalam dokumentasi tanpa berpindah halaman, lihat informasi di titik akhir Layanan dan halaman kuota di PDF sebagai gantinya.

Kuota Amazon Bedrock AgentCore

Untuk penerapan Agent Builder, perhatikan kuota layanan Amazon Bedrock berikut AgentCore :

Kuota	AS Timur (Virginia Utara)	Daerah Lain
Beban kerja Sesi Aktif per akun	1000	500
Total agen per akun	1.000	1.000
Versi per akun	1.000	1.000

Terapkan solusinya

Solusi ini menggunakan CloudFormation templat dan tumpukan AWS untuk mengotomatiskan penerapannya. CloudFormation Template menentukan sumber daya AWS yang disertakan dalam solusi ini dan propertinya. CloudFormation Tumpukan menyediakan sumber daya yang dijelaskan dalam template.

Ikhtisar proses penyebaran

Sebelum Anda meluncurkan solusi, tinjau biaya, arsitektur, keamanan, dan pertimbangan lain yang dibahas dalam panduan ini.

> **Important**
>
> Jika Anda berencana untuk menggunakan Amazon Bedrock, Anda harus meminta akses ke model sebelum tersedia untuk digunakan. Lihat akses Model di Panduan Pengguna Amazon Bedrock untuk detail selengkapnya.

Waktu untuk menyebarkan: Sekitar 10 menit

Langkah 1: Luncurkan tumpukan dasbor Deployment

Langkah 2: Menyebarkan kasus penggunaan

Langkah 3: Menerapkan kasus penggunaan menggunakan wizard dasbor Deployment

Langkah 4: Konfigurasi pasca-penyebaran

Secara opsional, Anda dapat menerapkan kasus penggunaan secara terpisah dari solusi, jika Anda memilih untuk tidak memiliki UI dasbor Deployment atau. APIs

- Menerapkan kasus penggunaan Teks mandiri
- Menerapkan kasus penggunaan Agen Batuan Dasar mandiri

Anda juga dapat menyediakan konfigurasi obrolan DynamoDB.

> **Important**
>
> Solusi ini mengirimkan metrik operasional ke AWS (“Data”) tentang penggunaan solusi ini. Kami menggunakan Data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini serta layanan serta produk terkait. Pengumpulan AWS atas Data ini tunduk pada Kebijakan Privasi AWS.

CloudFormation Templat AWS

Anda dapat mengunduh CloudFormation template untuk solusi ini sebelum menerapkannya.

View template

genera
ai-application-builder-on-aws.template - Gunakan template ini untuk meluncurkan solusi dan semua komponen terkait. Konfigurasi default menerapkan solusi inti dan pendukung yang ditemukan di layanan AWS di bagian solusi ini, tetapi Anda dapat menyesuaikan template untuk memenuhi kebutuhan spesifik Anda.

> **Note**
>
> CloudFormation Sumber daya AWS dibuat dari konstruksi AWS Cloud Development Kit (AWS CDK).

CloudFormation Template AWS ini menerapkan Generative AI Application Builder di AWS di AWS Cloud.

Langkah 1: Luncurkan tumpukan dasbor Deployment

Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menyebarkan solusi ke akun Anda.

Waktu untuk menyebarkan: Sekitar 10 menit

1. Masuk ke AWS Management Console dan pilih tombol untuk meluncurkan `generative-ai-application-builder-on-aws.template` CloudFormation template.

   Launch solution

2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di Wilayah AWS yang berbeda, gunakan pemilih Wilayah di bilah navigasi konsol.

> **Note**
>
> Solusi ini menggunakan Amazon Kendra dan Amazon Bedrock, yang saat ini tidak tersedia di semua Wilayah AWS. Jika menggunakan fitur-fitur ini, Anda harus meluncurkan solusi ini di Wilayah AWS tempat layanan ini tersedia. Untuk ketersediaan terbaru menurut Wilayah, lihat Daftar Layanan Regional AWS.

3. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ada di kotak teks URL Amazon S3 dan pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda. Untuk informasi tentang batasan penamaan karakter, lihat Batas IAM dan STS di Panduan Pengguna AWS Identity and Access Management.
5. Di bawah Parameter, tinjau parameter untuk templat solusi ini dan modifikasi sesuai kebutuhan. Solusi ini menggunakan nilai default berikut.

Parameter	Default	Deskripsi
Email Pengguna Admin	No	Alamat email pengguna admin yang akan memiliki akses ke dasbor Deployment. Jika disediakan, grup dan pengguna Amazon Cognito akan dibuat dengan izin untuk menerapkan dan mengelola kasus penggunaan. Anda juga dapat menggunakan `placeholder@example.com` untuk membuat Grup tetapi bukan Pengguna. Lihat Konfigurasi Kumpulan Pengguna Manual untuk informasi tentang pengaturan kumpulan pengguna Anda.

Parameter	Default	Deskripsi
VpcEnabled	No	Haruskah dasbor Deployment digunakan dalam VPC
CreateNewVpc	No	Hanya tersedia, jika VpcEnabledadaYes. Jika nilainyaYes, tumpukan akan membuat VPC dan menyebarkan solusi dalam VPC yang dibuat. Jika VpcEnabledada Yes dan CreateNewVpcsedangNo, maka Anda harus menyediakan konfigurasi VPC yang ada (ExistingVpcId,, ExistingPrivateSubnetIdsExistingSecurityGroupIds, VpcAzs).
IPAMPoolId	(Masukan opsional)	Anda dapat mengonfigurasi IPAM dan memberikan id yang dibuat sebagai input untuk menetapkan rentang alamat IP yang harus digunakan penyebaran tumpukan ini. Untuk detail mengenai IPAM, lihat Cara kerja IPAM.

Parameter	Default	Deskripsi
DeployUI	`Yes`	Anda memiliki opsi untuk menerapkan dasbor Deployment tanpa antarmuka pengguna web (dan sumber daya AWS yang diperluka n untuk penerapan web). Dalam hal ini, solusinya akan menerapkan semua infrastru ktur termasuk titik akhir REST API. Opsi ini berguna untuk mengintegrasikan antarmuka web Anda sendiri dengan dasbor APIs Deployment.
ExistingVpcId	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di VPC yang sudah ada yang telah Anda buat.
ExistingPrivateSubnetIds	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di VPC yang sudah ada yang telah Anda buat. Fungsi Lambda akan digunakan di subnet ini.
ExistingSecurityGroupIds	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di VPC yang sudah ada yang telah Anda buat. Pastikan grup keamanan memiliki izin untuk koneksi TCP keluar.

Parameter	Default	Deskripsi
VpcAzs	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di VPC yang sudah ada yang telah Anda buat.
CognitoDomainPrefix	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di kumpulan pengguna Amazon Cognito yang sudah ada yang Anda buat. Jika Anda tidak memberikan nilai, solusi menghasilkannya.
ExistingCognitoUserPoolId	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di kumpulan pengguna Amazon Cognito yang sudah ada yang Anda buat.
ExistingCognitoUserPoolClient	(Masukan opsional)	Diperlukan hanya jika Anda ingin menerapkan solusi di kumpulan pengguna Amazon Cognito yang sudah ada yang Anda buat. Jika Anda tidak memberikan nilai, solusinya akan menciptakan klien kumpulan pengguna. Parameter ini hanya dapat diberikan jika Anda memberikan ExistingCognitoUserPoolIdnilai.

6. Pilih Berikutnya.
7. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.

8. Pada halaman Tinjau dan buat, tinjau dan konfirmasikan pengaturan. Pilih kotak yang menyatakan bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
9. Pilih Kirim untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu sekitar 10 menit.

Langkah 2: Menyebarkan kasus penggunaan

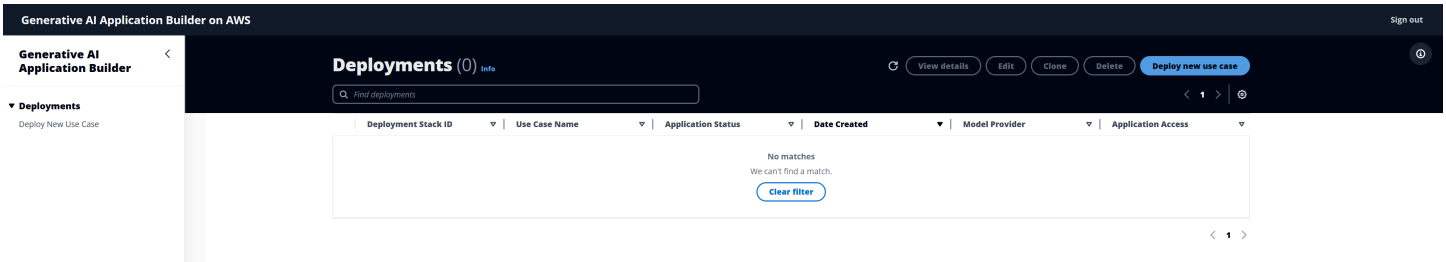
> **Important**
>
> Setelah tumpukan berhasil diterapkan, email pendaftaran dikirim ke email pengguna admin yang dikonfigurasi. Dengan menggunakan kredensi tersebut, pengguna admin dapat masuk ke dasbor Deployment untuk menggunakan aplikasi web.

> **Note**
>
> DevOps Pengguna yang memiliki akses ke AWS Management Console harus menyediakan CloudFront URL UI dasbor Deployment kepada pengguna admin saat tumpukan selesai. URL dapat ditemukan di tab Output dari CloudFormation tumpukan.

1. Masuk ke dasbor Deployment sebagai pengguna admin.
2. Pada halaman landing aplikasi, pilih Deploy new use case.

   Ini meluncurkan wizard penerapan, yang memandu Anda melalui pembuatan kasus penggunaan.

Menggambarkan halaman landing dasbor Deployment - penyebaran baru



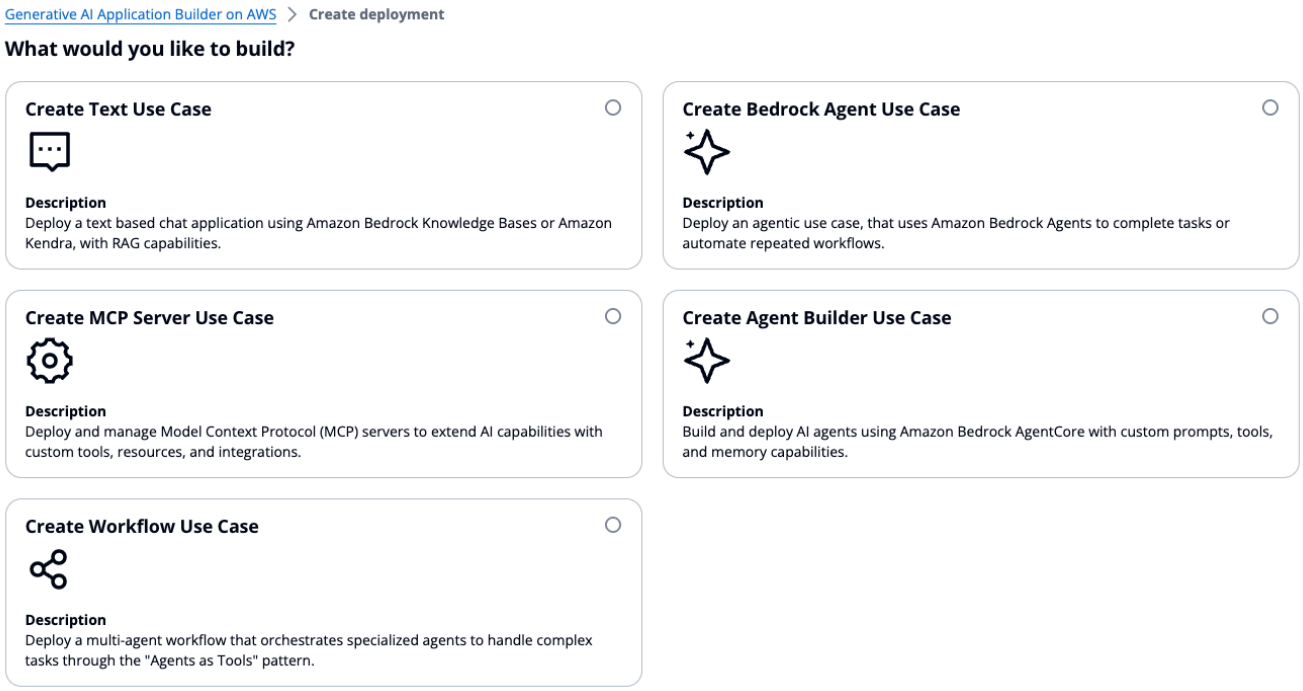
> **Note**
>
> Jika Anda perlu menambahkan pengguna tambahan ke penerapan Anda, lihat kumpulan pengguna Mengelola Cognito untuk detail selengkapnya.

Langkah 3: Menerapkan kasus penggunaan menggunakan wizard dasbor Deployment

Di wizard dasbor Deployment, Anda harus memilih antara yang berikut ini:

- Kasus penggunaan teks - Menyebarkan aplikasi obrolan, dengan kemampuan RAG opsional
- Kasus penggunaan Agen Batuan Dasar - Menggunakan Agen Bedrock Amazon untuk menyelesaikan tugas atau mengotomatiskan alur kerja berulang
- MCP Server - Menyebarkan dan mengelola server MCP dengan metode gateway atau runtime
- Agent Builder - Bangun dan terapkan agen kustom AgentCore dengan integrasi MCP dan manajemen memori
- Workflow Builder - Mengatur beberapa agen Agen Builder menggunakan delegasi hierarkis

Menampilkan lima opsi: Buat kasus penggunaan Teks, Buat kasus penggunaan Agen Batuan Dasar, Buat Kasus Penggunaan Server MCP, Buat Kasus Penggunaan Pembuat Agen, atau Buat Kasus Penggunaan Alur Kerja.



Langkah 3a: Menyebarkan kasus penggunaan Teks

Bagian ini memberikan instruksi untuk menerapkan kasus penggunaan Teks.

Pilih kasus penggunaan

Saat Anda memilih Buat kasus penggunaan Teks, UI akan membuka layar Select use case. Saat diminta, berikan informasi berikut:

- Gunakan nama kasus.
- Alamat email opsional untuk pengguna default kasus penggunaan yang akan ditambahkan ke kumpulan pengguna Amazon Cognito untuk kasus penggunaan, dan akan diberikan izin untuk berinteraksi dengannya.
- Apakah Anda ingin menerapkan UI dengan kasus penggunaan ini. Jika Anda tidak ingin menerapkan UI dengan kasus penggunaan, Anda dapat menggunakan titik akhir API yang diterapkan untuk digunakan dengan aplikasi Anda.

Detail kasus penggunaan

Langkah detail kasus penggunaan memungkinkan Anda mengonfigurasi pengaturan tambahan untuk penerapan Anda.

Secara default, kasus penggunaan Teks membuat dan mengonfigurasi kumpulan pengguna Amazon Cognito untuk Anda saat solusi menerapkan dasbor Deployment. Solusi ini mengautentikasi kasus penggunaan baru dengan klien yang baru dibuat di kumpulan pengguna yang sama. Namun, Anda dapat memberikan ID kumpulan pengguna dan ID klien yang ada di langkah ini jika Anda ingin menggunakan kumpulan pengguna dan klien Amazon Cognito Anda sendiri dengan kasus penggunaan.

Important

Pengguna admin memiliki akses ke semua kasus penggunaan yang diterapkan saat kumpulan pengguna Amazon Cognito dibuat melalui panduan penerapan. Jika Anda menyediakan kumpulan pengguna sendiri selama penerapan, Anda harus memastikan bahwa admin memiliki izin untuk mengakses kasus penggunaan yang diterapkan. Anda juga perlu memperbarui callback yang Diizinkan URLs dan keluar yang Diizinkan di klien Aplikasi Anda URLs di Cognito. Untuk melakukannya:

1. Arahkan ke konsol Cognito
2. Pilih Kolam Pengguna.
3. Pilih kolam pengguna Anda.
4. Pilih Klien Aplikasi di menu sebelah kiri.
5. Pilih klien aplikasi yang ingin Anda modifikasi.
6. Pilih tab Halaman Login.
7. Pilih Edit dan tambahkan URLs.
8. Pilih Simpan perubahan.

Selain itu, jika Anda perlu menambahkan lebih banyak pengguna ke kasus penggunaan, lihat bagian Mengelola kumpulan pengguna Cognito.

Pilih konfigurasi jaringan

Langkah panduan ini memungkinkan Anda untuk menerapkan kasus penggunaan dengan Amazon Virtual Private Cloud (Amazon VPC) yang sudah ada sebelumnya atau baru. Jika memilih VPC yang sudah ada sebelumnya, Anda harus memberikan ID VPC, hingga 16 ID subnet dan hingga 5 grup keamanan IDs untuk digunakan dengan VPC ini. Jika Anda tidak menggunakan VPC yang sudah ada sebelumnya, pengaturan ini akan dikonfigurasi untuk Anda.

Pilih model

Pada langkah Pilih model, Anda dapat memilih penyedia model Anda dari menu tarik-turun. Ada dua opsi: Bedrock dan SageMaker.

Jika Anda memilih SageMaker, Anda dapat membuat titik akhir model SageMaker AI di konsol SageMaker AI dan memberikan skema input yang diharapkan model dan output JSONPath untuk respons LLM. Anda dapat merujuk ke bagian Menggunakan Amazon SageMaker AI sebagai Penyedia LLM dan contoh muatan SageMaker AI yang disediakan di repositori solusi. GitHub

Jika Anda memilih Amazon Bedrock, Anda akan disajikan dengan empat opsi:

- Model Mulai Cepat - Memulai dengan cepat dengan koleksi model dengan price/performance karakteristik berbeda. Direkomendasikan untuk membangun aplikasi pertama Anda. Opsi ini memungkinkan Anda untuk memilih nama model dari daftar yang disediakan.
- Model Foundation Lainnya - Akses berbagai model pondasi dengan kemampuan dan spesialisasi yang berbeda. Opsi ini memungkinkan Anda memasukkan ID model untuk model fondasi sesuai permintaan Bedrock yang Anda inginkan.
- Profil Inferensi - Profil inferensi memanfaatkan inferensi lintas wilayah Bedrock untuk meningkatkan throughput dan meningkatkan ketahanan dengan merutekan permintaan Anda di beberapa Wilayah AWS selama ledakan pemanfaatan puncak. Opsi ini memungkinkan Anda untuk memasukkan ID profil inferensi yang ingin Anda gunakan.
- Model yang Disediakan - Kapasitas throughput khusus untuk beban kerja produksi yang membutuhkan kinerja yang konsisten. Opsi ini memungkinkan Anda untuk memasukkan ARN provisioned/custom model yang akan digunakan dari Amazon Bedrock.

Langkah pemilihan model juga memungkinkan Anda memilih pengaturan model lanjutan Anda. Lihat pengaturan LLM lanjutan untuk detail tentang mengonfigurasi Amazon Bedrock Guardrails, throughput yang disediakan untuk Amazon Bedrock, dan parameter model tambahan.

Inferensi lintas wilayah

Inferensi lintas wilayah membantu pengguna Amazon Bedrock mengelola ledakan lalu lintas yang tidak direncanakan dengan mulus dengan menggunakan komputasi di berbagai Wilayah AWS. Untuk menggunakan inferensi lintas wilayah, Anda memerlukan profil inferensi. Profil inferensi adalah abstraksi atas kumpulan sumber daya sesuai permintaan dari kumpulan Wilayah AWS yang dikonfigurasi. Ini dapat merutekan permintaan inferensi Anda, yang berasal dari Wilayah sumber

Anda, ke Wilayah lain yang dikonfigurasi dalam kumpulan itu. Ini memungkinkan distribusi lalu lintas di beberapa Wilayah AWS. Ini membantu memungkinkan throughput yang lebih tinggi dan peningkatan ketahanan selama periode permintaan puncak.

Profil inferensi dinamai sesuai model dan Wilayah yang mereka dukung. Anda harus memanggil profil inferensi dari salah satu Wilayah yang disertakan. Misalnya, seperti yang ditunjukkan pada tabel berikut, ID profil inferensi `us.anthropic.claude-3-haiku-20240307-v1:0` memungkinkan distribusi lalu lintas `us-east-1` dan `us-west-2` Wilayah model yang Anda pilih. Model tertentu hanya tersedia dengan profil inferensi di Wilayah tertentu.

Profil inferensi	ID profil inferensi	Wilayah termasuk
Antropik AS Claude 3 Haiku	`us.anthropic.claude-3-haiku-20240307-v1:0`	US East (N. Virginia) (`us-east-1`) US West (Oregon) (`us-west-2`)

Jika Anda ingin menggunakan ID profil inferensi alih-alih ID model, maka Anda harus mengidentifikasi ID profil inferensi yang sesuai. Lihat Wilayah dan model yang Didukung untuk profil inferensi di Panduan Pengguna Amazon Bedrock untuk informasi selengkapnya. Di konsol Amazon Bedrock, opsi inferensi lintas wilayah di menu navigasi kiri menyediakan profil inferensi ini. IDs

Setelah mengidentifikasi ID profil inferensi yang akan digunakan, Anda dapat menggunakannya selama tahap Pilih model dengan melakukan langkah-langkah berikut:

1. Pilih Amazon Bedrock sebagai penyedia model.
2. Pilih opsi tombol radio Profil Inferensi.
3. Masukkan ID profil inferensi Anda di kotak teks yang muncul.

Lihat Tingkatkan ketahanan dengan inferensi lintas wilayah di Panduan Pengguna Amazon Bedrock untuk detail selengkapnya tentang profil inferensi.

Pilih basis pengetahuan

Jika Anda ingin menerapkan kasus penggunaan Non-Retrieval Augmented Generation (RAG), Anda dapat melewati langkah ini.

Namun, jika Anda ingin mengaktifkan RAG sebagai bagian dari penerapan Anda, Anda sekarang dapat memberikan Id Indeks Amazon Kendra yang telah dikonfigurasi sebelumnya atau ID Basis Pengetahuan Amazon Bedrock. Anda juga dapat membuat Indeks Amazon Kendra baru untuk digunakan dengan solusinya. Solusinya saat ini mendukung Amazon Kendra dan Amazon Bedrock Knowledge Bases sebagai basis pengetahuan untuk penerapan kasus penggunaan berbasis RAG Anda.

Lihat bagian Mengonfigurasi Basis Pengetahuan untuk panduan tentang memasukkan data ke dalam basis pengetahuan untuk digunakan dengan penerapan berbasis RAG Anda.

Konfigurasi RAG tingkat lanjut

Wizard memungkinkan Anda memilih opsi lanjutan untuk digunakan dengan penyebaran RAG Anda seperti jumlah dokumen untuk diambil setiap kali kueri dikirim ke basis pengetahuan Anda, respons teks statis dari LLM ketika tidak ada dokumen yang ditemukan di basis pengetahuan, apakah Anda ingin menampilkan sumber dokumen dengan respons LLM Anda untuk pemeriksaan kewarasan, dll. Anda juga dapat mengonfigurasi konfigurasi khusus basis pengetahuan untuk Amazon Kendra seperti Kontrol Akses Berbasis Peran (RBAC), atau Ganti Jenis Pencarian saat menggunakan Amazon Tanpa Server dengan Pangkalan Pengetahuan Amazon OpenSearch Bedrock. Lihat bagian Pengaturan Dasar Pengetahuan Tingkat Lanjut untuk detail selengkapnya tentang pengaturan lanjutan ini.

> **Note**
>
> Basis pengetahuan Anda harus berada di akun dan Wilayah yang sama dengan dasbor Deployment yang diterapkan dan tumpukan kasus penggunaan.

Pilih prompt dan batas token

Pada langkah ini, Anda dapat mengonfigurasi prompt Anda untuk digunakan dengan LLM. Prompt mungkin memerlukan placeholder seperti`{input}`, dan. `{history}` `{context}` Placeholder ini menginstruksikan LLM tentang tempat untuk menarik masukan pengguna, riwayat percakapan, dan informasi yang diambil dari basis pengetahuan.

- Untuk penyedia model Bedrock, prompt sistem harus disediakan yang tidak memiliki batasan untuk kasus penggunaan non-RAG. Prompt disambiguasi untuk penyedia model Bedrock bagaimanapun, membutuhkan minimal dua placeholder - dan `{input}` `{history}`

- Untuk penyedia SageMaker model, prompt sistem dan disambiguasi, keduanya memerlukan minimal dua placeholder - dan. `{input}` `{history}`
- Untuk kasus penggunaan RAG, untuk setiap penyedia model, `{context}` placeholder juga diperlukan.

Untuk informasi selengkapnya, lihat Mengonfigurasi prompt Anda. Anda juga dapat merujuk ke bagian Tips untuk mengelola batasan token model saat memilih ukuran batas token untuk permintaan Anda.

Aktifkan masukan multimodal

Langkah ini memungkinkan Anda mengaktifkan kemampuan input multimodal untuk kasus penggunaan Anda. Saat diaktifkan, pengguna dapat mengunggah dan mengirim gambar dan dokumen beserta kueri teks mereka.

Jenis dan kendala file yang didukung:

- Gambar: Hingga 20 gambar per pesan. Setiap gambar harus berukuran tidak lebih dari 3,75 MB dan tinggi dan lebar 8.000 px. Format yang didukung: png, jpeg, gif, webp
- Dokumen: Hingga 5 dokumen per pesan. Setiap dokumen harus berukuran tidak lebih dari 4,5 MB. Format yang didukung: pdf, csv, doc, docx, xls, xlsx, html, txt, md

Cara menggunakan input multimodal:

1. Aktifkan MultimodalEnabledparameter selama penerapan kasus penggunaan
2. Di antarmuka obrolan, pengguna dapat mengunggah file dengan dua cara:
   - Mengklik tombol unggah di kotak input obrolan, atau
   - Menyeret dan menjatuhkan file langsung ke antarmuka obrolan
3. File diunggah ke Amazon S3 dan diproses oleh model yang dipilih
4. File yang diunggah secara otomatis dihapus setelah 48 jam

Pelacakan status file:

DevOps pengguna dapat memantau metadata file di DynamoDB, yang mencakup waktu unggah dan status pemrosesan. File dapat memiliki status berikut:

- pending - Pengunggahan file telah dimulai tetapi belum selesai. Ini adalah status awal ketika URL presigned dihasilkan.
- upload - File telah berhasil diunggah ke S3 dan siap untuk diproses oleh model.
- dihapus - File telah dihapus oleh pengguna dan seharusnya tidak lagi dapat diakses untuk diproses.
- tidak valid - Pemeriksaan validasi file gagal (misalnya, ketidakcocokan jenis file atau kegagalan validasi keamanan).

File dalam status tertunda yang tidak pernah diunggah akan dibersihkan secara otomatis ketika TTL mereka kedaluwarsa. Hanya file dengan status upload yang dapat diproses oleh model.

Bucket multimodal S3 dan tabel metadata DynamoDB tersedia di output Deployment Dashboard dengan kunci dan, masing-masing. `MultimodalDataBucketName` `MultimodalDataMetadataTable`

> **Note**
>
> Tidak semua model mendukung input multimodal. Pastikan model yang Anda pilih mendukung pemrosesan gambar dan dokumen sebelum mengaktifkan fitur ini. Lihat model foundation yang didukung dalam dokumentasi Amazon Bedrock untuk memeriksa model mana yang mendukung Image sebagai modalitas input.

> **Important**
>
> File yang diunggah oleh pengguna disimpan di Amazon S3 dengan kebijakan siklus hidup 48 jam. Metadata tentang file yang diunggah disimpan di Amazon DynamoDB dengan TTL 24 jam untuk riwayat percakapan.

Tinjau dan lakukan deployment

Setelah langkah ini, tinjau pengaturan yang Anda pilih dan pilih Deploy Use Case. Kasus penggunaan baru kemudian menyebar dan menjadi terlihat di tampilan dasbor Deployment Anda untuk mengelola lebih lanjut.

Langkah 3b: Menyebarkan kasus penggunaan Agen Batuan Dasar

Kasus penggunaan Agen Bedrock menyediakan mekanisme yang kuat dan aman untuk memanggil Agen Bedrock Amazon dalam kasus penggunaan Anda. Fitur ini memungkinkan pengembang untuk mengintegrasikan kemampuan agen otonom bertenaga AI dengan mulus yang dapat mengatur dan melaksanakan tugas multi-langkah di berbagai model dasar, sumber data, aplikasi perangkat lunak, dan percakapan pengguna sambil mempertahankan langkah-langkah keamanan yang kuat.

Prasyarat

Sebelum membuat agen Amazon Bedrock, pastikan Anda memiliki yang berikut:

1. Akun AWS tempat Generative AI Application Builder di AWS diterapkan, dengan akses ke konsol Amazon Bedrock.
2. Izin IAM yang sesuai untuk membuat dan mengelola Agen Batuan Dasar Amazon.

Membuat Agen Batuan Dasar Amazon

Lihat agen Membuat dan mengonfigurasi secara manual di Panduan Pengguna Amazon Bedrock untuk petunjuk terperinci tentang cara membuat agen. Anda dapat mengonfigurasi opsi seperti:

- Instruksi (petunjuk) untuk agen Anda
- Basis pengetahuan, yang digunakan untuk mencari informasi tambahan berdasarkan masukan pengguna
- Memori agen untuk memungkinkan agen mengingat informasi di beberapa sesi (selama maksimal 30 hari)

Setelah berhasil membuat agen Amazon Bedrock, Anda dapat melanjutkan ke alur panduan kasus penggunaan Generative AI Application Builder di AWS Bedrock Agent. Untuk melakukannya, pilih Menerapkan kasus penggunaan baru di dasbor Deployment dan pilih Create Bedrock Agent Use Case. Ikuti wizard dan gunakan langkah-langkah berikut untuk mengonfigurasi kasus penggunaan.

Pilih kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Pilih konfigurasi jaringan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya

Pilih agen

Pada langkah ini, Anda harus memberikan ID Agen dan ID Alias dari agen Amazon Bedrock yang Anda buat.

Langkah 3c: Menyebarkan kasus penggunaan MCP Server

Kasus penggunaan Server MCP (Model Context Protocol) memungkinkan Anda untuk menyebarkan dan mengelola server MCP yang dapat diintegrasikan dengan model dan agen AI. Server MCP menyediakan cara standar untuk mengekspos alat, sumber daya, dan kemampuan ke aplikasi AI. Anda dapat membuat server MCP dari fungsi Lambda yang ada APIs dan, atau meng-host server MCP khusus menggunakan gambar kontainer.

Prasyarat

Sebelum menerapkan kasus penggunaan MCP Server, pastikan Anda memiliki yang berikut:

1. Akun AWS tempat Generative AI Application Builder di AWS diterapkan.
2. Izin IAM yang sesuai untuk membuat dan mengelola sumber daya Amazon Bedrock AgentCore .
3. Tergantung pada metode pembuatan yang Anda pilih:
   - Untuk metode Gateway (Lambda/API/MCPServer): Fungsi Lambda, titik akhir API dengan file skema yang sesuai (format JSON untuk Lambda OpenAPI/Smithy , APIs untuk), atau titik akhir URL Server MCP
   - Untuk metode Runtime (ECR): Gambar kontainer Docker didorong ke Amazon ECR yang berisi implementasi server MCP Anda

Metode pembuatan MCP Server

Solusinya mendukung dua metode untuk membuat server MCP:

Buat dari Lambda, API, atau MCP Server (metode Gateway)

Metode ini menciptakan gateway MCP yang membungkus fungsi Lambda yang ada, REST, atau server MCP eksternal APIs, membuatnya dapat diakses sebagai alat MCP. Gateway menangani terjemahan protokol antara MCP dan layanan Anda yang ada.

- Target Lambda: Integrasikan fungsi Lambda yang ada dengan menyediakan fungsi ARN dan file skema JSON yang menjelaskan format fungsi input/output

- Target OpenAPI: Integrasikan REST APIs menggunakan spesifikasi OpenAPI (format JSON atau YAMG) dengan dukungan untuk otentikasi 2.0 atau API Key OAuth
- Target Smithy: Integrasikan yang APIs ditentukan menggunakan file model Smithy (format.smithy atau .json)
- Target MCP Server: Connect langsung ke server MCP eksternal melalui endpoint URL, memungkinkan integrasi server MCP yang ada tanpa menggunakan infrastruktur baru

Anda dapat mengonfigurasi beberapa target (hingga 10) dalam satu gateway MCP, masing-masing mewakili alat atau kemampuan yang berbeda.

Hosting dari ECR Image (metode Runtime)

Metode ini menerapkan server MCP kontainer dari image Amazon ECR. Gunakan pendekatan ini ketika Anda memiliki implementasi server MCP kustom yang perlu dijalankan sebagai layanan mandiri.

- Berikan URI gambar ECR (harus menyertakan tag, misalnya, `:latest` atau`:v1.0.0`)
- Konfigurasikan variabel lingkungan secara opsional untuk meneruskan konfigurasi ke wadah Anda
- Wadah harus mengimplementasikan protokol MCP dan mengekspos titik akhir yang diperlukan

Menyebarkan Server MCP

Untuk menerapkan kasus penggunaan MCP Server, pilih Menerapkan kasus penggunaan baru di dasbor Deployment dan pilih Buat Kasus Penggunaan Server MCP. Ikuti wizard dan gunakan langkah-langkah berikut untuk mengonfigurasi kasus penggunaan.

Pilih kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Pilih konfigurasi jaringan

Saat ini hanya akses publik yang diaktifkan dan VPC tidak didukung untuk konfigurasi netwrok.

Buat MCP Server

Pada langkah ini, Anda mengonfigurasi penyebaran server MCP Anda:

Metode pembuatan server MCP

Pilih di antara dua metode pembuatan:

- Buat dari Lambda, API, atau MCP Server: Buat gateway MCP dari fungsi Lambda yang ada, spesifikasi API, atau titik akhir server MCP eksternal
- Hosting dari Gambar ECR: Menyebarkan server MCP khusus dari gambar kontainer

> **Note**
>
> Metode pembuatan tidak dapat diubah setelah penerapan. Jika Anda perlu beralih metode, Anda harus menerapkan kasus penggunaan MCP Server baru.

Konfigurasi Gateway (untuk metode Lambda/API/MCP Server)

Jika Anda memilih metode Gateway, konfigurasikan satu atau beberapa target:

1. Nama target (wajib): Nama ramah untuk mengidentifikasi konfigurasi target ini
2. Deskripsi target (opsional): Deskripsi singkat tentang apa yang dilakukan target ini
3. Jenis Target: Pilih jenis target yang akan dikonfigurasi:
   - Lambda: Untuk fungsi AWS Lambda
   - OpenAPI: Untuk REST dengan spesifikasi APIs OpenAPI
   - Smithy: Karena APIs dengan definisi model Smithy
   - MCP Server: Untuk koneksi langsung ke server MCP eksternal melalui titik akhir URL
4. File Skema (wajib): Unggah file skema yang menjelaskan target Anda:
   - Untuk Lambda: File skema JSON yang menjelaskan format. input/output Untuk detail tentang membuat skema alat Lambda, lihat Skema alat Lambda di Panduan Pengembang Amazon Bedrock. AgentCore
   - Untuk OpenAPI: File spesifikasi OpenAPI (JSON atau YAMG). Untuk detail tentang persyaratan skema OpenAPI, lihat skema OpenAPI di Panduan Pengembang Amazon Bedrock. AgentCore
   - Untuk Smithy: File model Smithy (.smithy atau .json). Untuk detail tentang membangun target Smithy, lihat Membangun target Smithy di Panduan Pengembang Amazon Bedrock. AgentCore
5. Fungsi Lambda ARN (diperlukan untuk target Lambda): ARN dari fungsi Lambda untuk mengintegrasikan
6. MCP Server URL (diperlukan untuk target MCP Server): Endpoint URL dari server MCP eksternal untuk terhubung. URL harus dikodekan dengan benar dan server MCP harus mendukung

kemampuan alat dengan protokol MCP versi 2025-06-18. Untuk informasi selengkapnya, lihat target server MCP di Panduan AgentCore Pengembang Amazon Bedrock.

7. Otentikasi Keluar (diperlukan untuk target OpenAPI): Konfigurasikan otentikasi untuk panggilan REST API:
   - Jenis Otentikasi: Pilih OAuth 2.0 atau Kunci API
   - Penyedia Auth Keluar ARN: ARN dari penyedia kredensi di brankas token Amazon Bedrock AgentCore
   - Konfigurasi tambahan: Tergantung pada jenis otentikasi:
     - Untuk OAuth 2.0: Konfigurasikan cakupan dan parameter khusus
     - Untuk Kunci API: Tentukan lokasi (parameter header atau kueri), nama parameter, dan awalan opsional

Anda dapat menambahkan beberapa target (hingga 10) dengan memilih Tambahkan target lain. Setiap target mewakili alat atau kemampuan terpisah yang diekspos oleh server MCP Anda.

Konfigurasi ECR (untuk metode Gambar ECR)

Jika Anda memilih metode Runtime, berikan:

1. URI Gambar ECR (wajib): URI lengkap gambar Docker Anda di Amazon ECR
   - Format: `account-id.dkr.ecr.region.amazonaws.com/repository-name:tag`
   - Gambar harus berada di Wilayah AWS yang sama dengan penerapan Anda
   - Sebuah tag diperlukan (misalnya,`:latest`,`:v1.0.0`)
2. Variabel lingkungan (opsional): Konfigurasikan pasangan nilai kunci untuk diteruskan ke wadah Anda saat runtime
   - Gunakan ini untuk menyediakan konfigurasi, kredensyal, atau bendera kustom
   - Anda dapat menambahkan hingga 10 variabel lingkungan

Tinjau dan lakukan deployment

Setelah mengonfigurasi server MCP Anda, tinjau pengaturan yang Anda pilih dan pilih Deploy Use Case. Kasus penggunaan Server MCP yang baru kemudian menyebar dan menjadi terlihat di tampilan dasbor Deployment Anda untuk pengelolaan lebih lanjut.

> **Note**
>
> Penerapan MCP Server membuat sumber daya di Amazon Bedrock AgentCore, termasuk gateway, runtime, dan identitas beban kerja. Sumber daya ini secara otomatis dikelola oleh solusi dan akan dibersihkan ketika Anda menghapus kasus penggunaan.

Langkah 3d: Menyebarkan kasus penggunaan Agent Builder

Agen Builder memungkinkan Anda membuat, mengonfigurasi, dan menerapkan agen AI siap produksi di Amazon Bedrock. AgentCore Fitur ini memberikan kontrol penuh atas perilaku agen melalui prompt sistem, pemilihan model, integrasi server MCP, dan manajemen memori.

Proses penyebaran terutama sama dengan kasus penggunaan Teks, dengan beberapa perbedaan penting.

Pilih kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Detail kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Konfigurasikan agen

Pada langkah ini, Anda mengonfigurasi pengaturan agen inti termasuk prompt sistem, servers/ Strands alat MCP yang tersedia, dan memori.

Prompt Sistem

Prompt sistem mendefinisikan perilaku, kepribadian, dan kemampuan agen. Anda dapat:

- Edit templat prompt sistem default
- Gunakan tombol Reset to default untuk mengembalikan template asli
- Sertakan instruksi untuk penggunaan alat dan pemformatan respons

Integrasi Server MCP (Opsional)

Konfigurasikan server Protokol Konteks Model untuk memberi agen Anda akses ke alat dan data perusahaan:

1. Pilih dari server MCP yang tersedia di dropdown
2. Tinjau alat yang tersedia di luar kotak yang akan dapat diakses oleh agen

> **Note**
>
> Server MCP harus dikonfigurasi dan dapat diakses sebelum penerapan. Lihat dokumentasi MCP untuk instruksi penyiapan server.

Konfigurasi Memori

Konfigurasikan bagaimana agen mempertahankan konteks dan pengetahuan:

- Memori Jangka Pendek: Diaktifkan secara default untuk semua agen. Mempertahankan konteks percakapan dalam sesi.
- Memori Jangka Panjang: Beralih untuk mengaktifkan ekstraksi dan penyimpanan wawasan di seluruh sesi. Menggunakan AgentCore Memori dengan strategi memori semantik.

Tinjau dan lakukan deployment

Setelah langkah ini, tinjau pengaturan yang Anda pilih dan pilih Deploy Use Case. Penyebaran Agent Builder biasanya selesai dalam 10-15 menit. Kasus penggunaan baru kemudian menjadi terlihat di tampilan dasbor Deployment Anda untuk mengelola lebih lanjut.

Langkah 3e: Menerapkan kasus penggunaan Alur Kerja

Workflow Builder memungkinkan Anda membuat agen pengawas yang mengatur beberapa agen Pembuat Agen menggunakan pola delegasi Agen sebagai Alat. Fitur ini memungkinkan Anda untuk membangun alur kerja multi-agen yang kompleks dengan menggunakan kembali penerapan Agent Builder yang ada.

Proses penyebaran mengikuti pola yang mirip dengan Agent Builder, dengan langkah-langkah tambahan untuk penemuan dan pemilihan agen.

Pilih kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Detail kasus penggunaan

Langkah ini sama dengan kasus penggunaan Teks yang dijelaskan sebelumnya.

Konfigurasikan agen pengawas

Pada langkah ini, Anda mengonfigurasi agen supervisor yang akan mengoordinasikan agen Agen Builder khusus.

Prompt Sistem

Prompt sistem mendefinisikan bagaimana delegasi agen supervisor bekerja kepada agen khusus. Anda dapat:

- Edit templat prompt sistem default
- Sertakan instruksi untuk pemilihan dan delegasi agen
- Tentukan cara mengumpulkan hasil dari beberapa agen
- Gunakan tombol Reset to default untuk mengembalikan template asli

> **Note**
>
> Prompt sistem harus menjelaskan dengan jelas kapan dan bagaimana menggunakan masing-masing agen khusus. Deskripsi agen sangat penting untuk pendelegasian yang tepat.

Pemilihan Model

Pilih model pondasi untuk agen supervisor. Agen supervisor menggunakan model ini untuk:

- Memahami permintaan pengguna
- Pilih agen khusus yang sesuai
- Mengkoordinasikan eksekusi agen
- Tanggapan agregat dan format

Pilih agen khusus

Pada langkah ini, Anda memilih agen Agen Pembangun mana yang dapat didelegasikan oleh supervisor.

Menambahkan Agen

1. Klik Tambahkan Agen untuk membuka dialog pemilihan agen
2. Pilih satu atau beberapa agen Agen Builder dari daftar
3. Tinjau deskripsi agen yang akan diberikan kepada supervisor
4. Konfirmasikan pilihan

> **Note**
>
> - Alur kerja memerlukan setidaknya 1 kasus penggunaan Agen Builder sebagai agen khusus
> - Semua agen khusus harus berhasil digunakan sebelum membuat alur kerja

Tinjau dan lakukan deployment

Tinjau konfigurasi alur kerja termasuk:

- Prompt dan model sistem agen pengawas
- Daftar agen khusus
- Pengaturan memori

Pilih Terapkan Kasus Penggunaan. Penyebaran Alur Kerja biasanya selesai dalam 15-20 menit. Alur kerja baru akan terlihat di tampilan dasbor Deployment Anda untuk mengelola lebih lanjut.

Langkah 4: Konfigurasi pasca-penyebaran

Bagian ini memberikan rekomendasi untuk mengonfigurasi solusi setelah penerapan.

Pembuatan versi bucket Amazon S3, kebijakan siklus hidup, dan replikasi lintas wilayah

Solusi ini tidak menerapkan konfigurasi siklus hidup pada bucket yang dibuatnya. Sebaiknya lakukan hal berikut:

- Menyetel konfigurasi siklus hidup untuk penerapan produksi. Untuk detailnya, lihat Menyetel konfigurasi siklus hidup pada bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Mengaktifkan pembuatan versi dan replikasi lintas wilayah untuk bucket Amazon S3 berdasarkan kasus penggunaan solusi yang digunakan.

Pencadangan Amazon DynamoDB

Solusi ini menggunakan DynamoDB untuk beberapa tujuan (lihat layanan AWS dalam solusi ini). Solusinya tidak mengaktifkan cadangan untuk tabel yang dibuatnya. Sebaiknya buat cadangan fitur ini untuk penerapan produksi. Lihat Mencadangkan tabel DynamoDB dan Menggunakan AWS Backup untuk DynamoDB untuk detailnya.

CloudWatch Dasbor dan alarm Amazon

Solusi ini menerapkan dasbor khusus CloudWatch untuk merender bagan dari metrik yang dipublikasikan khusus dan metrik layanan AWS. Sebaiknya buat CloudWatch alarm dan tambahkan notifikasi berdasarkan kasus penggunaan yang solusinya digunakan.

CloudWatch Log Amazon

Log Lambda dikonfigurasi agar tidak pernah kedaluwarsa dan log API Gateway dikonfigurasi dengan masa kadaluwarsa 10 tahun. Anda dapat memperbarui kedaluwarsa grup log masing-masing agar selaras dengan kebijakan penyimpanan catatan perusahaan Anda.

Domain web khusus dengan TLS v1.2 atau sertifikat yang lebih tinggi

Solusinya menggunakan UI web dan Edge Optimized API Gateway. CloudFront CloudFrontdomain tidak memberlakukan TLS v1.2 atau sertifikat yang lebih tinggi. Sebaiknya buat domain kustom menggunakan Amazon Route 53, membuat sertifikat menggunakan AWS Certificate Manager, atau menggunakan sertifikat yang ada jika organisasi Anda memilikinya.

Untuk detail tambahan, lihat Panduan Pengembang Amazon Route 53 dan Memilih versi TLS minimum untuk domain kustom di API Gateway.

Penskalaan dengan Amazon Kendra

Solusi ini memberikan kemampuan untuk menggunakan Amazon Kendra untuk melakukan pencarian cerdas bertenaga NLP di seluruh dokumen yang dicerna. Anda dapat meningkatkan kapasitas

Amazon Kendra menggunakan CloudFormation parameter berikut untuk beban kerja yang lebih besar:

Parameter	Default	Deskripsi
Amazon Kendra kapasitas kueri tambahan	`0`	Jumlah kapasitas kueri ekstra untuk indeks dan GetQuerySuggestionskapasitas. Unit kapasitas tambahan untuk indeks menyediakan sekitar 8.000 kueri per hari.
Amazon Kendra kapasitas penyimpanan tambahan	`0`	Jumlah kapasitas penyimpanan tambahan untuk indeks. Unit kapasitas tunggal menyediakan 30 GB ruang penyimpanan atau 100.000 dokumen, mana yang mencapai lebih dulu.
Edisi Amazon Kendra	`Developer`	Amazon Kendra menyediakan Edisi Pengembang dan Perusahaan untuk membuat indeks. Untuk informasi lebih lanjut tentang perbedaan antara Edisi Amazon Kendra, lihat harga Amazon Kendra.

Untuk memodifikasi nilai CloudFormation parameter ini, pilih nilai yang sesuai pada saat penyebaran tumpukan. Untuk informasi selengkapnya tentang unit kueri dan kapasitas penyimpanan, lihat Menyesuaikan kapasitas.

> **Note**
>
> Jika kasus penggunaan Teks tidak diterapkan dengan RAG diaktifkan, maka indeks Amazon Kendra tidak digunakan atau dibuat.

Menyiapkan SSO menggunakan federasi Idp

Solusi ini memungkinkan integrasi dengan penyedia identitas eksternal yang mendukung federasi identitas berbasis SAMP atau OIDC. Saat solusi diterapkan, solusi akan membuat kumpulan pengguna Amazon Cognito dan integrasi klien aplikasi individual untuk dasbor Deployment dan kasus penggunaan individual. Berdasarkan Idp eksternal, ikuti langkah-langkah yang disediakan di bagian Mengonfigurasi penyedia identitas untuk kumpulan pengguna Anda di Panduan Pengembang Amazon Cognito dan pilih integrasi klien aplikasi untuk dasbor Deployment atau kasus penggunaan yang ingin Anda setel dengan SSO.

Untuk meneruskan informasi grup pengguna ke basis pengetahuan atau penyimpanan vektor dalam arsitektur berbasis RAG, Anda perlu memetakan grup pengguna dari grup pengguna Idp eksternal ke Amazon Cognito. Solusinya menyediakan pemicu fungsi Lambda perancah awal untuk dipetakan dengan fase pembuatan token pra. Fungsi Lambda memiliki file group_mapping.json yang harus diperbarui untuk menyediakan pemetaan grup. Lihat Menyesuaikan alur kerja kumpulan pengguna dengan pemicu Lambda untuk pemicu Lambda yang didukung oleh Amazon Cognito.

Konfigurasi User Pool Manual

Jika Anda memilih untuk tidak meneruskan Admin atau email pengguna default selama penerapan, Anda harus membuat grup pengguna yang sesuai secara manual di Amazon Cognito untuk memastikan izin yang benar:

1. Untuk dasbor Deployment, buat grup bernama `Admin` di kumpulan pengguna Cognito Anda.
2. Untuk setiap kasus penggunaan, buat grup bernama `${UseCaseName}-Users` di kumpulan pengguna Cognito Anda, di mana `${UseCaseName}` nama kasus penggunaan yang Anda gunakan.

Kelompok-kelompok ini diperlukan agar mekanisme otorisasi berfungsi dengan benar. Setiap pengguna yang ingin Anda berikan akses harus ditambahkan ke grup yang sesuai.

Jika `placeholder@example.com` diteruskan, grup Cognito akan dibuat, tetapi Anda masih harus membuat pengguna terkait dan menetapkannya ke grup.

Menyesuaikan layar login

Solusi ini menggunakan UI yang dihosting Amazon Cognito untuk merender halaman login. Untuk menyesuaikan halaman login bawaan, lihat Menyesuaikan halaman web masuk dan pendaftaran bawaan di Panduan Pengembang Amazon Cognito.

Pertimbangan keamanan tambahan

Berdasarkan kasus penggunaan yang Anda gunakan solusinya, tinjau rekomendasi keamanan berikut:

- Kunci enkripsi AWS KMS yang dikelola pelanggan - Solusinya menggunakan kunci AWS KMS yang dikelola AWS secara default, karena ini tersedia tanpa biaya tambahan. Tinjau kasus penggunaan Anda untuk menentukan apakah Anda harus memperbarui solusi untuk menggunakan kunci AWS KMS yang dikelola pelanggan.
- Aturan pelambatan API Gateway - Solusinya diterapkan dengan aturan pembatasan default pada API Gateway. Berdasarkan kasus penggunaan dan volume transaksi yang diharapkan, kami menyarankan Anda mengonfigurasi pelambatan untuk. APIs Untuk detailnya, lihat Permintaan Throttle API untuk throughput yang lebih baik di Panduan Pengembang Amazon API Gateway.
- Mengaktifkan AWS CloudTrail - Sebagai praktik keamanan yang direkomendasikan, pertimbangkan untuk mengaktifkan AWS CloudTrail di akun AWS tempat solusi diterapkan untuk mencatat panggilan API di akun AWS. Untuk detailnya, lihat Panduan CloudTrail Pengguna AWS.
- Deteksi drift - Sebaiknya konfigurasi deteksi drift pada CloudFormation tumpukan untuk mengidentifikasi dan diberi tahu tentang perubahan yang tidak disengaja atau berbahaya pada tumpukan solusi yang diterapkan. Untuk detailnya, lihat Menerapkan alarm untuk mendeteksi drift di CloudFormation tumpukan AWS secara otomatis.
- Cognito JSON Web Tokens (JWTs) - Solusinya menggunakan Amazon Cognito yang dikeluarkan JWTs untuk mengautentikasi dengan titik akhir REST API. Kami mengonfigurasi solusi dengan kedaluwarsa lima menit untuk token ID dan token akses. Saat pengguna log out, kemampuan mereka untuk menghasilkan token baru dicabut (token penyegaran dicabut). Namun, hingga token saat ini berakhir, setiap permintaan ke titik akhir API akan berhasil diautentikasi, karena mereka memiliki token yang valid. Tinjau pertimbangan keamanan untuk kasus penggunaan Anda dan sesuaikan periode validitas token.

Menyesuaikan kebijakan siklus hidup:

Untuk penerapan produksi, tinjau dan sesuaikan kebijakan siklus hidup berdasarkan persyaratan retensi Anda. Lihat Menyetel konfigurasi siklus hidup pada bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Penyimpanan file multimodal dan siklus hidup

Jika Anda mengaktifkan kemampuan input multimodal (MultimodalEnableddisetel ke`Yes`) untuk kasus penggunaan, solusinya akan membuat bucket Amazon S3 untuk menyimpan file yang diunggah dan tabel DynamoDB untuk melacak metadata file.

Kebijakan siklus hidup default:

- File S3: Secara otomatis dihapus setelah 48 jam
- Metadata DynamoDB: Rekaman kedaluwarsa setelah 24 jam (riwayat percakapan TTL)

Pertimbangan keamanan:

- File dipartisi berdasarkan ID kasus penggunaan, ID pengguna, ID percakapan dan ID pesan dan file disimpan dengan nama UUID sebagai gantinya. Pemetaan untuk UUID ke nama file tersedia di tabel metadata DynamoDB
- Pengguna hanya dapat mengakses file yang mereka unggah dalam percakapan mereka sendiri
- Validasi tipe file dilakukan dengan menggunakan deteksi angka ajaib
- Sebaiknya aktifkan Amazon GuardDuty Malware Protection for S3 untuk memindai file yang diunggah dari konten berbahaya

Menerapkan kasus penggunaan Teks mandiri

Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menyebarkan solusi ke akun Anda.

Waktu untuk menyebarkan: Sekitar 10-30 menit

1. Masuk ke AWS Management Console dan pilih tombol untuk meluncurkan CloudFront template yang ingin Anda terapkan.

BedrockChat.template	
SageMakerChat.template	

2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di Wilayah AWS yang berbeda, gunakan pemilih Wilayah di bilah navigasi konsol.

   Catatan: Solusi ini menggunakan Amazon Kendra dan Amazon Bedrock, yang saat ini tidak tersedia di semua Wilayah AWS. Jika menggunakan fitur-fitur ini, Anda harus meluncurkan solusi ini di Wilayah AWS tempat layanan ini tersedia. Untuk ketersediaan terbaru menurut Wilayah, lihat Daftar Layanan Regional AWS.
3. Pada halaman Create stack *, verifikasi bahwa URL template yang benar ada di kotak teks*Amazon S3 URL *dan pilih *Next.
4. Pada halaman *Tentukan detail tumpukan*, tetapkan nama ke tumpukan solusi Anda. Untuk informasi tentang batasan penamaan karakter, lihat Batas IAM dan STS di Panduan Pengguna AWS Identity and Access Management.
5. Di bawah Parameter, tinjau parameter untuk templat solusi ini dan modifikasi sesuai kebutuhan. Solusi ini menggunakan nilai default berikut.

UseCaseUUID	*<_Requires input_>*	Panjang 36 karakter UUIDv4 untuk mengidentifikasi kasus penggunaan yang diterapkan ini dalam suatu aplikasi.
UseCaseConfigRecordKey	*<_Requires input_>*	Kunci yang sesuai dengan catatan yang berisi konfigurasi yang diperlukan oleh penyedia obrolan Lambda saat runtime. Catatan dalam tabel harus memiliki atribut kunci yang cocok dengan nilai ini, dan atribut config yang berisi konfigurasi yang diinginkan. Catatan ini akan diisi oleh platform penyebaran jika digunakan. Untuk penerapan mandiri dari kasus penggunaan ini, diperlukan entri yang dibuat secara manual dalam tabel yang

		ditentukan. UseCaseConfigTableName
UseCaseConfigTableName	*<_Requires input_>*	Tumpukan akan membaca konfigurasi dari tabel dengan nama ini di kuncinya UseCaseConfigRecordKey

ExistingRestApiId	(Masukan opsional)	ID API Gateway REST API yang ada untuk digunakan . Jika tidak disediakan, API API Gateway REST API baru akan dibuat. Biasanya disediakan saat menerapkan dari dasbor Deployment. Catatan: Menggunak an Existing APIs dapat membantu mengurangi duplikasi sumber daya dan menyederhanakan pengelola an APIs kapan Anda perlu menerapkan beberapa kasus penggunaan mandiri. Saat menyediakan yang ada APIs untuk kasus penggunaan mandiri, Anda bertanggu ng jawab untuk memastika n bahwa API dikonfigurasi dengan rute yang diperluka n dengan model yang diharapkan. Rute /detail pra-konfigurasi yang diperluka n (mengambil detail kasus penggunaan selama obrolan) dan secara opsional, rute / umpan balik (jika FeedbackE nableddiatur untuk mengaktif kan pengumpulan umpan balik `Yes` untuk respons obrolan LLM) harus dikonfigu rasi. Selain itu ExistingA piRootResourceId,, ExistingC ognitoUserPoolIddan juga

		ExistingCognitoGroupPolicyTableNameharus disediakan.
ExistingApiRootResourceId	(Masukan opsional)	API Gateway REST API Root Resource ID yang ada untuk digunakan. REST API Root Resource ID dapat diperoleh dari konsol AWS dengan memilih sumber daya root (/) di bagian “Sumber Daya” dari API. Resource ID kemudian akan ditampilkan di panel Resource details. Sebagai alternatif, Anda dapat menjalankan panggilan describe API di REST API untuk menemukan ID Sumber Daya Root.
FeedbackEnabled	No	Jika disetel ke Tidak, tumpukan kasus penggunaan yang diterapkan tidak akan memiliki akses ke fitur umpan balik.
ExistingModelInfoTableName	(Masukan opsional)	Nama tabel DynamoDB untuk tabel yang berisi info model dan default. Digunakan oleh platform penyebaran. Jika dihilangkan, tabel baru akan dibuat untuk menampung default model.

DefaultUserEmail	`placeholder@example.com`	Email pengguna default untuk kasus penggunaan ini. Pengguna Amazon Cognito untuk email ini dibuat untuk mengakses kasus penggunaan. Jika tidak disediakan, Grup dan Pengguna Cognito tidak akan dibuat. Anda juga dapat menggunakan `placeholder@example.com` untuk membuat Grup tetapi bukan Pengguna. Lihat Konfigurasi Kumpulan Pengguna Manual untuk informasi tentang pengaturan kumpulan pengguna Anda.
ExistingCognitoUserPoolId	(Masukan opsional)	UserPoolIddari kumpulan pengguna Amazon Cognito yang ada yang akan diautentikasi dengan kasus penggunaan ini. Biasanya disediakan saat menerapkan dari dasbor Deployment, tetapi dapat dihilangkan saat menerapkan tumpukan kasus penggunaan ini secara mandiri.
CognitoDomainPrefix	(Masukan opsional)	Masukkan nilai jika Anda ingin memberikan domain untuk Klien Kumpulan Pengguna Cognito. Jika Anda tidak memberikan nilai, penerapan akan menghasilkannya.

ExistingCognitoUserPoolClient	(Masukan opsional)	Menyediakan Klien Kumpulan Pengguna (Klien Aplikasi) untuk menggunakan yang sudah ada. Jika Anda tidak menyediakan Klien Kumpulan Pengguna, yang baru akan dibuat. Parameter ini hanya dapat diberikan jika User Pool Id yang ada disediakan.
ExistingCognitoGroupPolicyTableName	(Masukan opsional)	Nama tabel DynamoDB yang berisi kebijakan grup pengguna. Ini digunakan oleh otorisasi khusus pada API kasus penggunaan. Biasanya, Anda dapat memberikan input saat menerapkan dari platform penerapan, tetapi dapat dihilangkan saat menerapkan tumpukan kasus penggunaan ini secara mandiri.
RAGEnabled	`true`	Jika disetel ke true, tumpukan kasus penggunaan yang diterapkan menggunakan indeks Amazon Kendra yang disediakan yang dibuat untuk menyediakan fungsionalitas RAG. Jika diatur ke`false`, pengguna berinteraksi langsung dengan LLM.

KnowledgeBaseType	Batuan dasar	Tipe dasar pengetahuan yang akan digunakan untuk RAG. Hanya atur jika RAGEnable d ada`true`. Bisa berupa Bedrock atau Kendra. Catatan: Hanya relevan jika RAGEnabledbenar.
ExistingKendraIndexId	(Masukan opsional)	ID indeks indeks Kendra yang ada untuk digunakan untuk kasus penggunaan. Jika tidak ada yang Knowledge BaseTypedisediakan dan Kendra, indeks baru akan dibuat untuk Anda. Catatan: Hanya relevan jika RAGEnabledada `true` dan Knowledge BaseTypeadalah`Kendra`.
NewKendraIndexName	(Masukan opsional)	Nama untuk indeks Kendra baru yang akan dibuat untuk kasus penggunaan ini. Hanya berlaku jika ExistingK endraIndexIdtidak disediakan. Catatan: Hanya relevan jika RAGEnabledbenar dan KnowledgeBaseTypeKendra.

NewKendraQueryCapacityUnits	0	Unit kapasitas kueri tambahan untuk indeks Amazon Kendra baru yang akan dibuat untuk kasus penggunaan ini. Hanya berlaku jika ExistingKendraIndexIdtidak disediakan, lihat CapacityUnitsConfiguration. Catatan: Hanya relevan jika RAGEnabledada `true` dan Knowledge BaseTypeadalah`Kendra`.
NewKendraStorageCapacityUnits	`0`	Unit kapasitas penyimpanan tambahan untuk indeks Amazon Kendra baru yang akan dibuat untuk kasus penggunaan ini. Hanya berlaku jika ExistingKendraIndexIdtidak disediakan, lihat CapacityUnitsConfiguration. Catatan: Hanya relevan jika RAGEnabledada `true` dan Knowledge BaseTypeadalah`Kendra`.

NewKendraIndexEdition	(Masukan opsional)	Edisi Amazon Kendra yang akan digunakan untuk indeks Amazon Kendra baru yang akan dibuat untuk kasus penggunaan ini. Hanya berlaku jika tidak ExistingKendraIndexIddisediakan, lihat Edisi Amazon Kendra. Catatan: Hanya relevan jika RAGEnabledada `true` dan Knowledge BaseTypeadalah`Kendra`.
BedrockKnowledgeBaseId	(Masukan opsional)	ID basis pengetahuan batuan dasar untuk digunakan dalam kasus penggunaan RAG. Tidak dapat diberikan jika ExistingKendraIndexIdatau NewKendraIndexName disediakan. Catatan: Hanya relevan jika RAGEnabledada `true` dan Knowledge BaseTypeadalah`Bedrock`.
VpcEnabled	No	Haruskah sumber daya tumpukan digunakan dalam VPC.

CreateNewVpc	No	PilihYes, jika Anda ingin solusi untuk membuat VPC baru untuk Anda dan digunakan untuk kasus penggunaan ini. Catatan: Hanya relevan jika VpcEnabledadaYes.
IPAMPoolId	(Masukan opsional)	Jika Anda ingin menetapkan rentang CIDR menggunakan Amazon VPC IP Address Manager, berikan Id kolam IPAM untuk digunakan. Catatan: Hanya relevan jika VpcEnabledada Yes dan CreateNewVpcadalahNo.
ExistingVpcId	(Masukan opsional)	ID VPC dari VPC yang ada untuk digunakan untuk kasus penggunaan. Catatan: Hanya relevan jika VpcEnabledada Yes dan CreateNewVpcadalahNo.
ExistingPrivateSubnetIds	(Masukan opsional)	Daftar subnet yang dipisahkan koma dari subnet IDs pribadi yang ada untuk digunakan untuk menyebarkan fungsi Lambda. Catatan: Hanya relevan jika VpcEnabledada Yes dan CreateNewVpcadalahNo.

ExistingSecurityGroupIds	(Masukan opsional)	Daftar grup keamanan yang dipisahkan koma dari VPC yang ada yang akan digunakan untuk mengkonfigurasi fungsi Lambda. Catatan: Hanya relevan jika VpcEnabledada `Yes` dan CreateNewVpcadalah`No`.
VpcAzs	(Masukan opsional)	Daftar dipisahkan koma AZs di mana subnet dibuat VPCs Catatan: Hanya relevan jika VpcEnabledada `Yes` dan CreateNewVpcadalah`No`.
UseInferenceProfile	`No`	Jika model yang dikonfigurasi adalah Bedrock, Anda dapat menunjukkan apakah Anda menggunakan Profil Inferensi Batuan Dasar. Ini akan memastikan bahwa kebijakan IAM yang diperlukan akan dikonfigurasi selama penerapan tumpukan. Untuk lebih jelasnya, lihat https://docs.aws.amazon.com/bedrock/ latest/userguide/cross -region-inference.html berikut

DeployUI	Ya	Pilih opsi untuk menerapkan UI frontend untuk penerapan ini. Memilih Tidak, hanya akan membuat infrastru ktur untuk meng-host APIs, otentikasi untuk APIs, dan pemrosesan backend.

6. Pilih Berikutnya.
7. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Pilih kotak yang menyatakan bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
9. Pilih Membuat tumpukan untuk menerapkannya.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu sekitar 10-30 menit.

Menerapkan kasus penggunaan Agen Batuan Dasar mandiri

Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menyebarkan solusi ke akun Anda.

Waktu untuk menyebarkan: Sekitar 10-30 menit

1. Masuk ke AWS Management Console dan pilih tombol untuk meluncurkan CloudFront template.

BedrockAgent.template	Launch solution

2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi di Wilayah AWS yang berbeda, gunakan pemilih Wilayah di bilah navigasi konsol.

> **Note**
>
> Solusi ini menggunakan Amazon Bedrock, yang saat ini tidak tersedia di semua Wilayah AWS. Jika Anda menggunakan fitur-fitur ini, Anda harus meluncurkan solusi ini di Wilayah

AWS tempat layanan ini tersedia. Untuk ketersediaan terbaru menurut Wilayah, lihat Daftar Layanan Regional AWS.

3. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ada di kotak teks URL Amazon S3 dan pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda. Untuk informasi tentang batasan penamaan karakter, lihat {https---docs-aws-amazon-com- https---docs-aws-amazon-com -IAM-latest- UserGuide -reference-iam-limits-html} [kuota IAM dan AWS STS] di Panduan Pengguna AWS Identity and Access Management.
5. Di bawah Parameter, tinjau parameter untuk templat solusi ini dan modifikasi sesuai kebutuhan. Solusi ini menggunakan nilai default berikut.

Parameter	Entri default	Deskripsi
UseCaseUUID	*<_Requires input_>*	Panjang 36 karakter UUIDv4 untuk mengidentifikasi kasus penggunaan yang diterapkan ini dalam suatu aplikasi.
UseCaseConfigRecordKey	*<Requires input>*	Kunci yang sesuai dengan catatan yang berisi konfigurasi yang diperlukan oleh penyedia obrolan fungsi Lambda saat runtime. Catatan dalam tabel harus memiliki atribut kunci yang cocok dengan nilai ini, dan atribut config yang berisi konfigurasi yang diinginkan. Catatan ini akan diisi oleh platform penyebaran jika sedang digunakan. Untuk penerapan mandiri dari kasus penggunaan ini, diperlukan entri yang dibuat secara

Parameter	Entri default	Deskripsi
		manual dalam tabel yang ditentukan. UseCaseConfigTableName
UseCaseConfigTableName	`<Requires input>`	Tumpukan akan membaca konfigurasi kasus penggunaan dari tabel yang disediakan di sini dan menggunakan kunci catatan yang ditentukan dalam UseCaseConfigRecordKey.
DefaultUserEmail	`placeholder@example.com`	Email pengguna default untuk kasus penggunaan ini. Solusinya membuat pengguna Amazon Cognito untuk email ini untuk mengakses kasus penggunaan.

Parameter	Entri default	Deskripsi
ExistingRestApiId	(Masukan opsional)	ID API Gateway REST API yang ada untuk digunakan . Jika tidak disediakan, API API Gateway REST API baru akan dibuat. Biasanya disediakan saat menerapkan dari dasbor Deployment. Catatan: Menggunak an Existing APIs dapat membantu mengurangi duplikasi sumber daya dan menyederhanakan pengelola an APIs kapan Anda perlu menerapkan beberapa kasus penggunaan mandiri. Saat menyediakan yang ada APIs untuk kasus penggunaan mandiri, Anda bertanggu ng jawab untuk memastika n bahwa API dikonfigurasi dengan rute yang diperluka n dengan model yang diharapkan. Rute /detail pra-konfigurasi yang diperluka n (mengambil detail kasus penggunaan selama obrolan) dan secara opsional, rute / umpan balik (jika FeedbackE nableddiatur untuk mengaktif kan pengumpulan umpan balik `Yes` untuk respons obrolan LLM) harus dikonfigu rasi. Selain itu ExistingA piRootResourceId,, ExistingC

Parameter	Entri default	Deskripsi
		ognitoUserPoolIddan juga ExistingCognitoGroupPolicyTableNameharus disediakan.
ExistingApiRootResourceId	(Masukan opsional)	API Gateway REST API Root Resource ID yang ada untuk digunakan. REST API Root Resource ID dapat diperoleh dari konsol AWS dengan memilih sumber daya root (/) di bagian “Sumber Daya” API. ID Sumber Daya kemudian akan ditampilkan di panel Rincian sumber daya. Sebagai alternatif, Anda dapat menjalankan panggilan describe API di REST API untuk menemukan ID Sumber Daya Root.
FeedbackEnabled	No	Jika disetel ke Tidak, tumpukan kasus penggunaan yang diterapkan tidak akan memiliki akses ke fitur umpan balik.
CognitoDomainPrefix	(Masukan opsional)	Masukkan nilai jika Anda ingin memberikan domain untuk klien kumpulan pengguna Amazon Cognito. Jika Anda tidak memberikan nilai, solusinya menghasilkan satu.

Parameter	Entri default	Deskripsi
ExistingCognitoUserPoolId	(Masukan opsional)	UserPoolIddari kumpulan pengguna Amazon Cognito yang ingin Anda autentikasi dengan kasus penggunaan ini. CATATAN: Anda biasanya memberikan ID ini saat menerapkan dari dasbor Deployment, tetapi Anda dapat menghilangkannya saat menerapkan tumpukan kasus penggunaan ini secara mandiri.
ExistingCognitoUserPoolClient	(Masukan opsional)	Menyediakan klien kumpulan pengguna (klien aplikasi) untuk menggunakan yang sudah ada. Jika Anda tidak menyediakan klien kumpulan pengguna, solusinya akan membuatnya. Anda hanya dapat memberikan parameter ini jika Anda memberikan file ExistingCognitoUserPoolId.

Parameter	Entri default	Deskripsi
ExistingCognitoGroupPolicyTableName	(Masukan opsional)	Nama tabel DynamoDB yang berisi kebijakan grup pengguna. Ini digunakan oleh otorisasi khusus pada API kasus penggunaan. CATATAN: Anda biasanya memberikan nama ini saat menerapkan dari dasbor Deployment, tetapi Anda dapat menghilangkannya saat menerapkan tumpukan kasus penggunaan ini secara mandiri.
VpcEnabled	No	Apakah sumber daya tumpukan digunakan dalam VPC.
CreateNewVpc	No	Pilih Yes apakah Anda ingin solusi untuk membuat VPC baru untuk Anda dan menggunakannya untuk kasus penggunaan ini. CATATAN: Parameter ini hanya relevan jika VpcEnable dadaYes.
IPAMPoolId	(Masukan opsional)	Jika Anda ingin menetapkan rentang CIDR menggunakan IPAM, berikan ID kolam IPAM untuk digunakan. CATATAN: Parameter ini hanya relevan jika VpcEnabledada Yes dan CreateNewVpcsedangNo.

Parameter	Entri default	Deskripsi
ExistingVpcId	(Masukan opsional)	ID VPC dari VPC yang ada untuk digunakan untuk kasus penggunaan. CATATAN: Parameter ini hanya relevan jika VpcEnabledada `Yes` dan CreateNewVpcsedangNo.
ExistingPrivateSubnetIds	(Masukan opsional)	Daftar subnet yang dipisahkan koma dari subnet IDs pribadi yang ada untuk digunakan untuk menyebarkan fungsi Lambda. CATATAN: Parameter ini hanya relevan jika VpcEnable dada `Yes` dan CreateNew VpcsedangNo.
ExistingSecurityGroupIds	(Masukan opsional)	Daftar grup keamanan yang dipisahkan koma dari VPC yang ada yang akan digunakan untuk mengonfig urasi fungsi Lambda. CATATAN: Parameter ini hanya relevan jika VpcEnable dada `Yes` dan CreateNew VpcsedangNo.
VpcAzs	(Masukan opsional)	Daftar dipisahkan koma AZs di mana subnet dibuat VPCs Catatan: Hanya relevan jika VpcEnabledada `Yes` dan CreateNewVpcadalahNo.
BedrockAgentId	*`<Requires input>`*	ID Agen Bedrock Amazon yang akan digunakan.

Parameter	Entri default	Deskripsi
BedrockAgentAliasId	*<Requires input>*	ID alias dari Amazon Bedrock Agent yang akan digunakan.
DeployUI	Yes	Pilih opsi untuk menerapkan UI obrolan frontend untuk penerapan ini. Memilih No hasil dalam membuat infrastruktur untuk meng-host APIs, otentikasi untuk APIs, dan pemrosesan backend tanpa UI obrolan.

6. Pilih Berikutnya.
7. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Pilih kotak yang mengakui bahwa template akan membuat sumber daya IAM.
9. Pilih Membuat tumpukan untuk menerapkannya.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu sekitar 10-30 menit.

Menyediakan konfigurasi obrolan DynamoDB

Saat menerapkan kasus penggunaan, UseCaseConfigRecordKeydan UseCaseConfigTableNamemerupakan CloudFormation parameter yang diperlukan yang biasanya diisi oleh dasbor Deployment. Tumpukan dasbor penerapan menangani pembuatan dan konfigurasi tabel ini, sementara panggilan ke API penerapan memicu populasi parameter.

Saat melakukan penerapan mandiri, Anda harus melakukan hal berikut:

1. Buat tabel DynamoDB dengan kunci hash kunci.
2. Buat catatan dalam tabel yang berisi konfigurasi untuk kasus penggunaan sebagai catatan format: `{key: some_use_case_key, config: {your_configuration}.`

3. Teruskan parameter yang dipilih UseCaseConfigTableNamedan UseCaseConfigRecordKey(`some_use_case_key`dalam contoh ini) ke tumpukan kasus penggunaan saat menerapkan.

Untuk membuat konfigurasi yang sesuai untuk penerapan mandiri, Anda dapat membuat kasus penggunaan yang diperlukan dari dasbor Deployment, dan menyalin catatan dari tabel konfigurasi. Jika tidak, Anda dapat membuat konfigurasi Anda sendiri berdasarkan contoh berikut untuk penerapan Bedrock:

```
{
 "UseCaseName": "SampleUseCase",
 "ConversationMemoryParams": {
 "ConversationMemoryType": "DynamoDB",
 "HumanPrefix": "H",
 "AiPrefix": "A",
 "ChatHistoryLength": 20
 },
 "KnowledgeBaseParams": {
 "KnowledgeBaseType": "Bedrock",
 "NumberOfDocs": 2,
 "ScoreThreshold": 0,
 "ReturnSourceDocs": false,
 "BedrockKnowledgeBaseParams": {
 "BedrockKnowledgeBaseId": "SOME_ID",
 "OverrideSearchType": null
 }
 },
 "LlmParams": {
 "ModelProvider": "Bedrock",
 "BedrockLlmParams": { "ModelId": "anthropic.claude-v2" },
 "PromptParams": {
 "PromptTemplate": "some prompt",
 "MaxPromptTemplateLength": 187500,
 "MaxInputTextLength": 187500,
 "UserPromptEditingEnabled": true,
 "DisambiguationEnabled": true,
 "DisambiguationPromptTemplate": "some prompt"
 },
 "ModelParams": {},
 "Temperature": 1,
 "RAGEnabled": true,
 "Streaming": true,
```

```
 "Verbose": false
 }
}
```

Pantau solusinya dengan Service Catalog AppRegistry

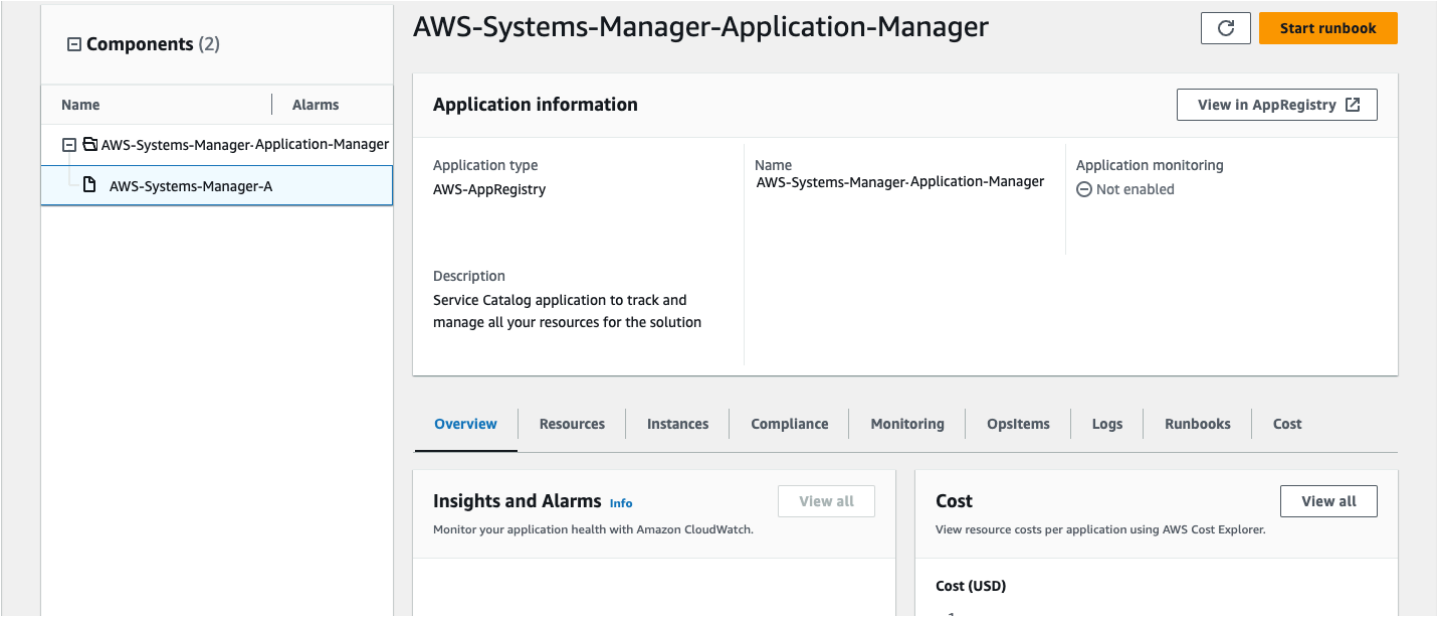
Solusinya mencakup AppRegistry sumber daya Service Catalog untuk mendaftarkan CloudFormation template dan sumber daya yang mendasarinya sebagai aplikasi di Service Catalog AppRegistry dan Systems Manager Application Manager.

Systems Manager Application Manager memberi Anda tampilan tingkat aplikasi ke dalam solusi ini dan sumber dayanya sehingga Anda dapat:

- Pantau sumber dayanya, biaya untuk sumber daya yang diterapkan di seluruh tumpukan dan akun AWS, dan log yang terkait dengan solusi ini dari lokasi pusat.
- Lihat data operasi untuk sumber daya solusi ini dalam konteks aplikasi. Misalnya, status penerapan, CloudWatch alarm, konfigurasi sumber daya, dan masalah operasional.

Gambar berikut menggambarkan contoh tampilan aplikasi untuk tumpukan solusi di Application Manager.

Menggambarkan tumpukan solusi di Manajer Aplikasi



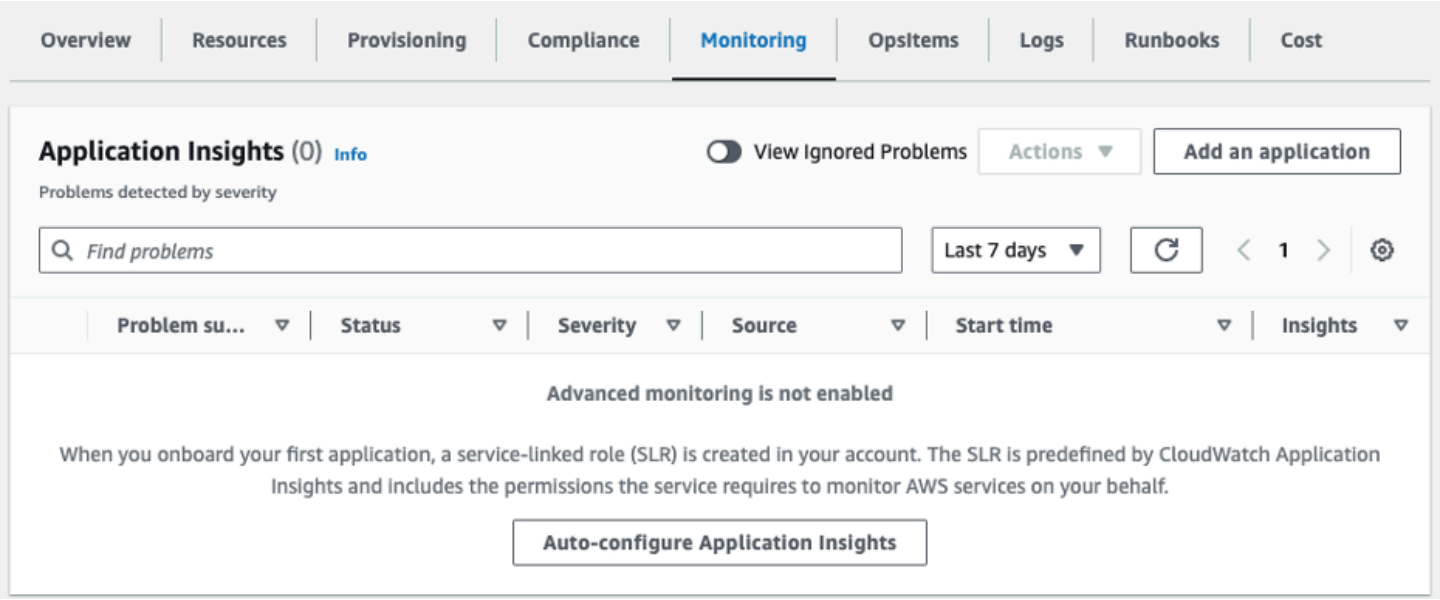
Aktifkan Wawasan CloudWatch Aplikasi

1. Masuk ke konsol Systems Manager.
2. Pada panel navigasi, pilih Manajer Aplikasi.

3. Di Aplikasi, cari nama aplikasi untuk solusi ini dan pilih.

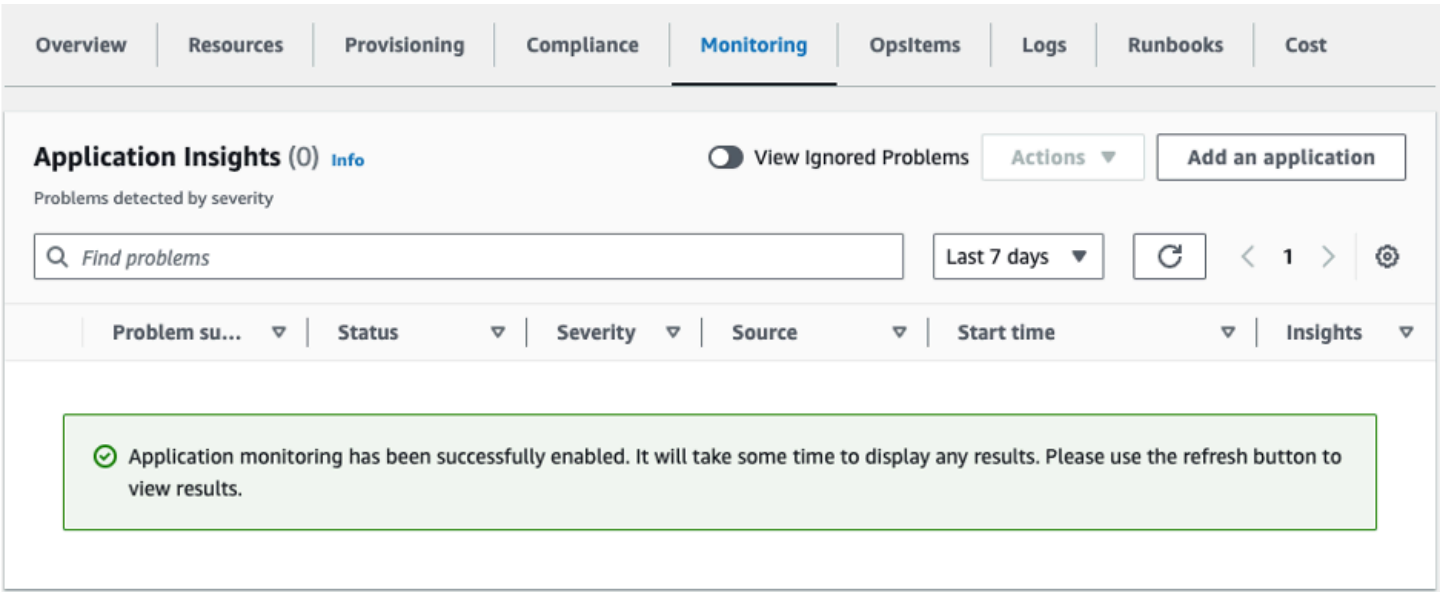
   Nama aplikasi akan memiliki App Registry di kolom Sumber Aplikasi, dan akan memiliki kombinasi nama solusi, Wilayah, ID akun, atau nama tumpukan.
4. Di pohon Komponen, pilih tumpukan aplikasi yang ingin Anda aktifkan.
5. Di tab Monitoring, di Application Insights, pilih Konfigurasi Otomatis Wawasan Aplikasi.

Dasbor Application Insights tidak menunjukkan masalah yang terdeteksi dan opsi untuk mengkonfigurasi otomatis.



Pemantauan untuk aplikasi Anda sekarang diaktifkan dan kotak status berikut muncul:

Dasbor Application Insights yang menunjukkan pesan aktivasi pemantauan yang berhasil.



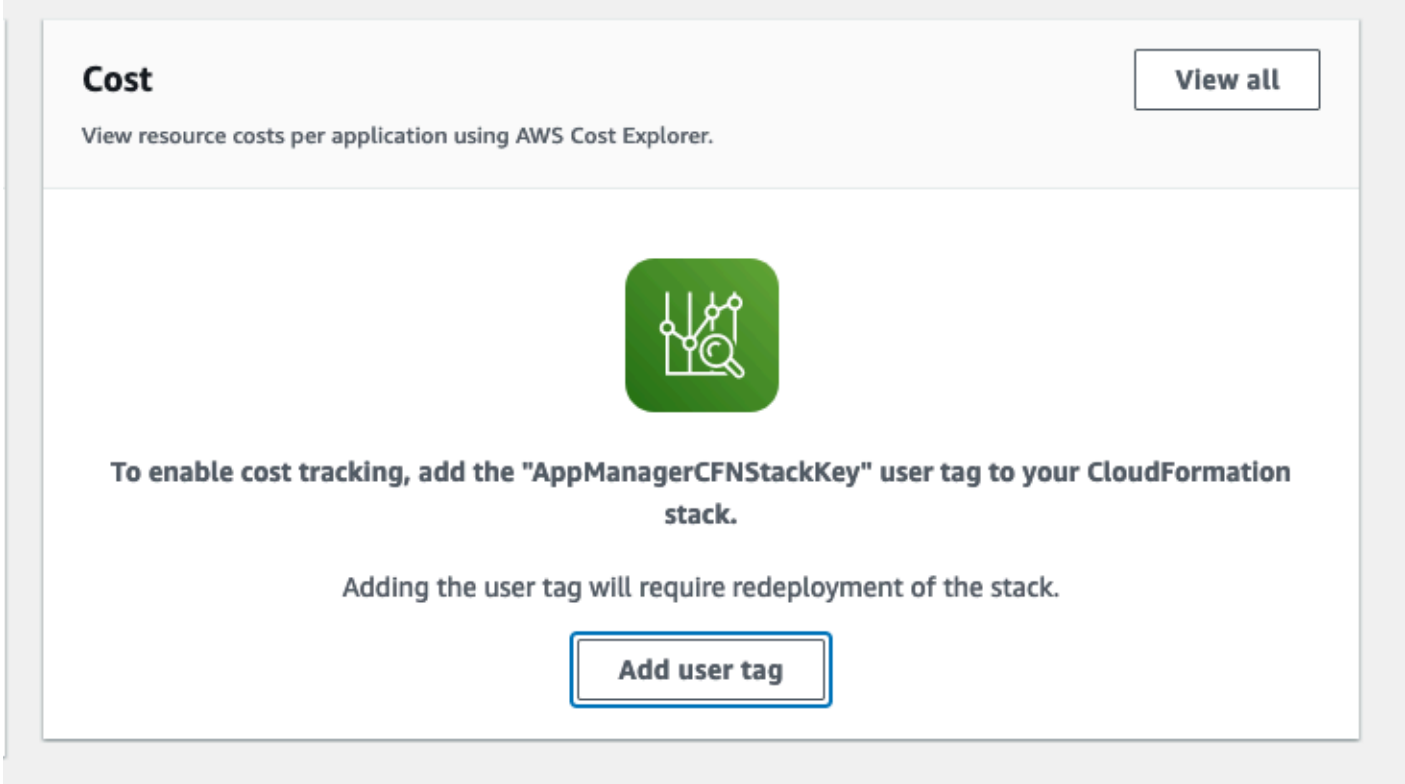
Konfirmasikan tag biaya yang terkait dengan solusi

Setelah Anda mengaktifkan tag alokasi biaya yang terkait dengan solusi, Anda harus mengonfirmasi tag alokasi biaya untuk melihat biaya untuk solusi ini. Untuk mengonfirmasi tag alokasi biaya:

1. Masuk ke konsol Systems Manager.
2. Pada panel navigasi, pilih Manajer Aplikasi.
3. Di Aplikasi, pilih nama aplikasi untuk solusi ini dan pilih.

   Nama aplikasi akan memiliki App Registry di kolom Sumber Aplikasi, dan akan memiliki kombinasi nama solusi, Wilayah, ID akun, atau nama tumpukan.
4. Di tab Ikhtisar, di Biaya, pilih Tambahkan tag pengguna.

   Screenshot yang menggambarkan layar Application Cost add user tag



5. Pada halaman Tambahkan tag pengguna, masukkan`confirm`, lalu pilih Tambahkan tag pengguna.

Proses aktivasi dapat memakan waktu hingga 24 jam untuk menyelesaikan dan data tag muncul.

Aktifkan tag alokasi biaya yang terkait dengan solusi

Setelah Anda mengaktifkan Cost Explorer, Anda harus mengaktifkan tag alokasi biaya yang terkait dengan solusi ini untuk melihat biaya untuk solusi ini. Tag alokasi biaya hanya dapat diaktifkan dari akun manajemen untuk organisasi. Untuk mengaktifkan tag alokasi biaya:

1. Masuk ke konsol AWS Billing and Cost Management dan Cost Management dan Cost Management.
2. Di panel navigasi, pilih Tag Alokasi Biaya.
3. Pada halaman Tag alokasi biaya, filter untuk tag AppManager CFNStack Kunci, lalu pilih tag dari hasil yang ditampilkan.
4. Pilih Aktifkan.

AWS Cost Explorer

Anda dapat melihat ikhtisar biaya yang terkait dengan komponen aplikasi dan aplikasi dalam konsol Application Manager melalui integrasi dengan AWS Cost Explorer, yang harus diaktifkan terlebih dahulu. Cost Explorer membantu Anda mengelola biaya dengan memberikan tampilan biaya dan penggunaan sumber daya AWS Anda dari waktu ke waktu. Untuk mengaktifkan Cost Explorer untuk solusinya:

1. Masuk ke konsol AWS Cost Management.
2. Di panel navigasi, pilih Cost Explorer untuk melihat biaya dan penggunaan solusi dari waktu ke waktu.

Perbarui solusinya

Jika sebelumnya Anda telah menerapkan solusi, ikuti prosedur ini untuk memperbarui CloudFormation tumpukan solusi untuk mendapatkan fitur dan penyempurnaan terbaru. Ada tiga bagian untuk proses upgrade:

- Langkah 1: Perbarui dasbor Deployment
- Langkah 2: Migrasikan konfigurasi kasus penggunaan
- Langkah 3: Perbarui kasus penggunaan

> **Note**
>
> 1. Di v2.0.0, integrasi dengan Anthropic dan Hugging Face tidak digunakan lagi demi Amazon Bedrock dan Amazon AI. SageMaker Anda dapat menggunakan model yang tersedia melalui Hugging Face melalui. SageMaker JumpStart Lihat Gunakan Hugging Face dengan SageMaker Amazon AI untuk detail selengkapnya.
> 2. Pastikan Anda menguji proses pembaruan di lingkungan non-produksi sebelum menjalankan langkah-langkah ini.

Langkah 1: Perbarui dasbor Deployment

1. Masuk ke CloudFormation konsol, pilih CloudFormation tumpukan yang ada, dan pilih Perbarui.
2. Pilih Ganti template saat ini.
3. Di bawah Tentukan template:
   a. Pilih URL Amazon S3.
   b. Salin tautan CloudFormation templat terbaru.
   c. Tempel tautan di kotak URL Amazon S3.
   d. Verifikasi bahwa URL templat yang benar ditampilkan di kotak teks URL Amazon S3, dan pilih Berikutnya. Pilih Selanjutnya sekali lagi.
4. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi sesuai kebutuhan. Untuk detail tentang parameter, lihat Langkah 1: Luncurkan tumpukan dasbor Deployment.
5. Pilih Berikutnya.

6. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
7. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Centang kotak yang mengakui bahwa template akan membuat sumber daya IAM.
8. Pilih Lihat set perubahan dan verifikasi perubahan.
9. Pilih Perbarui tumpukan untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status UPDATE_COMPLETE dalam waktu sekitar 10 menit.

Jika versi Solusi yang ada sebelum v2.0.0, pembaruan akan membuat tumpukan UI web (yang menggantikan `amplify-ui` implementasi layar login dengan UI yang dihosting Cognito) dan CloudFront URL baru, yang dapat diperoleh dari bagian Output CloudFormation konsol setelah status tumpukan UPDATE_COMPLETE.

> **Note**
>
> Kasus penggunaan yang ada yang dibuat menggunakan versi sebelum v2.0.0 TIDAK akan ditampilkan sampai Anda menyelesaikan langkah-langkah yang diuraikan di bawah ini.

Langkah 2: Migrasikan konfigurasi kasus penggunaan (Hanya pembaruan dari versi di bawah 2.0.0)

Skema penyimpanan dan layanan AWS untuk menyimpan konfigurasi kasus penggunaan telah berubah di versi 2.0.0. Ikuti langkah-langkah yang dijelaskan dalam Panduan Pengguna Migrasi GAAB v2 menggunakan skrip gaab_v2_migration.py. Setelah menjalankan skrip, Anda dapat mengakses dasbor Deployment untuk melihat kasus penggunaan yang diterapkan.

> **Note**
>
> Anda harus mengikuti langkah-langkah di bawah ini untuk menyelesaikan migrasi kasus penggunaan.

Langkah 3: Perbarui kasus penggunaan

Anda dapat mengedit kasus penggunaan yang diterapkan dengan fitur baru yang tersedia di GAAB versi terbaru. Lihat Gunakan solusi untuk informasi tentang cara menggunakan fitur dalam solusi ini.

Untuk memperbarui kasus penggunaan ke versi terbaru, Anda harus menyelesaikan langkah-langkah kasus `Edit`use di dasbor Deployment (meskipun Anda mungkin tidak membuat perubahan apa pun). Tindakan ini memicu pembaruan CloudFormation tumpukan dengan versi template terbaru.

> **Note**
>
> Kasus penggunaan yang dibuat dengan versi 1.x atau 2.x dari solusi mungkin tidak berfungsi dengan versi yang lebih baru. Oleh karena itu, kami merekomendasikan kloning kasus penggunaan yang ada yang dibuat dengan versi sebelum v3.0.0 melalui dasbor Deployment. Kemudian, secara bertahap bermigrasi dan ganti dengan kasus penggunaan baru yang dibuat menggunakan v3.0.0 atau yang lebih baru.

Pemecahan Masalah

Bagian ini menyediakan instruksi pemecahan masalah untuk menerapkan dan menggunakan solusi.

Jika petunjuk ini tidak mengatasi masalah Anda, Contact Support memberikan petunjuk untuk membuka kasus Support untuk solusi ini.

Masalah: Menerapkan konfigurasi berkemampuan VPC, dengan Buat VPC untuk saya, gagal

Tumpukan dasbor Deployment atau tumpukan kasus penggunaan gagal diterapkan karena tidak dapat CloudFormation menyediakan sumber daya jaringan VPC.

Resolusi

Periksa batas kuota untuk VPCs, dan Elastic IPs di akun Anda. Batas default masing-masing 5 untuk Elastic IPs dan VPCs per akun AWS, per Wilayah AWS.

> **Note**
>
> Saat solusi membuat VPC, satu penyebaran berkemampuan VPC (Deployment Dashboard atau Use Case) adalah penyebaran 2-AZ dengan 1 subnet publik dan 1 subnet pribadi di setiap AZ, setiap subnet publik menyebarkan 1 NAT Gateway. Dengan 2 NAT Gateways, penerapan menggunakan 2 alamat IP publik dari batas kuota.

Beberapa batasan yang harus diperhatikan (per akun, per Wilayah):

- Jumlah VPCs - 5
- Jumlah alamat IP publik - 5
- Jumlah Titik Akhir VPC Gateway - 20
- Jumlah Titik Akhir VPC Inteface - 20

Masalah: Tumpukan kasus penggunaan tidak dapat dihapus CloudFormation setelah tumpukan dasbor Deployment dihapus

Jika tumpukan dasbor Deployment dihapus CloudFormation sebelum semua tumpukan kasus penggunaan dihapus, kasus penggunaan dapat berakhir dalam status terkunci (tidak dapat digunakan). Ini karena peran IAM yang dibuat oleh tumpukan dasbor Deployment tidak lagi ada yang mencegah modifikasi pada tumpukan kasus penggunaan.

Resolusi

> **Warning**
>
> Pastikan Anda membersihkan peran yang dibuat secara manual segera setelah digunakan. Ini adalah izin tinggi yang dapat dimanfaatkan pengguna untuk peningkatan peran.

Buat ulang peran IAM yang dihapus untuk mengaktifkan penghapusan tumpukan: CloudFormation

1. Buka CloudFormation konsol dan tentukan peran yang terkait dengan tumpukan terkunci Anda.
   a. Peran ARN dapat ditemukan di bagian info tumpukan berlabel peran IAM.
   b. Nama peran adalah yang berikut setelah:role/ dalam peran IAM ARN (misalnya, arn:aws:iam: ::role/) <account-id><role-name>
2. Buat peran baru di IAM dengan nama yang sama dengan peran yang dihapus.
   a. Pilih layanan AWS sebagai entitas tepercaya dan pilih CloudFormationdari drop-down.
   b. Tambahkan izin yang diperlukan. Jika Anda tidak yakin tentang izin yang diperlukan, Anda dapat menggunakan kebijakan yang dikelola AdministratorAccessAWS.
   c. Masukkan nama peran persis seperti yang diperoleh pada Langkah 1.
3. Kembali ke CloudFormation konsol dan hapus tumpukan yang terkunci.
4. Setelah semua tumpukan terkunci berhasil dihapus, kembali ke IAM dan hapus peran apa pun yang dibuat di Langkah 2.

Masalah: UI kasus penggunaan tidak mencerminkan perubahan dalam pengaturan

Saat kasus penggunaan diperbarui, UI diterapkan ke CloudFront. Namun, karena penerapan CloudFront cache serta file konfigurasi yang menentukan bagaimana beberapa pengaturan ditampilkan kepada pengguna, perubahan ini mungkin tidak segera tercermin.

Resolusi

CloudFront Distribusi dapat dibatalkan untuk memaksa konfigurasi baru disebarkan ke pengguna frontend.

1. Buka CloudFormation konsol dan tentukan CloudFront distribusi yang terkait dengan tumpukan kasus penggunaan Anda.
   a. Tumpukan kasus penggunaan harus dimulai dengan nama yang sama dengan yang Anda gunakan saat menerapkan kasus penggunaan.
   b. Temukan tumpukan bersarang yang sesuai dengan UI. Nama tumpukan bersarang harus dimulai dengan WebAppS3 StackS3 UINested. UINested StackResource
   c. Di bawah tab Sumber Daya, cari sumber daya jenis AWS::CloudFront::Distribution, lalu pilih ID fisik. Ini akan membuka distribusi di CloudFront konsol.
2. Arahkan ke tab Invalidations, lalu pilih Create Invalidation, dan masukkan path /*. Ini akan membatalkan semua jalur.
3. Di browser Anda sendiri, hapus cookie dan file cache apa pun yang terkait dengan kasus penggunaan.

Hubungi AWS Support

Jika Anda memiliki AWS Business Support+, AWS Enterprise Support, atau Unified Operations, Anda dapat menggunakan AWS Support Center untuk mendapatkan bantuan ahli terkait solusi ini. Bagian berikut memberikan petunjuk.

Buat kasus

1. Masuk ke Support Center.
2. Pilih Buat kasus.

Bagaimana kami bisa membantu?

1. Pilih Teknis.
2. Untuk Layanan, pilih Solusi.
3. Untuk Kategori, pilih Solusi Lain.
4. Untuk Keparahan, pilih opsi yang paling cocok dengan kasus penggunaan Anda.
5. Saat Anda memasukkan Layanan, Kategori, dan Tingkat Keparahan, antarmuka akan mengisi tautan ke pertanyaan pemecahan masalah umum. Jika Anda tidak dapat menyelesaikan pertanyaan Anda dengan tautan ini, pilih Langkah selanjutnya: Informasi tambahan.

Informasi tambahan

1. Untuk Subjek, masukkan teks yang merangkum pertanyaan atau masalah Anda.
2. Untuk Deskripsi, jelaskan masalah secara rinci, termasuk nama solusi ini: Generative AI Application Builder di AWS.
3. Pilih Lampirkan file.
4. Lampirkan informasi yang dibutuhkan AWS Support untuk memproses permintaan.

Bantu kami menyelesaikan kasus Anda lebih cepat

1. Masukkan informasi yang diminta.
2. Pilih Langkah selanjutnya: Selesaikan sekarang atau hubungi kami.

Selesaikan sekarang atau hubungi kami

1. Tinjau solusi Selesaikan sekarang.
2. Jika Anda tidak dapat menyelesaikan masalah Anda dengan solusi ini, pilih Hubungi kami, masukkan informasi yang diminta, dan pilih Kirim.

Copot pemasangan solusinya

> **Note**
>
> Penerapan yang dibuat melalui dasbor Deployment tidak dimaksudkan untuk dikelola di luar solusi. Pastikan untuk menghapus dan membersihkan penerapan apa pun dari dalam dasbor Deployment, sebelum menghapus tumpukan. CloudFormation

Anda dapat menghapus instalasi Generative AI Application Builder pada solusi AWS dari AWS Management Console atau dengan menggunakan AWS Command Line Interface. Anda harus secara manual menghapus bucket Amazon S3, indeks Amazon Kendra, atau CloudWatch Log yang dibuat oleh solusi ini. AWS Solutions tidak secara otomatis menghapus bucket Amazon S3, indeks Amazon Kendra, atau CloudWatch Log jika Anda telah menyimpan data untuk disimpan.

Menggunakan Konsol Manajemen AWS

1. Masuk ke CloudFormation konsol AWS.
2. Pada halaman Stacks, pilih tumpukan instalasi solusi ini.
3. Pilih Hapus.

Menggunakan AWS Command Line Interface

Tentukan apakah AWS Command Line Interface (AWS CLI) tersedia di lingkungan Anda. Untuk petunjuk penginstalan, lihat Apa itu Antarmuka Baris Perintah AWS di Panduan Pengguna AWS CLI. Setelah mengonfirmasi bahwa AWS CLI tersedia, jalankan perintah berikut.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Langkah-langkah uninstall manual

Menghapus bucket Amazon S3

Solusi ini dikonfigurasi untuk mempertahankan bucket Amazon S3 yang dibuat solusi jika Anda memutuskan untuk menghapus CloudFormation tumpukan AWS untuk mencegah kehilangan data

yang tidak disengaja. Setelah menghapus instalan solusi, Anda dapat menghapus bucket Amazon S3 ini secara manual jika Anda tidak perlu menyimpan data. Ikuti langkah-langkah ini untuk menghapus bucket Amazon S3.

1. Masuk ke konsol Amazon S3.
2. Di panel navigasi, pilih Bucket.
3. Temukan ember <stack-name>S3.
4. Pilih bucket S3 dan pilih Delete.

Untuk menghapus bucket S3 menggunakan AWS CLI, jalankan perintah berikut. Anda tidak perlu mengosongkan bucket terlebih dahulu saat menggunakan opsi --force.

```
$ aws s3 rb s3://<bucket-name> --force
```

Menghapus indeks Amazon Kendra

Untuk mencegah kehilangan data yang tidak disengaja, solusi ini dikonfigurasi untuk mempertahankan indeks Amazon Kendra yang dibuat solusi saat tumpukan AWS telah dihapus. CloudFormation Setelah menghapus instalan solusi, Anda dapat secara manual menghapus indeks Amazon Kendra yang tidak perlu lagi Anda simpan datanya. Ikuti langkah-langkah ini untuk menghapus indeks Amazon Kendra.

1. Masuk ke konsol Amazon Kendra.
2. Di panel navigasi, pilih Indeks.
3. Cari dan pilih indeks yang ingin Anda hapus.
4. Pilih Hapus untuk menghapus indeks yang dipilih.

Untuk menghapus indeks Amazon Kendra menggunakan AWS CLI, jalankan perintah berikut:

```
$ aws kendra delete-index --id<index-id>
```

Menghapus Log CloudWatch

Untuk mencegah kehilangan data yang tidak disengaja, kami mengonfigurasi solusi ini untuk mempertahankan CloudWatch Log jika Anda memutuskan untuk menghapus CloudFormation

tumpukan. Setelah menghapus instalan solusi, Anda dapat menghapus log secara manual jika Anda tidak perlu menyimpan data. Ikuti langkah-langkah ini untuk menghapus CloudWatch Log.

1. Masuk ke CloudWatch konsol Amazon.
2. Di panel navigasi, pilih Grup Log.
3. Temukan grup log yang dibuat oleh solusi.
4. Pilih salah satu grup log.
5. Pilih Tindakan dan kemudian pilih Hapus.

Ulangi langkah-langkahnya hingga Anda menghapus semua grup log solusi.

Gunakan solusinya

Mengakses UI

Selama proses penyebaran tumpukan (untuk dasbor Deployment dan kasus penggunaan) email dikirim ke alamat email yang dikonfigurasi. Email berisi kredensyal sementara pengguna yang dapat mereka gunakan untuk mendaftar dan mengakses antarmuka web.

> **Note**
>
> DevOps Pengguna yang memiliki akses ke AWS Management Console harus menyediakan CloudFront URL UI dasbor Deployment kepada pengguna admin saat tumpukan selesai.

Untuk kasus penggunaan, pengguna admin dengan akses ke UI dasbor Deployment harus memberi pengguna bisnis CloudFront URL UI kasus penggunaan saat penerapan selesai.

Setelah masuk, pengguna dapat berinteraksi dengan solusinya UIs, baik dasbor Deployment dalam kasus admin, atau kasus penggunaan dalam kasus pengguna bisnis.

Cara memperbarui penerapan

Saat berada di halaman beranda dasbor Deployment (atau halaman detail penerapan), Anda dapat mengedit konfigurasi yang digunakan oleh penerapan. Anda hanya dapat mengedit penerapan yang ada dalam status CREATE_COMPLETE atau UPDATE_COMPLETE.

Kecuali untuk nama kasus penggunaan, semua opsi lain dapat diedit untuk penerapan. Cukup ubah nilai yang ingin Anda edit dan gunakan kembali.

Bergantung pada ruang lingkup pengeditan yang dilakukan, waktu pemindahan akan bervariasi. Mungkin perlu beberapa detik jika pengaturan sederhana telah berubah (contoh, parameter model), menjadi lebih dari 30 menit jika opsi terkait infrastruktur yang lebih besar telah berubah (misalnya, permintaan untuk membuat indeks Amazon Kendra untuk kasus penggunaan Teks RAG).

Setelah pengeditan berhasil diselesaikan, status aplikasi akan melaporkan status UPDATE_COMPLETE. Pada saat ini, Anda dapat mengakses UI yang diterapkan melalui CloudFront URL dan berinteraksi dengan penerapan yang dimodifikasi.

> **Note**
>
> Mungkin lebih mudah untuk menjalankan beberapa penerapan side-by-side jika Anda ingin membandingkan pengaturan yang berbeda atau. LLMs Gunakan fitur Clone untuk menggunakan konfigurasi yang ada dengan cepat untuk meluncurkan penerapan baru.

Cara mengkloning penerapan

Saat berada di halaman beranda dasbor Deployment (atau halaman detail penerapan), Anda dapat mengkloning konfigurasi yang digunakan oleh penerapan. Mengkloning penerapan meluncurkan panduan kasus penggunaan baru Deploy, tetapi dengan sebagian besar bidang yang telah diisi sebelumnya dengan nilai yang sama.

Ini adalah pengoperasian yang mudah untuk membantu Anda menduplikasi penerapan dengan cepat dengan pengaturan yang diubah, menghidupkan kembali penerapan yang dihapus, atau membandingkan beberapa LLMs dalam penerapan yang identik.

Cara menghapus penerapan

Saat berada di halaman beranda dasbor Deployment (atau halaman detail penerapan), Anda dapat menghapusnya setelah Anda tidak lagi memerlukan penerapan. Menghapus penerapan akan memanggil operasi penghapusan CloudFormation tumpukan dan membatalkan ketentuan sumber daya untuk penerapan.

Secara default, penerapan yang dihapus masih tetap ada di dasbor untuk mengaktifkan fungsionalitas klon. Untuk menghapus penerapan sepenuhnya dari dasbor sehingga berhenti dilacak di UI, pilih Hapus secara permanen di jendela konfirmasi hapus.

> **Important**
>
> Beberapa sumber daya tertinggal selama penghapusan tumpukan dan harus dihapus secara manual. Lihat bagian Penghapusan pemasangan manual untuk detail tentang sumber daya apa yang disimpan dan cara membersihkannya.

Mengkonfigurasi Model Bahasa Besar (LLM)

LLM mana yang tepat untuk kasus penggunaan Anda tergantung pada serangkaian besar faktor khusus untuk kebutuhan Anda dan jenis pengalaman pelanggan yang ingin Anda kurasi. Solusi ini tidak terlihat preskriptif, melainkan bertujuan untuk memberi Anda alat yang diperlukan untuk mengevaluasi apa yang paling cocok untuk aplikasi Anda.

Ruang yang dihasilkan AI berkembang pesat, jadi Anda harus tetap up to date pada model terbaru, teknik optimasi, dan praktik terbaik untuk memastikan Anda membangun pengalaman yang tepat bagi pelanggan Anda.

> **Note**
>
> Jika Anda bekerja dengan data non-publik atau sensitif, pastikan untuk memilih opsi LLM menggunakan layanan AWS (seperti Amazon Bedrock atau Amazon SageMaker AI). Ini meningkatkan postur keamanan keseluruhan penerapan Anda dengan menyimpan data di dalam Wilayah Anda dan di jaringan AWS jika dibandingkan dengan menggunakan LLM yang dihosting oleh penyedia pihak ketiga.

Menggunakan Amazon SageMaker AI sebagai Penyedia LLM

Mulai v1.3.0, Amazon SageMaker AI tersedia sebagai penyedia model untuk kasus penggunaan Teks. Fitur ini memungkinkan Anda untuk menggunakan titik akhir inferensi SageMaker AI yang sudah ada dalam akun AWS dalam solusinya. Berikut adalah beberapa cara untuk memulai.

> **Important**
>
> Solusinya tidak mengelola siklus hidup titik akhir SageMaker AI Anda. Anda bertanggung jawab untuk menghapus titik akhir SageMaker AI setelah mereka tidak lagi diperlukan untuk berhenti menimbulkan biaya tambahan.

Membuat titik akhir SageMaker AI

Anda dapat menggunakan Amazon SageMaker AI JumpStart untuk menyebarkan titik akhir dengan cepat.

Anda juga dapat menggunakan titik akhir SageMaker AI berbasis generasi teks dan menerapkan menggunakan layanan AI dasar. SageMaker Lihat JumpStart dokumentasi SageMaker AI untuk panduan langkah demi langkah tentang cara menerapkan model untuk inferensi.

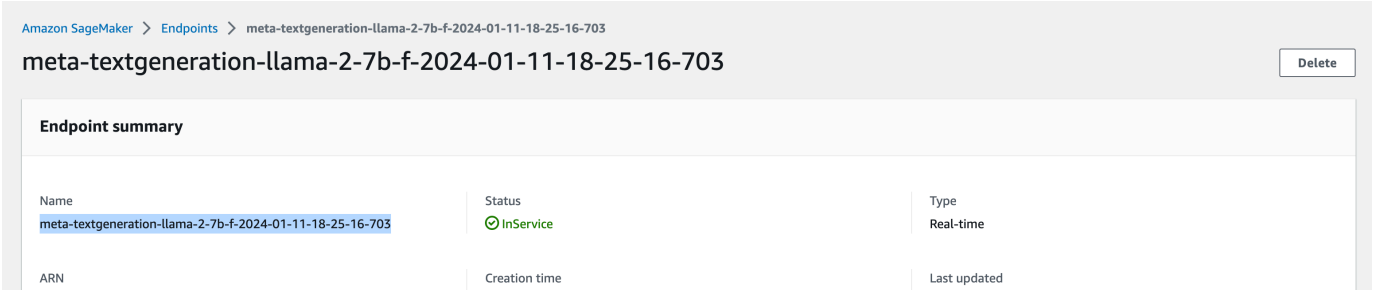
> **Note**
>
> Foundation models/LLMs biasanya cukup besar dan seringkali memerlukan penggunaan instance komputasi dipercepat yang besar. Banyak dari instans yang lebih besar ini mungkin tidak tersedia secara default di akun AWS Anda. Lihat kuota SageMaker AI default dan pastikan untuk meminta peningkatan kuota sebelum menerapkan untuk menghindari kegagalan penerapan umum.

Gunakan titik akhir SageMaker AI untuk membuat penerapan kasus penggunaan Teks

Untuk menerapkan kasus penggunaan Teks baru menggunakan titik akhir SageMaker AI untuk inferensi:

1. Buat kasus penggunaan baru melalui panduan dasbor Deployment dan lengkapi formulir hingga Anda mencapai halaman pemilihan Model.
2. Pada halaman Model, pilih SageMaker AI sebagai penyedia model. Ini akan menghasilkan formulir khusus yang membutuhkan tiga bagian kunci input pengguna:
   - Nama titik akhir SageMaker AI yang ingin Anda gunakan. DevOps pengguna dapat memperoleh ini dari konsol AWS. Perhatikan bahwa titik akhir harus berada di akun dan Wilayah yang sama dengan solusi yang digunakan.

     Lokasi nama titik akhir di konsol AWS

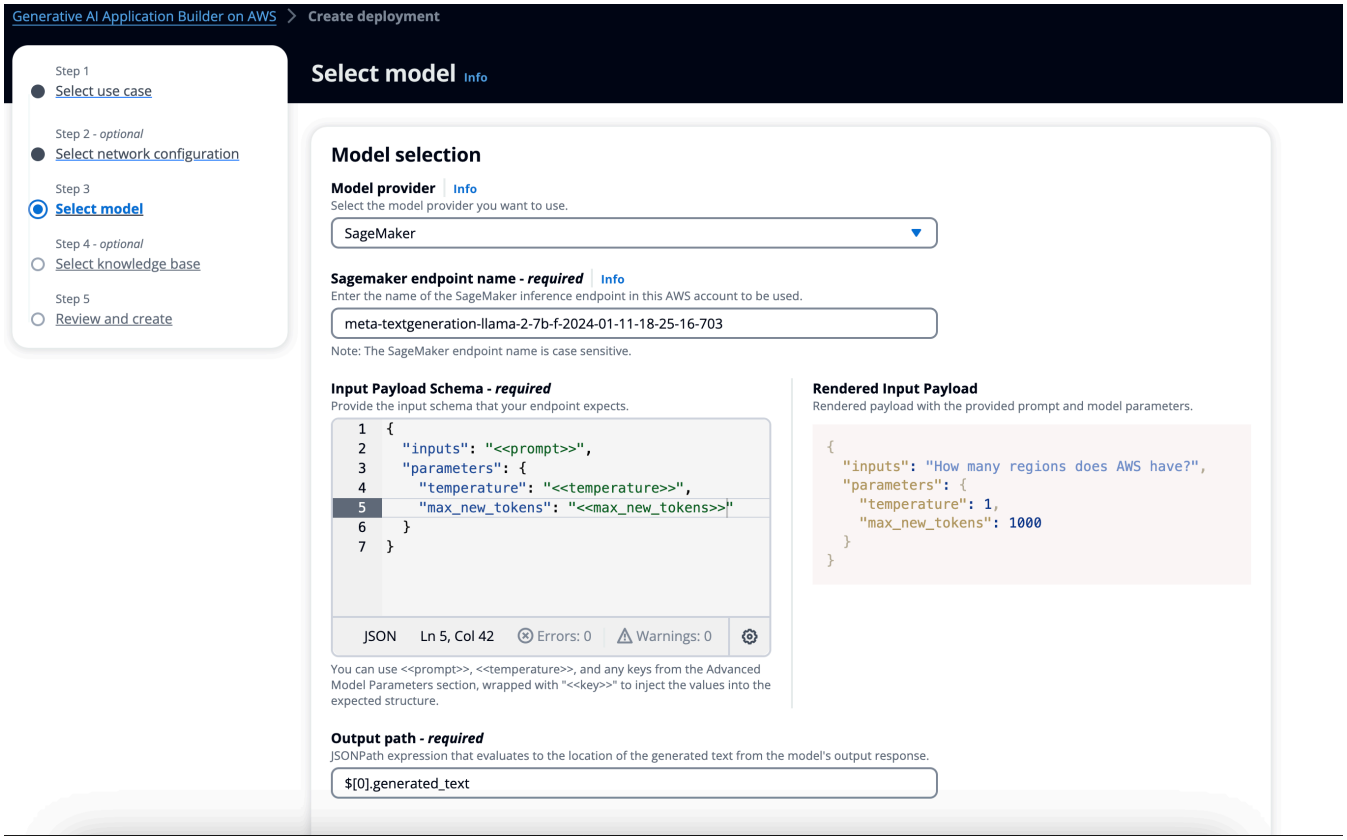
     

   - Skema payload input yang diharapkan oleh titik akhir. Untuk mendukung kumpulan titik akhir terluas, pengguna admin diharuskan memberi tahu solusi bagaimana titik akhir mereka mengharapkan input diformat. Dalam panduan pemilihan model, berikan skema JSON

untuk solusi yang akan dikirim ke titik akhir. Anda dapat menambahkan placeholder untuk menyuntikkan nilai statis dan dinamis ke payload permintaan. Opsi yang tersedia adalah:

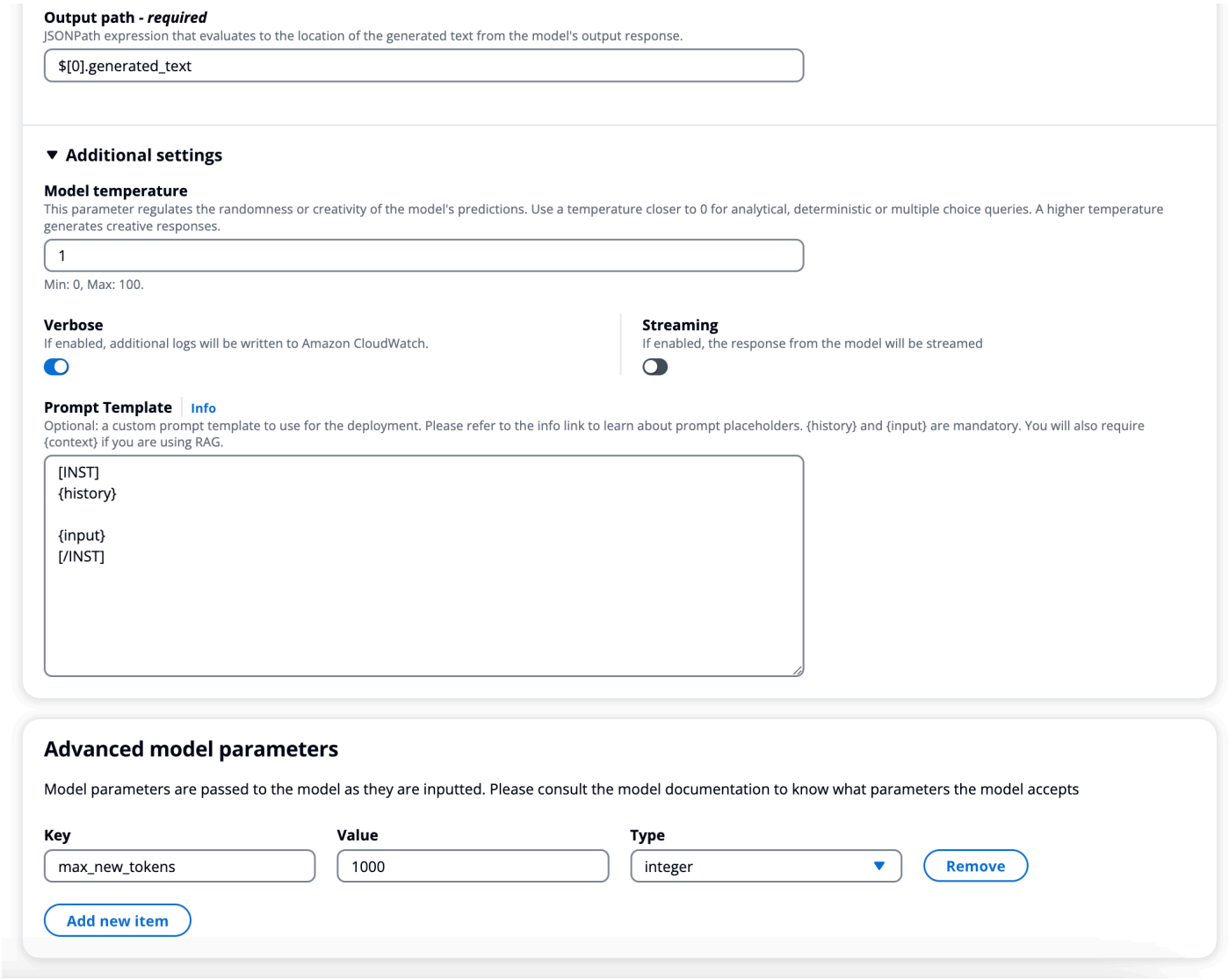
- Placeholder wajib:\ <\ <prompt\ >\ > akan diganti secara dinamis dengan input penuh (misalnya, riwayat, konteks, dan input pengguna sesuai template prompt) untuk dikirim ke titik akhir SageMaker AI saat runtime.
- Placeholder opsional:\ <\ <temperature\ >\ > *,\ * serta parameter apa pun yang ditentukan dalam parameter model lanjutan dapat diberikan ke titik akhir. Setiap string yang berisi placeholder tertutup dalam\ <\ < and\ >\ > (misalnya,\ <\ <max_new_tokens\ >\ >) akan diganti dengan nilai parameter model lanjutan dengan nama yang sama.

  Contoh skema masukan - pengaturan bidang wajib, prompt dan suhu, bersama dengan parameter lanjutan khusus, max_new_tokens. Jalur keluaran harus disediakan sebagai JSONPath string yang valid



3. Lokasi respon string LLMs yang dihasilkan dalam payload output. Ini harus diberikan sebagai JSONPath ekspresi untuk menunjukkan di mana respons teks akhir yang ditampilkan kepada pengguna diharapkan dapat diakses dari dalam objek dan respons titik akhir yang dikembalikan.

   Contoh penambahan parameter model Lanjutan untuk digunakan dalam skema input SageMaker AI (lihat Gambar 2 untuk opsi/pengaturan sebelumnya)



> **Note**
>
> SageMaker AI sekarang mendukung hosting beberapa model di belakang titik akhir yang sama, dan ini adalah konfigurasi default saat menerapkan titik akhir di versi SageMaker AI Studio saat ini (bukan Studio Classic).
> Jika titik akhir Anda dikonfigurasi dengan cara ini, Anda akan diminta untuk menambahkan InferenceComponentNameke bagian parameter model lanjutan, dengan nilai yang sesuai dengan nama model yang ingin Anda gunakan.

Pengaturan LLM Tingkat Lanjut

Saat menggunakan Amazon Bedrock, Anda dapat mengonfigurasi beberapa pengaturan lanjutan untuk model Anda seperti Amazon Bedrock Guardrails, Provisioned Throughput untuk Amazon Bedrock, dan parameter model tambahan.

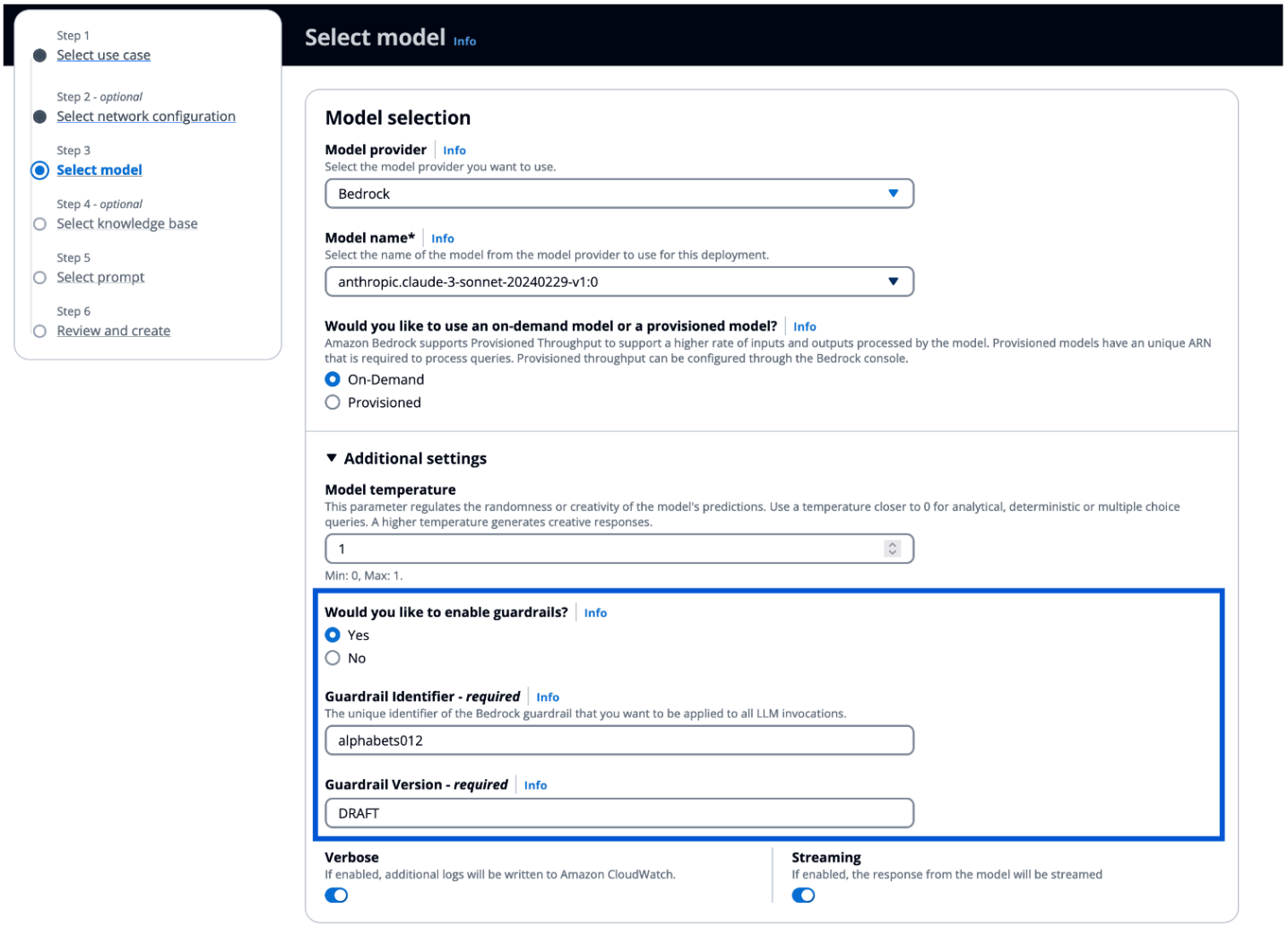
Pagar Batuan Dasar Amazon

Amazon Bedrock Guardrails adalah fitur dengan Amazon Bedrock yang mengevaluasi input pengguna dan respons LLM berdasarkan kebijakan yang dikonfigurasi pengguna dan menyediakan lapisan perlindungan tambahan, terlepas dari LLM yang mendasari yang dipilih pengguna untuk kasus penggunaan. Pagar Pembatas terdiri dari 2 kebijakan untuk menghindari konten yang termasuk dalam kategori yang tidak diinginkan atau berbahaya:

1. Topik yang ditolak untuk menentukan serangkaian topik yang tidak diinginkan dalam konteks aplikasi pengguna, misalnya, saran investasi dalam aplikasi keuangan, dan,
2. Filter konten****yang memungkinkan pemfilteran input permintaan pengguna atau respons model yang berisi konten berbahaya.

Untuk penggunaan dalam solusi Generative AI Application Builder, Guardrail harus dikonfigurasi di konsol Amazon Bedrock menggunakan wizard Create guardrail. Setelah dibuat, Anda dapat menambahkan Guardrail ini ke kasus penggunaan obrolan yang dibuat melalui panduan solusi Generative AI Application Builder di pengaturan Tambahan di langkah Pemilihan Model dengan menyediakan versi Guardrail Identifier dan Guardrail Anda.

Menggambarkan wizard Penerapan - mengaktifkan Amazon Bedrock Guardrails



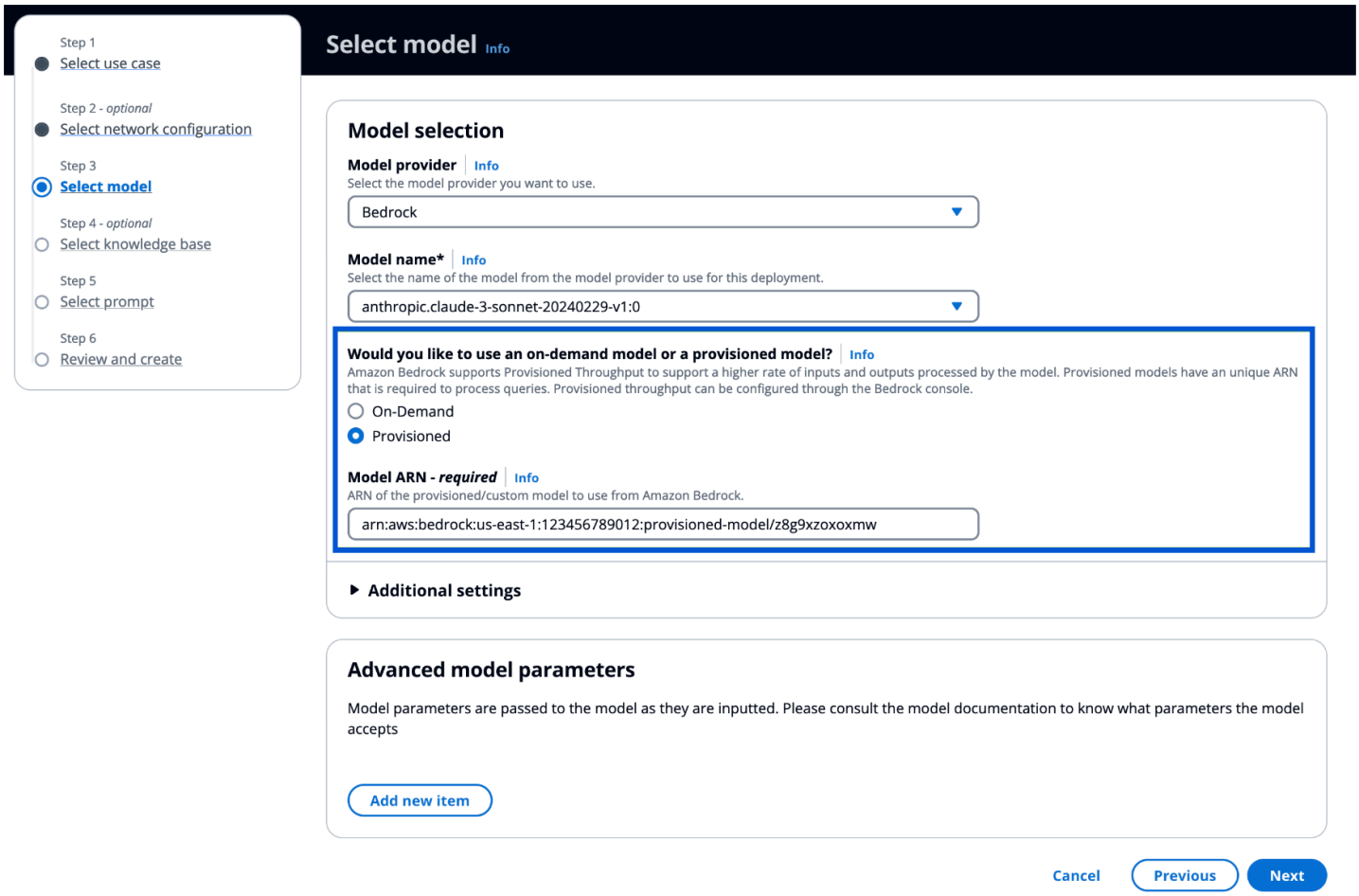
Throughput yang Disediakan untuk Amazon Bedrock

Setiap model Amazon Bedrock sesuai permintaan mengikuti batas kuota akun khusus wilayah untuk inferensi model. Misalnya, Anthropic Claude 2.x di Bedrock saat ini memungkinkan 500 permintaan dan 500.000 token diproses per menit di wilayah us-east-1 dan us-west-2. Anda mungkin juga ingin menggunakan solusi dengan model pra-terlatih yang telah disetel atau dilanjutkan. Untuk contoh seperti itu, Amazon Bedrock memungkinkan throughput yang disediakan yang memungkinkan menjalankan beban kerja inferensi konsisten yang besar untuk model dasar, yang disetel dengan baik, atau dilanjutkan yang dilatih sebelumnya untuk digunakan dalam aplikasi tingkat produksi.

Setelah Throughput yang Disediakan dibeli dalam konsol Amazon Bedrock, Model ARN dibuat untuk digunakan. Anda sekarang dapat menyediakan ARN Model ini di wizard Generative AI Application Builder di langkah pemilihan Model. Untuk melakukannya, pilih Bedrock sebagai penyedia model dan nama model dasar yang digunakan untuk menghasilkan Model ARN yang disediakan ini di

konsol Amazon Bedrock. Kemudian, pilih 'Model yang disediakan' saat memilih antara model sesuai permintaan dan model yang disediakan, dan berikan ARN Model Anda.

Menggambarkan panduan Penerapan - Mengaktifkan Throughput yang Disediakan untuk Amazon Bedrock



> **Note**
>
> Pagar pembatas dan throughput yang disediakan harus berada di Wilayah yang sama dengan Dasbor Deployment yang diterapkan dan tumpukan kasus penggunaan.

Parameter model

LLMs sering menerima berbagai parameter khusus untuk implementasinya. Penyedia model sering memberikan dokumentasi yang menguraikan kumpulan parameter yang didukung dan penggunaannya.

Solusi meneruskan parameter model langsung ke model yang mendasarinya sehingga penting untuk memastikan parameter diatur dengan benar. Lihat dokumentasi penyedia model untuk informasi terbaru tentang parameter yang didukung.

Mengkonfigurasi Agen Builder

Agent Builder menyediakan opsi konfigurasi komprehensif untuk membuat agen AI siap produksi. Bagian ini menjelaskan cara mengonfigurasi dan mengelola penerapan Agent Builder.

Konfigurasi prompt sistem

Prompt sistem mendefinisikan perilaku, kepribadian, dan kemampuan agen Anda. Untuk mengkonfigurasi prompt sistem:

1. Di wizard Agent Builder, navigasikan ke langkah Configure Agent.
2. Edit template prompt sistem di editor teks.
3. Sertakan instruksi yang jelas untuk:
   - Peran dan tujuan agen
   - Cara menggunakan alat yang tersedia (server MCP)
   - Preferensi pemformatan respons
   - Pedoman perilaku
4. Gunakan tombol Reset to default untuk mengembalikan template asli jika diperlukan.

Praktik terbaik untuk permintaan agen:

- Jadilah spesifik tentang kemampuan dan keterbatasan agen
- Berikan contoh yang jelas tentang perilaku yang diinginkan
- Sertakan instruksi untuk penggunaan alat dan kapan harus memanggilnya
- Tentukan ekspektasi format respons
- Tetapkan batasan untuk perilaku agen

Integrasi server MCP

Server Model Context Protocol (MCP) menyediakan agen akses ke alat perusahaan dan sumber data. Untuk mengkonfigurasi server MCP:

1. Pada langkah Configure Agent, cari bagian MCP Servers.
2. Pilih dari server MCP yang tersedia di menu tarik-turun.

> **Note**
>
> Server MCP harus dikonfigurasi dan dapat diakses sebelum penyebaran agen. Agen akan secara otomatis menemukan dan menggunakan alat yang diekspos oleh server MCP yang dikonfigurasi. Lihat dokumentasi MCP untuk pengaturan server dan konfigurasi alat.

Pengaturan memori

Agen Builder menyediakan dua jenis memori untuk mempertahankan konteks dan pengetahuan:

Memori jangka pendek

Diaktifkan secara default untuk semua agen:

- Mempertahankan konteks percakapan dalam sesi
- Secara otomatis menangkap pesan pengguna dan tanggapan agen
- Diorganisir oleh ActorID dan SessionID untuk isolasi yang tepat
- Tidak diperlukan konfigurasi

Memori jangka panjang

Fitur opsional untuk menyimpan wawasan di seluruh sesi:

1. Pada langkah Configure Agent, cari bagian Memory Configuration.
2. Alihkan Aktifkan memori jangka panjang untuk mengaktifkan.
3. Saat diaktifkan, agen dapat:
   - Ekstrak dan simpan informasi penting di seluruh percakapan
   - Mengambil konteks yang relevan dari sesi sebelumnya
   - Membangun pengetahuan tentang preferensi dan riwayat pengguna

> **Note**
>
> Memori jangka panjang menggunakan AgentCore Memori dengan strategi memori semantik dan pengaturan retensi default.

Pemantauan penyebaran Agen Builder

Agent Builder menyediakan pemantauan komprehensif melalui CloudWatch dasbor dan metrik.

Mengakses dasbor CloudWatch

1. Arahkan ke CloudWatch konsol di akun AWS Anda.
2. Pilih Dasbor dari navigasi kiri.
3. Temukan dasbor bernama`AgentBuilder-<UseCaseId>`.
4. Lihat metrik waktu nyata dan data kinerja historis.

Akses log dan analisis

Log agen tersedia di CloudWatch Log:

1. Arahkan ke CloudWatch Log di konsol AWS.
2. Temukan grup log yang diawali dengan`/aws/bedrock-agentcore/runtimes/`.
3. Gunakan CloudWatch Wawasan untuk menanyakan dan menganalisis log.
4. Cari permintaan IDs atau pola kesalahan tertentu.

Mengkonfigurasi Pembuat Alur Kerja

Workflow Builder memungkinkan orkestrasi multi-agen melalui agen supervisor yang mendelegasikan pekerjaan ke agen Agen Builder khusus.

Membuat alur kerja

1. Arahkan ke Dasbor Deployment
2. Pilih Buat Kasus Penggunaan Alur Kerja

3. Konfigurasikan agen supervisor:
   - Nama: Nama deskriptif untuk alur kerja
   - Deskripsi: Tujuan dan kemampuan
   - System Prompt: Instruksi untuk delegasi dan koordinasi agen
   - Model: Model dasar untuk agen pengawas

Praktik terbaik untuk petunjuk supervisor:

- Jelaskan dengan jelas kapan harus menggunakan masing-masing agen khusus
- Sertakan instruksi untuk mengumpulkan hasil dari beberapa agen
- Tentukan harapan pemformatan respons
- Tetapkan batasan untuk perilaku delegasi

Pemilihan agen

Pilih agen Builder Agen untuk dimasukkan sebagai agen khusus:

1. Klik Tambahkan Agen dalam konfigurasi alur kerja
2. Jelajahi atau cari agen agen pembangun yang tersedia
3. Tinjau deskripsi agen
4. Pilih agen untuk disertakan dalam alur kerja

Deskripsi agen

Agen supervisor menggunakan deskripsi agen untuk memutuskan agen mana yang akan didelegasikan. Pastikan deskripsi menjelaskan dengan jelas:

- Domain atau kemampuan khusus agen
- Jenis tugas yang ditangani agen
- Harapan input/output

Menguji alur kerja

Setelah penyebaran:

1. Akses alur kerja melalui Dasbor Deployment
2. Uji dengan kueri yang membutuhkan banyak agen
3. Pantau delegasi agen di log CloudWatch
4. Tinjau kualitas respons dan pola delegasi
5. Sesuaikan prompt supervisor jika delegasi kurang optimal

Kiat untuk mengelola batas token model

Catatan: Solusinya tidak secara langsung mencoba mengelola batas token yang diberlakukan oleh berbagai macam LLMs. Uji dan pastikan prompt Anda tetap dalam batas yang tersedia yang diberlakukan oleh penyedia model.

Untuk membantu mengontrol ukuran petunjuk, coba yang berikut ini:

1. Biasakan diri Anda dengan batasan yang diberlakukan oleh model yang ingin Anda gunakan. Nilai-nilai ini dapat berbeda secara dramatis di seluruh model sehingga penting untuk mengetahui berapa anggaran Anda yang tersedia sebelum memulai.
2. Buat prompt awal Anda dengan mempertimbangkan anggaran itu dan pertimbangkan berapa banyak yang ingin Anda simpan untuk elemen dinamis apa pun dari prompt. Misalnya, input pengguna, riwayat obrolan, kutipan dokumen, dan sebagainya.
3. Di halaman konfigurasi prompt, tetapkan batas untuk Ukuran riwayat tambahan untuk membatasi jumlah giliran percakapan yang disertakan dalam prompt.
4. Tetapkan batas pengembalian dokumen di wizard konfigurasi Basis Pengetahuan. Anda perlu mencoba dan mencapai keseimbangan yang tepat antara menyediakan LLM dengan konteks yang cukup untuk melakukan tugas, tetapi tidak melebihi batas token atau memengaruhi latensi secara negatif.
5. Tinggalkan beberapa buffer. Jangan menganggarkan untuk kasus tipikal, pikirkan dan bereksperimen dengan kasus tepi seperti kueri input panjang, kutipan dokumen besar, atau percakapan panjang.

Langkah-langkah untuk membangun MCP server Docker Image

Untuk menggunakan server MCP (Model Context Protocol) dengan Generative AI Application Builder di AWS, Anda memerlukan image Docker yang dibuat dan disimpan dalam repositori Amazon ECR pribadi sebagai langkah pertama.

> **Note**
>
> Sampai sekarang, server MCP yang digunakan di AgentCore runtime Amazon Bedrock tidak dapat diekspor ke GAAB. Agar server MCP dapat dilampirkan ke Agen yang dibuat melalui GAAB, mereka harus dibuat melalui GAAB.

Langkah 1: Buat server MCP Anda

Pertama, Anda harus menyiapkan implementasi server MCP Anda. Untuk petunjuk terperinci tentang membuat server MCP, lihat Panduan AgentCore Pengembang Amazon Bedrock - Buat server MCP.

Kami merekomendasikan struktur proyek berikut:

```
.
### __init__.py
### extras/
#   ### extra_dependencies.py
#   ### Dockerfile
### requirements.txt
### server.py <-- Server Entry point
```

Untuk struktur Dockerfile, sebaiknya gunakan format yang mirip dengan contoh berikut:

```
FROM ghcr.io/astral-sh/uv:python3.13-bookworm-slim
WORKDIR /app

# All environment variables in one layer
ENV UV_SYSTEM_PYTHON=1 \
    UV_COMPILE_BYTECODE=1 \
    UV_NO_PROGRESS=1 \
    PYTHONUNBUFFERED=1 \
    DOCKER_CONTAINER=1 \
    AWS_REGION=us-east-1 \
    AWS_DEFAULT_REGION=us-east-1

COPY requirements.txt requirements.txt
# Install from requirements file
RUN uv pip install -r requirements.txt

RUN uv pip install aws-opentelemetry-distro>=0.10.1
```

```
# Signal that this is running in Docker for host binding logic
ENV DOCKER_CONTAINER=1

# Create non-root user
RUN useradd -m -u 1000 bedrock_agentcore
USER bedrock_agentcore

EXPOSE 9000
EXPOSE 8000
EXPOSE 8080

# Copy entire project (respecting .dockerignore)
COPY . .

# Use the full module path
CMD ["opentelemetry-instrument", "python", "-m", "server"]
```

Langkah 2: Uji server MCP Anda secara lokal

Sebelum menerapkan ke AWS, penting untuk menguji server MCP Anda secara lokal untuk memastikannya berfungsi seperti yang diharapkan. Untuk petunjuk terperinci tentang pengujian lokal, lihat Panduan AgentCore Pengembang Amazon Bedrock - Uji server MCP Anda secara lokal.

Langkah 3: Terapkan ke Amazon ECR

Setelah server MCP Anda dibuat dan diuji secara lokal, ikuti langkah-langkah berikut untuk menerapkannya ke Amazon ECR:

1. Pastikan Anda telah menginstal AWS CLI dan Docker versi terbaru. Untuk informasi selengkapnya, lihat Memulai Amazon ECR.
2. Ambil token otentikasi dan autentikasi klien Docker Anda ke registri Anda. Gunakan AWS CLI:

   ```
   aws ecr get-login-password --region us-east-1 | docker login --username AWS --
   password-stdin <account-id>.dkr.ecr.us-east-1.amazonaws.com
   ```

3. Bangun image Docker Anda menggunakan perintah berikut. Untuk informasi tentang membuat file Docker dari awal, lihat dokumentasi Docker. Anda dapat melewati langkah ini jika gambar Anda sudah dibuat:

   ```
   docker build -t <repository-name> .
   ```

4. Setelah build selesai, beri tag gambar Anda sehingga Anda dapat mendorong gambar ke repositori ini:

```
docker tag <repository-name>:latest <account-id>.dkr.ecr.us-east-1.amazonaws.com/
<repository-name>:latest
```

5. Jalankan perintah berikut untuk mendorong gambar ini ke repositori AWS yang baru dibuat:

```
docker push <account-id>.dkr.ecr.us-east-1.amazonaws.com/<repository-name>:latest
```

Untuk petunjuk penerapan lengkap, lihat Panduan AgentCore Pengembang Amazon Bedrock - Menerapkan server MCP Anda ke AWS.

Langkah 4: Gunakan URI ECR di GAAB

Setelah berhasil mendorong image Docker Anda ke Amazon ECR, salin URI gambar dari konsol ECR. Anda akan menggunakan URI ini saat menerapkan server MCP Anda melalui Generative AI Application Builder on AWS deployment wizard.

Langkah-langkah untuk membuat Target MCP Gateway yang berbeda

Amazon Bedrock AgentCore Gateway memungkinkan Anda mengubah layanan AWS yang ada dan APIs menjadi alat MCP yang dapat digunakan oleh agen Anda. Gateway mendukung beberapa jenis target, memungkinkan Anda mengintegrasikan berbagai layanan backend dengan mulus.

Jenis target berikut didukung:

- Target Lambda: Ubah fungsi AWS Lambda menjadi alat MCP. Untuk petunjuk terperinci, lihat Panduan AgentCore Pengembang Amazon Bedrock - Tambahkan target Lambda.
- Target OpenAPI: Gunakan spesifikasi OpenAPI untuk mendefinisikan dan mengekspos REST sebagai alat MCP. APIs Untuk petunjuk terperinci, lihat Panduan AgentCore Pengembang Amazon Bedrock - skema OpenAPI.
- Target Smithy: Bangun alat MCP menggunakan definisi model Smithy untuk integrasi API yang aman untuk tipe aman. Untuk petunjuk terperinci, lihat Panduan AgentCore Pengembang Amazon Bedrock - Membangun target Smithy.

- Target MCP Server: Connect langsung ke server MCP eksternal melalui endpoint URL, memungkinkan Anda untuk mengintegrasikan server MCP yang ada. Untuk petunjuk terperinci, lihat Panduan AgentCore Pengembang Amazon Bedrock - target server MCP.

Untuk contoh dan tutorial tambahan tentang membuat target MCP Gateway, kunjungi repositori AgentCore sampel Amazon Bedrock.

Mengkonfigurasi basis pengetahuan

Bagian ini menjelaskan cara memasukkan data ke dalam basis pengetahuan yang telah Anda pilih untuk solusinya. Solusinya saat ini mendukung Amazon Kendra dan Amazon Bedrock Knowledge Bases sebagai basis pengetahuan untuk penerapan kasus penggunaan berbasis RAG Anda.

Amazon Kendra

Jika Anda menggunakan Amazon Kendra sebagai basis pengetahuan Anda, lihat Panduan Pengembang Amazon Kendra untuk informasi tentang cara menggunakan berbagai konektor sumber data untuk membantu Anda menyerap data dari berbagai sumber pilihan.

Penting: Untuk mencegah kehilangan data yang tidak disengaja, solusi tidak secara otomatis menghapus indeks Kendra (baik yang dibuat oleh solusi atau sebaliknya) ketika penerapan atau tumpukan dihapus. Jika Anda ingin menghapus basis pengetahuan Anda dan berhenti mengeluarkan biaya, lihat bagian Penghapusan pemasangan manual untuk detail tentang sumber daya mana yang disimpan dan cara membersihkannya.

Basis Pengetahuan Amazon Bedrock

Amazon Bedrock Knowledge Bases dapat didukung oleh berbagai penyimpanan vektor yang berbeda, masing-masing dengan kemampuan mengindeks data Anda. Untuk mengatur dan mengisi basis pengetahuan Anda, lihat Panduan Pengguna Amazon Bedrock. Secara khusus, Anda akan ingin:

- Pertama, atur sumber data Anda
- Kemudian siapkan indeks vektor untuk basis pengetahuan Anda di penyimpanan vektor yang didukung. Perhatikan bahwa ini dapat dilewati jika Anda menggunakan opsi “Cepat buat penyimpanan vektor baru” di konsol Bedrock selama pembuatan basis pengetahuan.
- Terakhir, Anda dapat membuat basis pengetahuan dan menyinkronkan sumber data yang dikonfigurasi.

Pengaturan basis pengetahuan tingkat lanjut

Pengaturan Basis Pengetahuan Tingkat Lanjut seperti Penyaringan Basis Pengetahuan dan RAG dengan Kontrol Akses Berbasis Peran tersedia untuk digunakan dengan solusi. Penyaringan Basis Pengetahuan dapat diterapkan ke salah satu Basis Pengetahuan sementara RAG dengan Kontrol Akses Berbasis Peran secara khusus tersedia untuk Amazon Kendra.

Penyaringan basis pengetahuan

Solusinya memungkinkan Anda menentukan filter atribut Amazon Kendra atau filter pengambilan basis pengetahuan Bedrock saat menerapkan kasus penggunaan di bagian konfigurasi RAG Tingkat Lanjut pada langkah basis pengetahuan wizard. Filter ini menentukan bagaimana sumber data dalam basis pengetahuan ditanyakan, seperti strategi pencarian, bahasa dokumen yang mendasarinya menjadi kueri, dll.

Dalam kedua kasus, objek JSON digunakan untuk menentukan pengaturan filter per format yang ditentukan dalam setiap dokumentasi layanan (seperti yang ditautkan di atas).

Contoh 1: Kendra AttributeFilter

```
{
 "EqualsTo": {
 "Key": "_language_code",
 "Value": {
 "StringValue": "es"
 }
 }
}
```

Contoh 2: Bedrock RetrievalFilter

```
{
 "equals": {
 "key": "language",
 "value": "es"
 }
}
```

RAG dengan Kontrol Akses Berbasis Peran dengan Amazon Kendra

Kontrol akses berbasis peran (RBAC) memungkinkan pengendalian pengguna atau grup mana yang dapat mengakses dokumen tertentu dalam indeks Amazon Kendra Anda atau melihat dokumen tertentu dalam hasil pencarian mereka. Untuk mengonfigurasi RBAC untuk ID Indeks Amazon Kendra Anda dengan kasus penggunaan Generative AI Application Builder on AWS (GAAB), ikuti langkah-langkah berikut:

1. Konfigurasikan Indeks Amazon Kendra

1. Pastikan Anda memiliki indeks Amazon Kendra yang dibuat dan setidaknya satu sumber data ditambahkan ke dalamnya.
2. Konfigurasikan kontrol akses untuk sumber data Anda berdasarkan grup pengguna. Untuk sumber data S3, ikuti petunjuk yang tersedia dalam dokumentasi untuk menyiapkan daftar kontrol akses (ACLs) menggunakan nama grup yang sama yang dibuat di Kumpulan Pengguna Amazon Cognito Anda. Ini memastikan bahwa pengguna hanya dapat mengakses dokumen dan hasil pencarian yang diizinkan untuk dilihat berdasarkan keanggotaan grup mereka.

> **Note**
>
> Di bawah Kontrol Akses Pengguna di Indeks Kendra yang Anda buat, biarkan kontrol akses pengguna berbasis Token sebagai No. Saat Anda mengaktifkan Kontrol Akses Berbasis Peran di Langkah 2, Generative AI Application Builder on AWS mengekstrak klaim yang sesuai dari token autentikasi pengguna dan membuat Filter Atribut.

2. Terapkan Kasus Penggunaan RAG menggunakan GAAB Deployment Wizard

1. Ikuti petunjuk wizard di layar di GAAB Deployment Wizard hingga Anda mencapai langkah 4 dari wizard untuk mengonfigurasi RAG.
2. Pada langkah Select Knowledge Base dari panduan penerapan, pilih Amazon Kendra sebagai tipe basis pengetahuan.
3. Tentukan apakah Anda memiliki indeks Amazon Kendra yang ada atau jika Anda ingin membuat yang baru. Jika Anda memiliki indeks yang ada, berikan ID indeks Amazon Kendra Anda yang telah dikonfigurasi dengan daftar kontrol akses (ACLs) berdasarkan grup pengguna.
4. Aktifkan opsi Kontrol Akses Berbasis Peran. Opsi ini memastikan bahwa hasil pencarian yang dikembalikan dari indeks Amazon Kendra difilter berdasarkan peran pengguna dan izin grup.

5. Tinjau dan terapkan kasus penggunaan.

3. Konfigurasikan Amazon Cognito

1. Temukan Kumpulan Pengguna Amazon Cognito yang digunakan oleh penerapan GAAB Anda. Kumpulan Pengguna Amazon Cognito ini biasanya dibuat oleh tumpukan dasbor penerapan utama. CloudFormation
2. Buat pengguna baru di Kumpulan Pengguna Amazon Cognito. Saat membuat pengguna, pilih opsi 'Kirim undangan email' sehingga pengguna menerima kredensil login sementara melalui email. Ini memungkinkan pengguna baru untuk mendaftar dan mengakses aplikasi GAAB.
3. Buat grup pengguna di Kumpulan Pengguna Amazon Cognito. Pastikan nama grup sama persis dengan grup yang dikonfigurasi dalam indeks Amazon Kendra Anda. ACLs Ini sangat penting untuk mengaktifkan RBAC, karena keanggotaan grup pengguna akan menentukan hasil pencarian yang dapat mereka akses.
4. Tetapkan pengguna ke grup yang sesuai berdasarkan peran dan izin akses mereka. Pengguna harus ditambahkan ke grup yang diperlukan untuk indeks ACL Amazon Kendra, serta grup khusus kasus penggunaan yang dibuat selama penerapan GAAB. Ini memastikan bahwa pengguna memiliki izin yang diperlukan untuk mengakses kasus penggunaan tertentu dan hasil pencarian yang relevan.

Dengan mengikuti langkah-langkah ini, Anda akan mengonfigurasi kontrol akses berbasis peran (RBAC) untuk penerapan GAAB Anda, memastikan bahwa pengguna hanya dapat mengakses dan berinteraksi dengan informasi dan fitur yang mereka otorisasi, berdasarkan grup pengguna dan izin yang ditetapkan.

> **Note**
>
> Sampai sekarang, hanya Amazon Kendra yang mendukung RBAC untuk basis pengetahuan di Generative AI Application Builder di AWS. Untuk Pangkalan Pengetahuan Amazon Bedrock, RBAC tidak didukung, tetapi Anda dapat menggunakan filter metadata untuk mencapai beberapa tingkat pemfilteran. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon Bedrock.

Mengonfigurasi prompt Anda

Panduan dasbor Deployment memiliki langkah konfigurasi cepat yang memungkinkan Anda menyesuaikan pengalaman dan templat prompt yang akan memandu interaksi antara pengguna dan model AI. Mengkonfigurasi pengaturan ini dengan benar sangat penting untuk mendapatkan respons yang akurat dan relevan dari asisten AI.

Bagian ini mengontrol keseluruhan pengalaman dan perilaku prompt AI.

- Panjang template prompt maksimum: Pengaturan ini menentukan panjang maksimum (dalam karakter) dari template prompt. Nilai yang lebih tinggi memungkinkan lebih banyak konteks diberikan pada model AI, yang berpotensi mengarah pada respons yang lebih akurat. Namun, petunjuk yang terlalu lama juga dapat menimbulkan kebisingan dan berdampak negatif pada kinerja. Untuk model Amazon Bedrock, nilai default untuk panjang template prompt maks (dalam karakter) dihitung menggunakan batas token model yang mendasarinya. Jika Anda mengedit dan mengubah nama model dalam Bedrock, tombol 'Reset to default' disorot dan dapat digunakan untuk mengadopsi default model yang baru dipilih. Untuk model Amazon SageMaker AI, nilai default yang wajar disediakan, tetapi Anda disarankan untuk memeriksa model yang mendasarinya dan memilih panjang templat prompt maksimum ini dan panjang teks input yang sesuai. Lihat bagian Tips mengelola batasan token model untuk informasi selengkapnya.
- Panjang teks masukan maksimum: Pengaturan ini membatasi panjang maksimum (dalam karakter) teks input pengguna. Input yang lebih lama mungkin berisi informasi yang tidak relevan, meningkatkan risiko mendapatkan respons yang tidak relevan atau tidak akurat dari model AI.
- Pengeditan Prompt Pengguna: Opsi ini memungkinkan Anda mengaktifkan atau menonaktifkan kemampuan pengguna untuk memodifikasi templat prompt melalui UI Obrolan. Menonaktifkan fitur ini dapat membantu menjaga konsistensi dan mencegah perubahan yang tidak diinginkan pada prompt.

Templat cepat

Bagian ini memungkinkan Anda untuk menentukan template prompt aktual yang akan digunakan oleh model AI. Template prompt biasanya mengikuti struktur yang mencakup placeholder untuk berbagai komponen, seperti input pengguna, bagian referensi, dan riwayat obrolan.

- Template prompt: Ini adalah area teks utama tempat Anda dapat menulis atau menempelkan templat prompt yang diinginkan. Template harus dibuat untuk memberikan konteks dan instruksi yang diperlukan untuk model AI. Ini biasanya mencakup placeholder berikut:

  - `{input}`: Placeholder ini wajib untuk penerapan Sagemaker AI dan akan diganti dengan input atau kueri pengguna.
  - `{history}`: Placeholder ini wajib untuk penerapan Sagemaker AI dan akan diganti dengan riwayat obrolan percakapan saat ini.
  - `{context}`: Placeholder ini wajib untuk penerapan RAG dan akan diganti dengan kutipan dokumen yang diperoleh dari basis pengetahuan yang dikonfigurasi.
- Ulangi Pertanyaan? : Opsi ini (hanya tersedia untuk penerapan RAG) menentukan apakah kueri input asli pengguna harus diulang atau disambiguasi sebelum diteruskan ke model AI. Mengulangi kueri terkadang dapat membantu model lebih memahami maksud pengguna, yang berpotensi mengarah ke respons yang lebih akurat.

Saat mengonfigurasi templat dan pengalaman prompt, penting untuk menyeimbangkan antara memberikan konteks dan instruksi yang memadai untuk model AI sambil menghindari informasi yang terlalu panjang atau tidak relevan yang dapat menimbulkan masalah kebisingan atau kinerja.

Pengaturan prompt lanjutan

Bagian ini memungkinkan Anda untuk mengontrol bagaimana riwayat percakapan disajikan ke model AI.

- Ukuran riwayat trailing: Pengaturan ini menentukan jumlah pesan sebelumnya yang harus disertakan dalam prompt akhir. Menyetel nilai ini ke nol akan menghasilkan tidak ada riwayat yang disuntikkan ke templat prompt atau templat prompt disambiguasi. Harap dicatat: bahkan ketika disetel ke nol, placeholder {history} masih harus ada di templat prompt. Saat runtime, itu akan diganti dengan string kosong.
  - Catatan: Disarankan untuk memberikan angka genap untuk nilai ini. Memberikan angka ganjil hanya akan menghasilkan respons AI dari interaksi berpasangan yang dikembalikan.
- Awalan Manusia: Ini adalah awalan yang digunakan untuk mengidentifikasi pesan yang dikirim oleh pengguna dalam riwayat percakapan.
- Awalan AI: Ini adalah awalan yang digunakan untuk mengidentifikasi pesan yang dikembalikan oleh model AI dalam riwayat percakapan.

Konfigurasi Prompt Disambiguasi

Bagian ini memungkinkan Anda mengonfigurasi perilaku dan templat untuk menyingkapkan input pengguna sebelum mengirimnya ke basis pengetahuan yang dikonfigurasi.

- Aktifkan Disambiguasi: Opsi ini menentukan apakah input pengguna harus disambiguasi sebelum dikirim ke basis pengetahuan.
- Template Prompt Disambiguasi: Ini adalah template prompt yang digunakan untuk menyamarkan input pengguna saat terhubung ke basis pengetahuan. Output yang dihasilkan dari prompt ini akan digunakan sebagai kueri yang dikirim ke basis pengetahuan. Menonaktifkan disambiguasi akan mengakibatkan kueri mentah pengguna dikirim ke basis pengetahuan tidak berubah.

Misalnya, dengan disambiguasi diaktifkan, kueri pengguna tindak lanjut “Berapa biayanya?” mungkin disambiguasi menjadi “Berapa biayanya perbarui plat nomor saya?” , mengarah ke permintaan pencarian yang lebih baik.

Menggunakan kasus penggunaan Teks yang diterapkan

UI bawaan untuk kasus penggunaan Teks dimaksudkan untuk memungkinkan pengguna bisnis menjelajahi dan bereksperimen dengan cepat dengan penerapan yang dibuat oleh pengguna admin. Perubahan konfigurasi yang dibuat oleh pengguna bisnis hanya berlaku untuk sesi mereka. Pengguna bisnis harus membagikan perubahan ini dengan pengguna admin yang dapat memperbarui penerapan dasar dengan perubahan tersebut untuk digunakan semua orang.

UI obrolan terdiri dari komponen-komponen berikut:

- Jendela obrolan
- Kotak masukan obrolan
- Pengaturan
- Percakapan yang jelas

Jendela obrolan

Memegang belokan percakapan yang berbeda. Pesan yang dimulai di sebelah kanan berasal dari pengguna bisnis, dan pesan yang dimulai di sebelah kiri berasal dari LLM yang dikonfigurasi. Ikon clipboard kecil ada di semua respons LLM untuk memungkinkan penyalinan respons yang mudah.

Kotak masukan obrolan

Disematkan ke bagian bawah jendela obrolan adalah kotak input obrolan. Di sinilah pengguna bisnis dapat memasukkan pesan mereka untuk dikirim ke LLM. Tepat di atas kotak input adalah status koneksi. Jika koneksi terputus (misalnya, karena tidak aktif), koneksi baru secara otomatis dibuat

saat berikutnya pesan obrolan dikirim. Permintaan ini diharapkan memakan waktu sedikit lebih lama karena waktu WebSocket koneksi tambahan.

Berdasarkan konfigurasi tertentu, mungkin ada panjang maksimum yang diberlakukan pada input. Jika batas ini terlampaui, pengguna menerima peringatan dan pesan tidak terkirim.

Catatan: Jika menggunakan RAG dengan Amazon Kendra, Retrieve API akan memotong kueri menjadi 30 kata token. Jika mengharapkan input pengguna yang lebih lama, evaluasi bagaimana hal ini dapat memengaruhi kinerja penelusuran.

Pengaturan

Untuk memungkinkan pengguna bisnis bereksperimen dengan cepat dengan konfigurasi yang berbeda, panel pengaturan tersedia, yang memungkinkan on-the-fly pengeditan opsi konfigurasi penerapan tertentu

(contoh, template prompt). Perubahan ini hanya dapat dilakukan pada awal sesi baru. Setelah percakapan dimulai, menghapus percakapan mengaktifkan kembali pengeditan pengaturan konfigurasi.

Catatan: Pengguna admin dapat memilih untuk mengunci pengaturan penerapan. Mereka dapat mencegah pengeditan langsung pada waktu penerapan melalui wizard selama langkah prompt.

Percakapan yang jelas

Selama percakapan, solusinya mempertahankan riwayat obrolan, yang memungkinkan pengalaman percakapan. Ini memungkinkan disambiguasi kueri dan pertanyaan tindak lanjut. Untuk mengatur ulang percakapan dan menghapus semua riwayat obrolan untuk interaksi ini, pilih *Hapus percakapan*di bagian atas jendela obrolan. Setelah percakapan dihapus, sesi baru dibuat yang memungkinkan pengeditan pengaturan kembali.

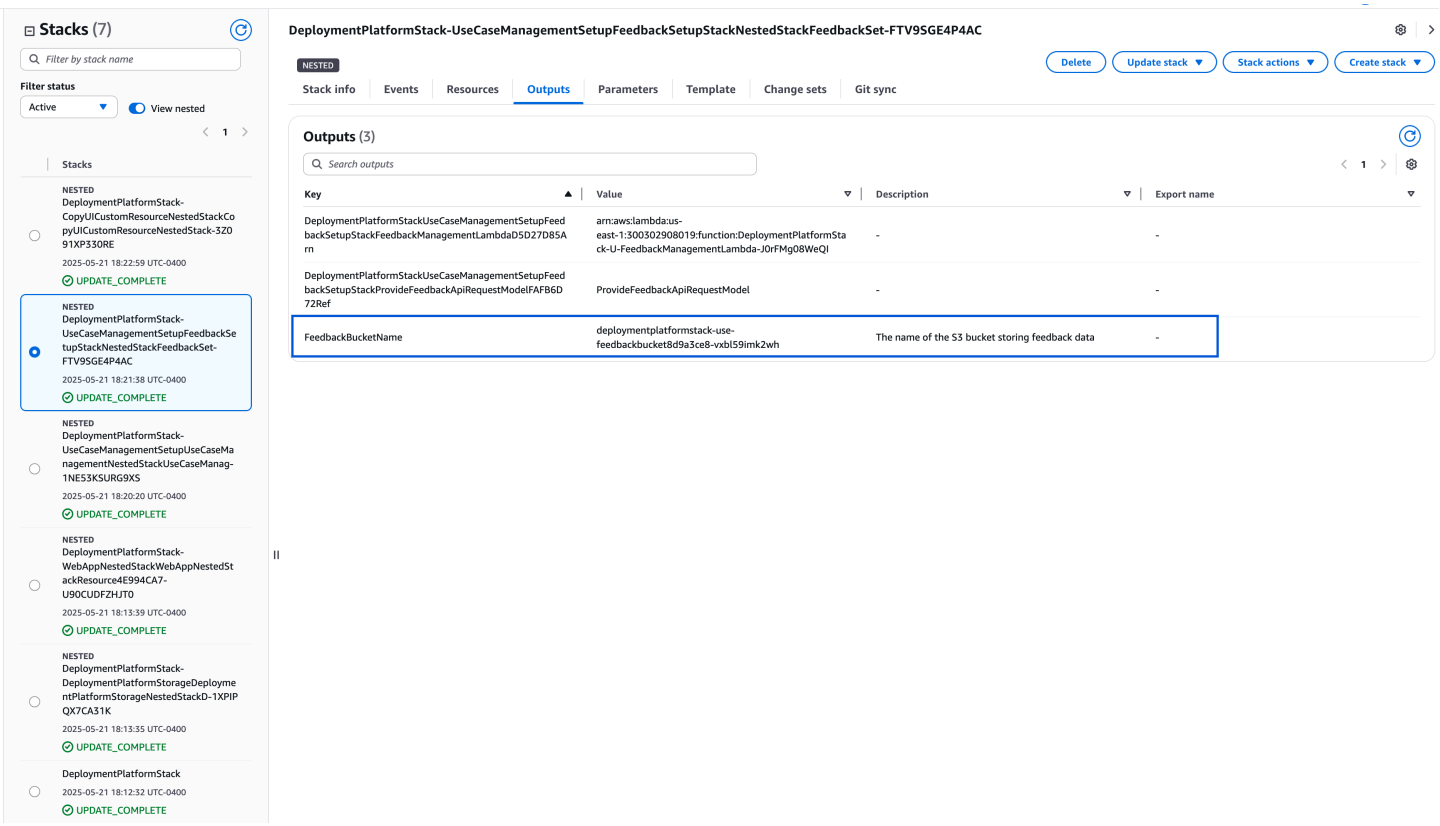
Mengakses dan menganalisis umpan balik yang dikumpulkan pengguna

Mulai v3.0.0, Dasbor Deployment menyebarkan tumpukan umpan balik bersarang yang memungkinkan kasus penggunaan Teks dan Agen Batuan Dasar yang digunakan dengan Dasbor memiliki fungsionalitas pengumpulan umpan balik untuk respons yang dihasilkan. LLM/Agent Khususnya, pengguna dapat memberikan umpan balik positif atau negatif bersama dengan komentar opsional. Jika pengguna memberikan umpan balik negatif, mereka selanjutnya dapat memilih salah

satu kategori negatif ini: 'Tidak Akurat', 'Tidak Lengkap atau tidak cukupi', 'Berbahaya' 'Lainnya'. and/or

Setelah pengguna memberikan umpan balik, umpan balik disimpan dalam bucket S3 yang dipartisi oleh Use Case ID, tahun dan bulan. ID Kasus Penggunaan dapat ditemukan di Dasbor Deployment dan bucket Feedback S3 dapat ditemukan di output tumpukan bersarang umpan balik dari tumpukan Dasbor Deployment:

Menggambarkan tumpukan Deployment - Menemukan Nama Bucket Umpan Balik



Umpan balik pengguna dikirim sebagai permintaan API yang berisi sekumpulan informasi minimal:

```
{
  "useCaseRecordKey": "a1b2c3d4-e5f6g7h8",
  "conversationId": "12345678-1234-1234-1234-123456789012",
  "messageId": "87654321-4321-4321-4321-210987654321",
  "rephrasedQuery": "What are the key features of the Generative AI Application Builder
 on AWS?",
  "sourceDocuments": [
    "s3://bucket-name/document1.pdf",
    "s3://bucket-name/document2.pdf"
  ],
```

```
  "feedback": "positive",
  "feedbackReason": [
    "Incomplete or insufficient"
  ],
  "comment": "The response was helpful but could include more details about important
 features."
}
```

Payload ini kemudian diproses oleh lambda menggunakan `useCaseRecordKey` yang mengidentifikasi konfigurasi kasus penggunaan yang benar pada saat penerapan. Konfigurasi ini digunakan untuk mendapatkan detail spesifik untuk umpan balik seperti nama (berisi semua percakapan dan urutan pesan manusia dan AI) yang selanjutnya digunakan untuk mengambil yang sebenarnya `userInput` dan. ConversationTable `llmResponse` Detail tambahan juga dilampirkan pada catatan umpan balik ini seperti `agentId` dan `agentAliasId` untuk kasus penggunaan Agen Batuan Dasar, dan, `modelProviderbedrockModelId`, dll. untuk kasus penggunaan Teks menggunakan konfigurasi ini. Untuk detail tentang cara mengakses konfigurasi ini, lihat bagian Pemetaan Umpan Balik Kustom di bawah ini. Setiap permintaan umpan balik yang masuk disimpan sebagai objek JSON dan catatan umpan balik sampel dapat terlihat seperti ini untuk kasus penggunaan Teks:

```
{
   "useCaseId": "12345678-1234-1234-1234-123456789012",
   "useCaseRecordKey": "c07a2e3b-2f31b1e0",
   "userId": "22345678-1234-1234-1234-123456789012",
   "conversationId": "dd51de5d-5af1-4ec6-91d2-aadf14352109",
   "messageId": "32345678-1234-1234-1234-123456789012",
   "userInput": "What are its key features?",
   "rephrasedQuery": "What are the key features of the Generative AI Application
 Builder on AWS?",
   "llmResponse": "Generative AI Application Builder on AWS can help you build
 production ready enterprise chatbots rapidly.",
   "feedback": "negative",
   "feedbackReason": [
      "Incomplete or insufficient"
   ],
   "comment": "The response was helpful but could include more details about important
 features.",
   "timestamp": "2025-05-22T18:48:08.340Z",
   "feedbackId": "42345678-1234-1234-1234-123456789012",
   "useCaseType": "Text",
   "modelProvider": "Bedrock",
```

```
   "bedrockModelId": "amazon.nova-lite-v1:0",
   "ragEnabled": "false"
}
```

atau seperti ini untuk kasus penggunaan Agen Batuan Dasar:

```
{
   "useCaseId": "12345678-1234-1234-1234-123456789012",
   "useCaseRecordKey": "c07a2e3b-2f31b1e0",
   "userId": "22345678-1234-1234-1234-123456789012",
   "conversationId": "dd51de5d-5af1-4ec6-91d2-aadf14352109",
   "messageId": "32345678-1234-1234-1234-123456789012",
   "userInput": "What are its key features?",
   "llmResponse": "Generative AI Application Builder on AWS can help you build
 production ready enterprise chatbots rapidly.",
   "feedback": "negative",
   "feedbackReason": [
      "Incomplete or insufficient"
   ],
   "comment": "The response was helpful but could include more details about important
 features.",
   "timestamp": "2025-05-22T18:48:08.340Z",
   "feedbackId": "42345678-1234-1234-1234-123456789012",
   "useCaseType": "Agent",
   "agentId": "AHFXUJCAK1",
   "agentAliasId": "KSEDKOS0BL"
}
```

Umpan balik ini kemudian dapat digunakan untuk pemrosesan lebih lanjut, menganalisis, dan memodelkan pelatihan ulang/loop umpan balik. Anda juga dapat menambahkan pemetaan khusus untuk menyempurnakan catatan umpan balik yang disimpan di lambda umpan balik.

Pemetaan Umpan Balik Kustom

Dasbor Deployment berisi `LLMConfigTable` yang dapat ditemukan di output tumpukan Dashboard Deployment dengan kuncinya. `LLMConfigTableName` `LLMConfigTable`berisi konfigurasi untuk setiap usecase berdasarkan pengaturan yang dipilih oleh admin saat menerapkan usecase melalui wizard Deployment Dashboard. Setiap konfigurasi usecase diidentifikasi oleh konfigurasi. `useCaseRecordKey` Berikut adalah contoh catatan konfigurasi usecase di`LLMConfigTable`:

```
{
   "key": "2dd76cfa-bc1a14da",
```

```
   "config": {
      "ConversationMemoryParams": {
        ...
      },
      "FeedbackParams": {
         "CustomMappings": {
            "NumberOfDocs": "$.KnowledgeBaseParams.NumberOfDocs",
            "ScoreThreshold": "$.KnowledgeBaseParams.ScoreThreshold"
         },
         "FeedbackEnabled": true
      },
      "IsInternalUser": "true",
      "KnowledgeBaseParams": {
         "KendraKnowledgeBaseParams": {
            "ExistingKendraIndexId": "d2831033-667f-4539-ab28-e6c7c7c5988b",
            "RoleBasedAccessControlEnabled": false
         },
         "KnowledgeBaseType": "Kendra",
         "NumberOfDocs": 5,
         "ReturnSourceDocs": false,
         "ScoreThreshold": 0.3
      },
      "LlmParams": {
         "BedrockLlmParams": {
            "BedrockInferenceType": "QUICK_START",
            "ModelId": "amazon.nova-lite-v1:0"
         },
         "ModelParams": {},
         "ModelProvider": "Bedrock",
         "PromptParams": {
            ...
         },
         "RAGEnabled": true,
         "Streaming": false,
         "Temperature": 0.1,
         "Verbose": false
      },
      "UseCaseName": "test-rag-usecase",
      "UseCaseType": "Text"
   }
}
```

Jika umpan balik diaktifkan untuk kasus penggunaan, konfigurasi ini akan berisi `FeedbackParams` objek yang memungkinkan `CustomMappings` objek di dalamnya yang dapat menentukan JSONPaths untuk semua bidang tambahan yang akan ditambahkan ke catatan JSON umpan balik yang disimpan di bucket S3 umpan balik. Misalnya, untuk konfigurasi contoh usecase di atas, CustomMappings berisi `NumberOfDocs` dan `ScoreThreshold` JSONPaths tambahan di `CustomMappings` objek yang dimulai dengan `config` sebagai root dari file. JSONPath Dengan konfigurasi ini, setiap catatan JSON yang disimpan dalam bucket S3 umpan balik akan mulai mendapatkan 2 nilai tambahan ini selain dari bidang yang telah disediakan.

Menganalisis data umpan balik

Data umpan balik disimpan di S3 sebagai objek JSON. Berikut adalah beberapa pendekatan untuk membuat data umpan balik ini lebih mudah diakses dan ditindaklanjuti:

Menggunakan AWS Glue dan Amazon Athena

AWS Glue dan Amazon Athena menyediakan cara tanpa server untuk membuat katalog, menanyakan, dan menganalisis data umpan balik Anda.

AWS Glue memungkinkan Anda membuat crawler AWS Glue yang memeriksa data dalam bucket S3, menyimpulkan skema, dan mencatat semua metadata yang relevan dalam katalog. Posting itu, layanan seperti Amazon Athena dapat digunakan untuk menanyakan data.

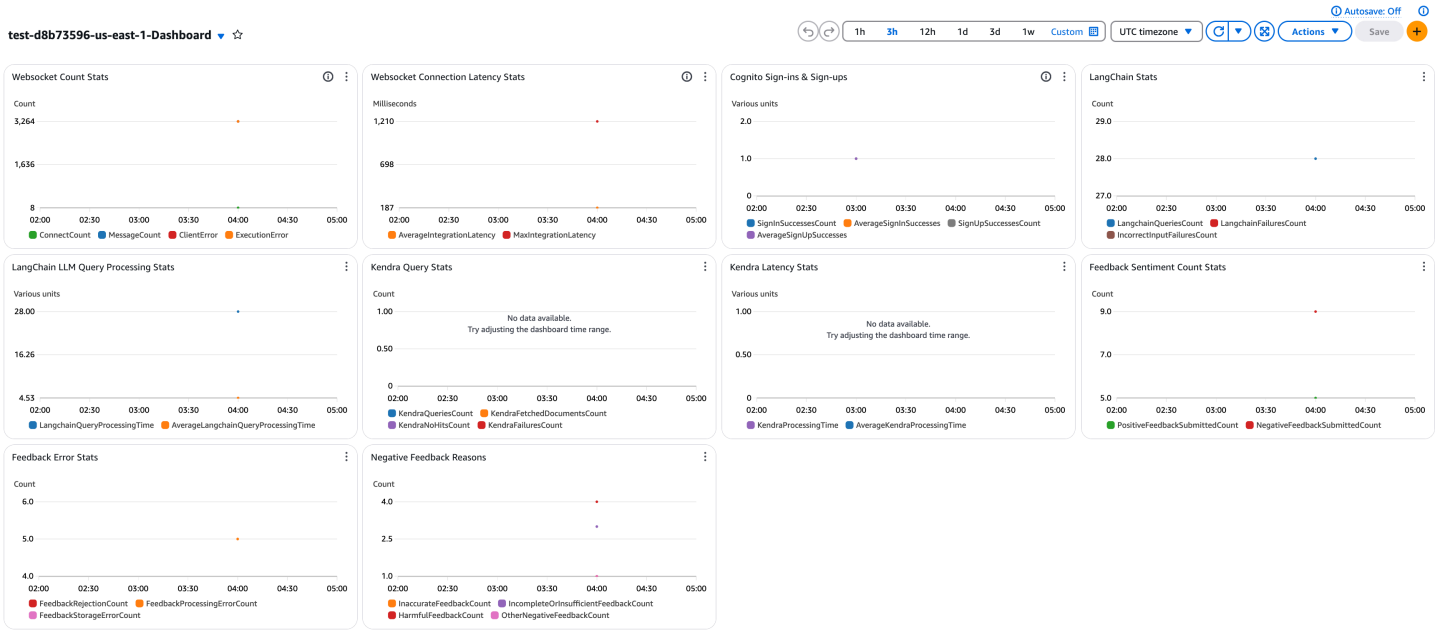
Anda dapat merujuk Dokumentasi AWS Athena tentang langkah-langkah untuk menghubungkan bucket S3 umpan balik dengan Amazon Athena menggunakan AWS Glue Data Catalog. Anda juga dapat menggunakan beberapa fitur Glue yang lebih canggih untuk melakukan pekerjaan Extract Transform & Load (ETL) pada data ini dan mengubahnya menjadi format yang sesuai dengan analisis atau kasus penggunaan pelatihan ulang model Anda. Dengan Glue, Anda dapat melakukan operasi seperti memfilter catatan dengan jenis umpan balik tertentu, mengisi informasi yang hilang, dan Anda juga dapat memuat data ini ke lokasi penyimpanan lain seperti bucket S3 lain atau penyimpanan data AWS yang berbeda.

> **Note**
>
> Bergantung pada kasus penggunaan Anda, pertimbangkan untuk menjadwalkan crawler Glue untuk berjalan secara berkala (misalnya, mingguan) daripada setiap malam untuk mengoptimalkan biaya karena data umpan balik dapat jarang.

Menggunakan CloudWatch Dasbor solusi

Anda juga memiliki akses ke CloudWatch Dasbor yang dikemas dengan solusi yang dapat memberi Anda tren umpan balik positif dan negatif, kategori alasan umpan balik negatif, dll berdasarkan kasus penggunaan. Anda dapat menemukan dasbor ini menggunakan nama usecase Anda di Dasbor di dalam konsol CloudWatch AWS:

Menggambarkan Dasbor Usecase CloudWatch



Anda juga dapat membuat widget tambahan di Dasbor ini atau membuat dasbor Amazon Quick Sight.

Praktik terbaik untuk analisis data umpan balik

- Menerapkan kebijakan siklus hidup data pada bucket S3 Anda untuk mengarsipkan data umpan balik yang lebih lama ke tingkatan penyimpanan berbiaya lebih rendah
- Buat analisis terpisah untuk setiap kasus penggunaan untuk mengidentifikasi peluang peningkatan spesifik model
- Tetapkan ambang umpan balik yang memicu peringatan ketika umpan balik negatif melebihi tingkat yang dapat diterima
- Ekspor wawasan kritis secara berkala untuk berbagi dengan pemangku kepentingan dan tim peningkatan model

Melihat metrik operasi untuk penerapan

Dasbor Deployment dan tumpukan kasus penggunaan masing-masing dilengkapi dengan CloudWatch dasbor mereka sendiri yang melacak berbagai metrik operasional solusi. Anda dapat menggunakan CloudWatch dasbor ini untuk membantu membandingkan penerapan yang berbeda. Untuk mengakses dasbor:

1. Navigasikan ke konsol CloudWatch tersebut.
2. Cari dasbor pra-bangun baik dengan mencari nama tumpukan, atau pengidentifikasi unik universal (UUID).

Misalnya, kasus penggunaan Teks dilengkapi dengan grafik yang melacak jumlah WebSocket koneksi, jumlah pengguna masuk dan mendaftar, jumlah waktu yang dibutuhkan LLM untuk memproses penyelesaian, dan sebagainya. Pelanggan dapat menggunakan grafik ini untuk membandingkan berbagai _kuantitatif _metrik penerapan.

Example

Sulit untuk membandingkan hasil kualitatif dari berbagai model yang diterapkan pada kasus penggunaan yang berbeda. Gunakan fitur Clone untuk memutar beberapa penerapan dengan cepat sehingga Anda dapat membandingkan output secara berdampingan.

Akses wawasan CloudWatch Log

Solusi ini mencatat pesan kesalahan, peringatan, informasi, dan debugging untuk fungsi Lambda. Untuk memilih jenis pesan yang akan dicatat:

1. Temukan fungsi yang berlaku di konsol AWS Lambda.
2. Tambahkan variabel lingkungan POWERTOOLS_LOG_LEVEL.
3. Atur variabel ke jenis pesan yang berlaku.

Untuk petunjuk lebih lanjut, lihat Membuat variabel lingkungan Lambda di Panduan Pengembang AWS Lambda.

Tabel berikut mencantumkan jenis tingkat log yang dapat Anda pilih.

Tingkat	Deskripsi
KESALAHAN	Log mencakup informasi tentang apa pun yang menyebabkan operasi gagal.
PERINGATAN	Log mencakup informasi tentang apa pun yang berpotensi menyebabkan ketidakkonsistenan dalam fungsi tetapi mungkin tidak selalu menyebabkan operasi gagal. Log juga menyertakan pesan ERROR.
INFO	Log mencakup informasi tingkat tinggi tentang bagaimana fungsi beroperasi. Log juga menyertakan pesan ERROR dan PERINGATAN.
DEBUG	Log menyertakan informasi yang mungkin berguna saat men-debug masalah dengan fungsi tersebut. Log juga menyertakan pesan ERROR, WARNING, dan INFO.

Gunakan prosedur berikut untuk menambahkan wawasan CloudWatch Log ke solusi ini.

1. Identifikasi grup log yang relevan:
   a. Masuk ke CloudFormation konsol AWS.
   b. Pilih tumpukan target Anda.
   c. Pilih tab Sumber Daya dan cari fungsi Lambda target Anda.
   d. Masuk ke konsol AWS Lambda dan pilih masing-masing fungsi Lambda target Anda.
   e. Untuk setiap fungsi Lambda target Anda, pilih tab Monitor dan pilih Lihat CloudWatch Log.
   f. Salin nama grup log yang ingin Anda ekstrak wawasannya.
2. Arahkan ke CloudWatch konsol Amazon.
3. Pada menu navigasi, di bawah Log, pilih Wawasan Log.
4. Pada halaman Wawasan Log, pilih tab Log.
5. Cari nama grup log dari langkah 1.
6. Salin salah satu contoh kueri berikut dan tempelkan ke bidang kueri:

a. Untuk mengidentifikasi semua pengecualian klien:

```
fields @message
|filter @message like /(?i)Exception/|stats count(*) as exceptionCount by @message
```

b. Untuk mengambil jumlah pemanggilan dengan nama fungsi:

```
stats count(*) by function_name
```

c. Untuk mengambil jumlah pemanggilan selama interval lima menit:

```
stats count(*) as invocations by bin(5m)
```

d. Untuk mengambil semua jejak IDsAWS X-Ray:

```
filter @message like "XRAY TraceId"
|parse @message "XRAY TraceId: * " as traceId|stats count(*) by traceId
```

e. Untuk mengambil log yang berkaitan dengan ID Jejak X-Ray tertentu:

```
filter @message like "your-traceid-here"
```

f. Untuk mengambil kesalahan yang tidak sah WebSocket :

```
fields
@ingestionTime,
@log,
@logStream,
@message,
@requestId,
@timestamp,
errorMessage,
errorType
|filter @message like /Unauthorized/ and @message like /websocket/|sort @timestamp
 desc
```

g. Untuk mengambil jumlah metrik yang diterbitkan:

```
filter @message like "CloudWatchMetrics"
|parse @message /"Metrics":\s*\[(?<metrics>.*?)\]/|stats count(*) as metric_count
 by metrics
```

Panduan developer

Bagian ini menyediakan kode sumber untuk solusi, panduan integrasi, panduan penyesuaian, dan referensi API.

Kode sumber

Kunjungi GitHub repositori kami untuk mengunduh file sumber untuk solusi ini dan untuk berbagi penyesuaian Anda dengan orang lain.

Generative AI Application Builder pada template AWS dibuat menggunakan AWS Cloud Development Kit (AWS CDK). Lihat file README.md untuk informasi tambahan.

Panduan integrasi

Seluruh solusi dirancang agar mudah diperluas. Lapisan orkestrasi dari solusi ini dibangun menggunakan. LangChain Anda dapat menambahkan penyedia model, basis pengetahuan, atau jenis memori percakapan apa pun yang didukung oleh LangChain (atau pihak ketiga yang menyediakan LangChain konektor untuk komponen ini) ke solusi ini.

Memperluas didukung LLMs

Untuk menambahkan penyedia model lain, seperti penyedia LLM khusus, Anda harus memperbarui tiga komponen solusi berikut:

1. Buat tumpukan `TextUseCase` CDK baru, yang menyebarkan aplikasi obrolan yang dikonfigurasi dengan penyedia LLM kustom Anda:
   a. Kloning GitHub repositori solusi ini, dan siapkan lingkungan build Anda dengan mengikuti instruksi yang diberikan dalam file README.md.
   b. Salin (atau buat baru) `source/infrastructure/lib/bedrock-chat-stack.ts` file, tempel ke direktori yang sama, dan ganti namanya menjadi`custom-chat-stack.ts`.
   c. Ganti nama kelas dalam file menjadi yang sesuai, seperti`CustomLLMChat`.
   d. Anda dapat memilih untuk menambahkan rahasia Secrets Manager ke tumpukan ini, yang menyimpan kredensi Anda untuk LLM kustom Anda. Anda dapat mengambil kredensi ini selama pemanggilan model di lapisan Lambda obrolan yang dibahas di paragraf berikutnya.

2. Bangun dan lampirkan layer Lambda yang berisi pustaka Python dari penyedia model yang akan ditambahkan. Untuk aplikasi obrolan kasus penggunaan Amazon Bedrock, pustaka `langchain-aws` Python berisi konektor khusus di atas paket untuk terhubung ke penyedia LangChain model AWS (Amazon Bedrock SageMaker dan AI), basis pengetahuan (Amazon Kendra dan Amazon Bedrock Knowledge Bases), dan jenis memori (seperti DynamoDB). Demikian pula, penyedia model lain memiliki konektor sendiri. Lapisan ini membantu Anda melampirkan pustaka Python penyedia model ini sehingga Anda dapat menggunakan konektor ini di lapisan Lambda obrolan, yang memanggil LLM (langkah 3). Dalam solusi ini, bundler aset khusus digunakan untuk membangun lapisan Lambda, yang dilampirkan menggunakan aspek CDK. Untuk membuat layer baru untuk pustaka penyedia model kustom:

   a. Arahkan ke `LambdaAspects` kelas dalam `source/infrastructure/lib/utils/lambda-aspects.ts` file.

   b. Ikuti petunjuk tentang cara memperluas fungsionalitas kelas aspek Lambda yang disediakan dalam file (seperti menambahkan `getOrCreateLangchainLayer` metode). Untuk menggunakan metode baru ini (misalnya,`getOrCreateCustomLLMLayer`), perbarui juga `LLM_LIBRARY_LAYER_TYPES` enum dalam `source/infrastructure/lib/utils/constants.ts` file.

3. Perluas fungsi `chat` Lambda untuk mengimplementasikan pembangun, klien, dan penangan untuk penyedia baru.

   `source/lambda/chat`Berisi LangChain koneksi untuk berbeda LLMs bersama dengan kelas pendukung untuk membangun ini LLMs. Kelas pendukung ini mengikuti pola desain Builder dan Object Oriented untuk membuat LLM.

   Setiap handler (misalnya,`bedrock_handler.py`) pertama membuat klien, memeriksa lingkungan untuk variabel lingkungan yang diperlukan, dan kemudian memanggil `get_model` metode untuk mendapatkan kelas LangChain LLM. Metode generate kemudian dipanggil untuk memanggil LLM dan mendapatkan responsnya. LangChain saat ini mendukung fungsionalitas streaming untuk Amazon Bedrock, tetapi tidak SageMaker AI. Berdasarkan fungsionalitas streaming atau non-streaming, WebSocket handler yang sesuai (`WebsocketStreamingCallbackHandler`atau`WebsocketHandler`) dipanggil untuk mengirim respons kembali ke WebSocket koneksi menggunakan metode ini`post_to_connection`.

   `clients/builder`Folder berisi kelas yang membantu membangun LLM Builder menggunakan pola Builder. Pertama, a `use_case_config` diambil dari toko konfigurasi DynamoDB, yang menyimpan detail tentang jenis basis pengetahuan, memori percakapan, dan model apa yang akan dibangun. Ini juga berisi detail model yang relevan seperti parameter model dan petunjuk.

Builder kemudian membantu mengikuti langkah-langkah untuk membuat basis pengetahuan, membuat memori percakapan untuk mempertahankan konteks percakapan untuk LLM, mengatur LangChain panggilan balik yang sesuai untuk kasus streaming dan non-streaming, dan membuat model LLM berdasarkan konfigurasi model yang disediakan. Konfigurasi DynamoDB disimpan pada saat pembuatan kasus penggunaan saat Anda menerapkan kasus penggunaan dari dasbor Deployment (atau saat disediakan oleh pengguna dalam penerapan tumpukan kasus penggunaan mandiri tanpa dasbor Deployment).

`clients/factories`Subfolder membantu mengatur memori percakapan dan kelas basis pengetahuan yang sesuai, berdasarkan konfigurasi LLM. Ini memungkinkan ekstensi mudah ke basis pengetahuan atau jenis memori lain yang Anda ingin implementasi Anda dukung.

`shared`Subfolder berisi implementasi spesifik dari basis pengetahuan dan memori percakapan yang dipakai di dalam pabrik oleh pembangun. Ini juga berisi Amazon Kendra dan Amazon Bedrock Knowledge Base retriever yang dipanggil dalam LangChain untuk mengambil dokumen untuk kasus penggunaan RAG, bersama dengan callback, yang digunakan oleh model LLM. LangChain

LangChain Implementasi menggunakan LangChain Expression Language (LCEL) untuk menyusun rantai percakapan bersama-sama. `RunnableWithMessageHistory`class digunakan untuk memelihara riwayat percakapan dengan rantai LCEL khusus, memungkinkan fungsionalitas seperti mengembalikan dokumen sumber dan menggunakan pertanyaan yang diulang (atau disambiguasi) yang dikirim ke basis pengetahuan untuk juga dikirim ke LLM.

Untuk membuat implementasi sendiri dari penyedia kustom, Anda dapat:

a. Salin `bedrock_handler.py` file dan buat handler kustom Anda (misalnya,`custom_handler.py`), yang membuat klien kustom Anda (misalnya,`CustomProviderClient`) (ditentukan dalam langkah berikut.)
b. `bedrock_client.py`Salin di folder klien. Ubah nama menjadi `custom_provider_client.py` (atau nama penyedia model spesifik Anda, seperti`CustomProvider`). Beri nama kelas di dalamnya dengan tepat, seperti `CustomProviderClient` yang mewarisi`LLMChatClient`.

   Anda dapat menggunakan metode yang disediakan oleh `LLMChatClient` atau menulis implementasi Anda sendiri untuk mengganti ini.

`get_model`Metode ini membangun `CustomProviderBuilder` (lihat langkah berikut), dan memanggil `construct_chat_model` metode yang membangun model obrolan menggunakan langkah-langkah pembangun. Metode ini bertindak sebagai Direktur dalam pola pembangun.

c. Salin `clients/builders/bedrock_builder.py` dan ganti namanya menjadi `custom_provider_builder.py` dan kelas di dalamnya menjadi `CustomProviderBuilder` yang mewarisi LLMBuilder ()`llm_builder.py`. Anda dapat menggunakan metode yang disediakan oleh LLMBuilder atau menulis implementasi Anda sendiri untuk mengganti ini. Langkah-langkah pembangun dipanggil secara berurutan di dalam `construct_chat_model` metode klien, seperti`set_model_defaults`,`set_knowledge_base`, dan`set_conversation_memory`.

   `set_llm_model`Metode ini akan membuat model LLM aktual menggunakan semua nilai yang ditetapkan menggunakan metode yang dipanggil sebelumnya. Secara khusus, Anda dapat membuat LLM RAG (`CustomProviderRetrievalLLM`) atau non-RAG (`CustomProviderLLM`), berdasarkan `rag_enabled variable` yang diambil dari konfigurasi LLM di DynamoDB.

   Konfigurasi ini diambil dalam `retrieve_use_case_config` metode di `LLMChatClient` kelas.

d. Terapkan `CustomProviderRetrievalLLM` implementasi `CustomProviderLLM` atau implementasi Anda di `llm_models` subfolder berdasarkan apakah Anda memerlukan kasus penggunaan RAG atau non-RAG. Sebagian besar fungsionalitas untuk mengimplementasikan model ini disediakan di `RetrievalLLM` kelasnya masing-masing, untuk kasus penggunaan non-RAG dan RAG. `BaseLangChainModel`

   Anda dapat menyalin `llm_models/bedrock.py` file dan membuat perubahan yang diperlukan untuk memanggil LangChain model yang merujuk ke penyedia khusus Anda. Misalnya, Amazon Bedrock menggunakan `ChatBedrock` kelas untuk membuat model obrolan menggunakan LangChain.

   Metode generate menghasilkan respons LLM menggunakan rantai LangChain LCEL.

   Anda juga dapat menggunakan `get_clean_model_params` metode ini untuk membersihkan parameter model per LangChain atau persyaratan model Anda.

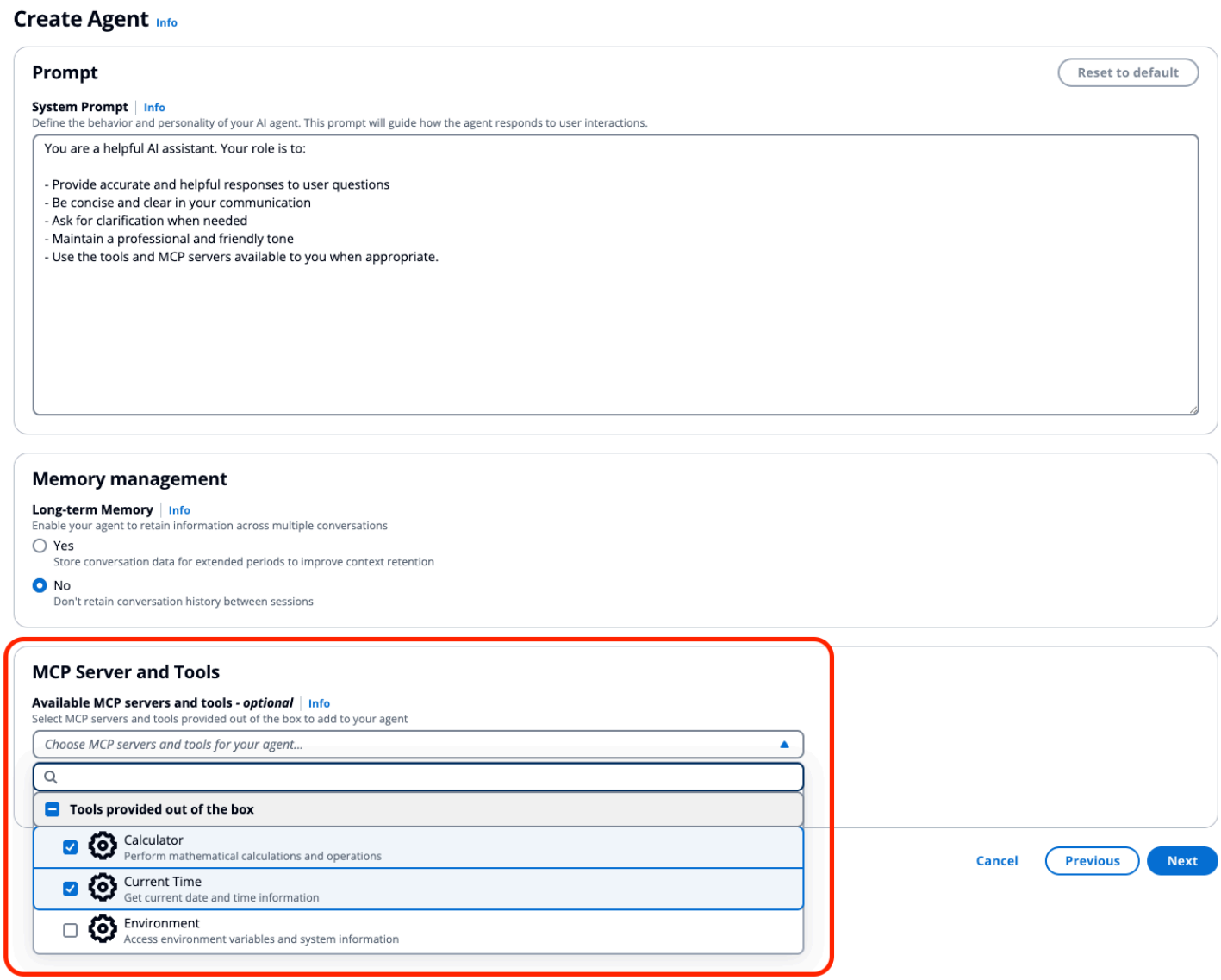
Memperluas alat Strands yang didukung

Solusi ini memungkinkan Anda untuk membangun dan menyebarkan server MCP, agen AI, dan alur kerja multi-agen. Dalam pengalaman Agen Builder, Anda dapat melampirkan server MCP untuk memberi agen Anda kemampuan tambahan. Selain server MCP, Anda dapat memanfaatkan alat bawaan yang disediakan oleh Strands (kerangka kerja dasar yang digunakan oleh solusi).

Di luar kotak, solusinya sudah dikonfigurasi sebelumnya dengan alat Strands berikut:

- Waktu Saat Ini (diaktifkan secara default)
- Kalkulator (diaktifkan secara default)
- Lingkungan

Pemilihan MCP Server dan Tools di wizard Agent Builder yang menampilkan alat Strands bawaan



Untuk memperluas agen Anda dengan alat Strands tambahan, ikuti proses empat langkah yang diuraikan di bagian ini.

Langkah 1: Temukan alat Strands

Jelajahi alat Strands yang tersedia untuk mengidentifikasi alat yang ingin Anda gunakan. Setiap alat memiliki kemampuan dan persyaratan konfigurasi khusus.

Misalnya, untuk menambahkan kemampuan pengambilan Amazon Bedrock Knowledge Base, Anda akan menggunakan alat pengambilan.

Langkah 2: Perbarui parameter SSM

Agar alat tersedia di UI penerapan Agent Builder, perbarui parameter AWS Systems Manager Parameter Store yang menentukan alat Strands mana yang didukung.

1. Arahkan ke AWS Systems Manager Parameter Store di akun AWS Anda.
2. Temukan parameternya: `/gaab/<stack-name>/strands-tools`
3. Tambahkan konfigurasi alat Anda ke akhir daftar yang ada menggunakan struktur JSON berikut:

```
{
  "name": "Bedrock KB Retrieve",
  "description": "Retrieve information from Bedrock Knowledge Base",
  "value": "retrieve",
  "category": "AI",
  "isDefault": false
}
```

Bidang	Deskripsi
name	Nama tampilan yang ditampilkan di UI Agent Builder
deskripsi	Deskripsi singkat tentang fungsionalitas alat
nilai	Nama alat yang tepat seperti yang didefinis ikan dalam paket alat Strands
kategori	Kategori organisasi untuk mengelompokkan alat di UI
isDefault	Apakah alat harus diaktifkan secara default untuk agen baru

Langkah 3: Konfigurasikan variabel lingkungan

Banyak alat Strands memerlukan variabel lingkungan untuk konfigurasi. Anda dapat mengatur variabel-variabel ini dengan dua cara:

Opsi 1: Konfigurasi langsung pada AgentCore Runtime

Perbarui agen yang diterapkan secara langsung di Amazon Bedrock AgentCore Runtime dengan variabel lingkungan yang diperlukan.

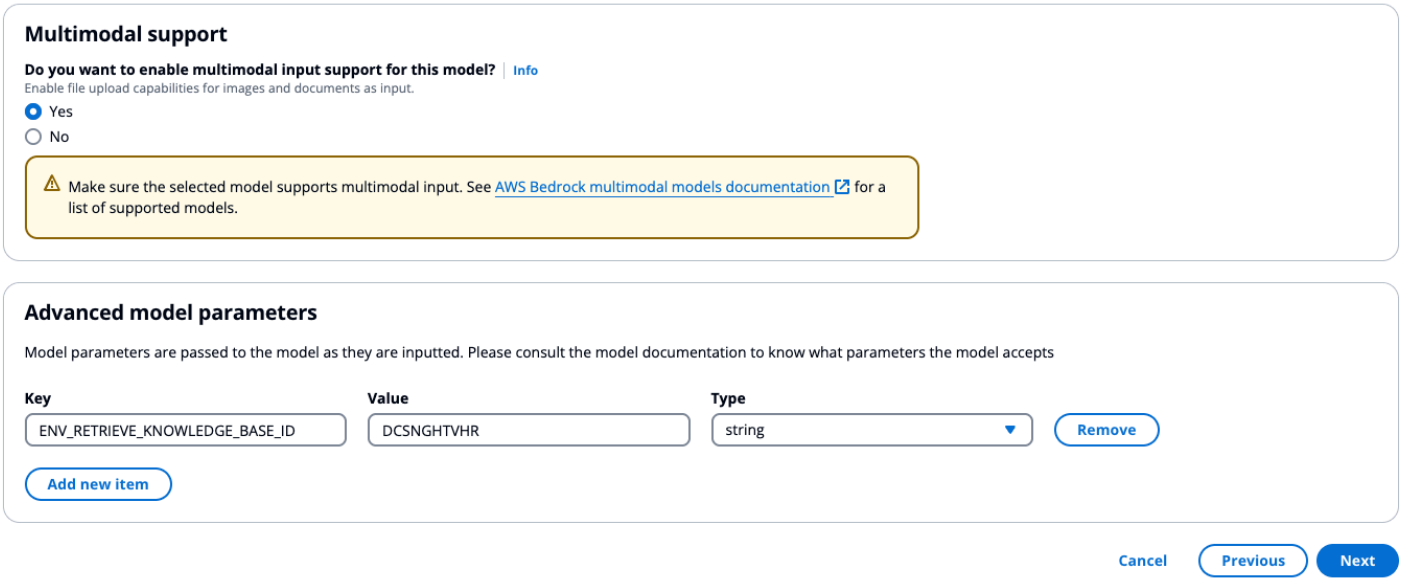
Opsi 2: Parameter Model di wizard penerapan

Tambahkan variabel lingkungan selama langkah pemilihan Model di wizard Agent Builder menggunakan bagian Parameter Model. Variabel lingkungan yang mengikuti konvensi penamaan `ENV_<ALL_CAPS_TOOL_NAME>_<env_variable_name>` akan secara otomatis dimuat saat runtime ke lingkungan eksekusi agen sebagai`<env_variable_name>`.

Contoh:

- `ENV_RETRIEVE_KNOWLEDGE_BASE_ID` menjadi `KNOWLEDGE_BASE_ID`
- `ENV_RETRIEVE_MIN_SCORE` menjadi `MIN_SCORE`

Bagian parameter model lanjutan yang menunjukkan konfigurasi ENV_RETRIEVE_KNOWLEDGE_BASE_ID



Lihat dokumentasi atau kode sumber alat khusus untuk mengidentifikasi variabel lingkungan yang diperlukan. Untuk alat pengambilan, Anda dapat menemukan opsi konfigurasi di kode sumber.

Langkah 4: Tambahkan izin IAM

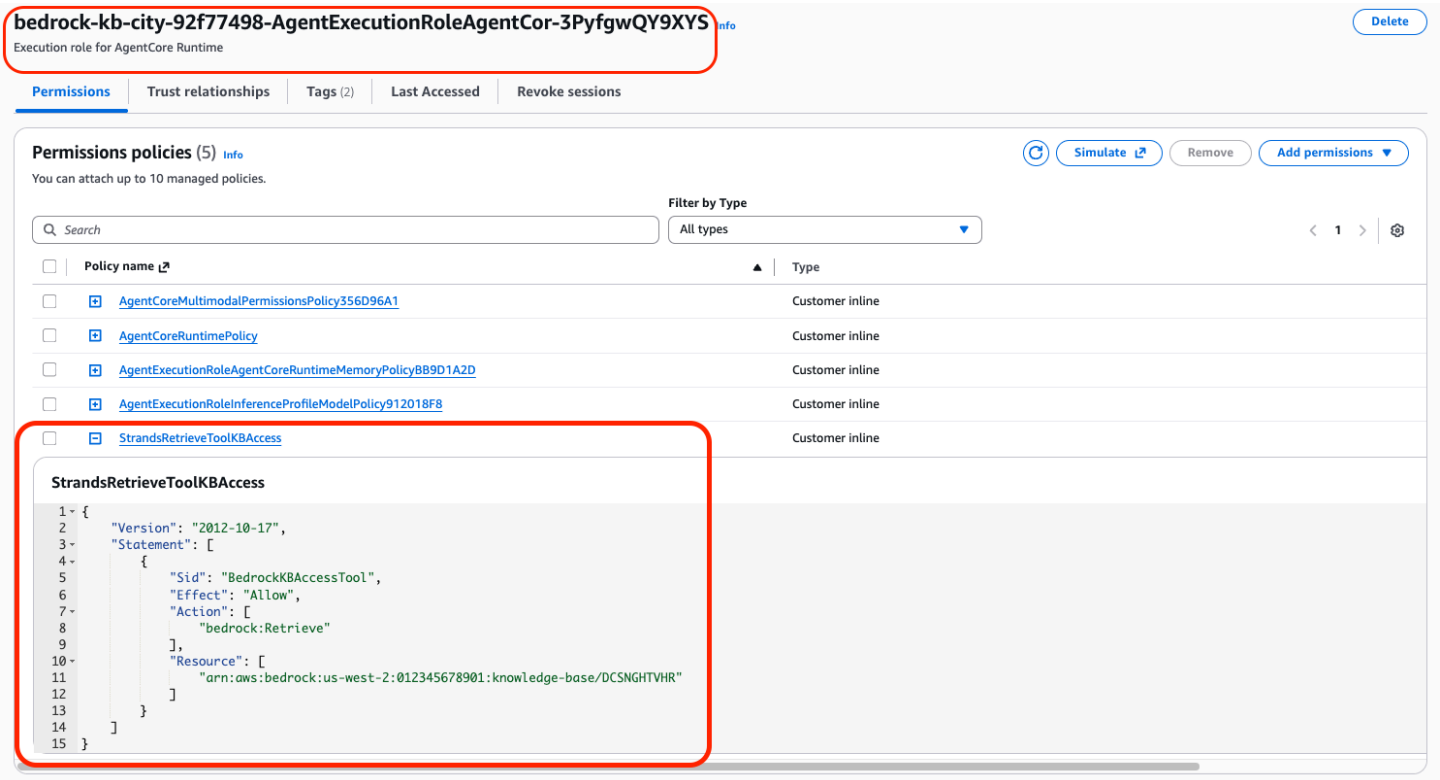
Tambahkan izin IAM yang diperlukan secara manual ke peran eksekusi AgentCore Runtime Anda untuk memungkinkan agen menggunakan alat ini.

Misalnya, untuk menggunakan alat pengambilan dengan Pangkalan Pengetahuan Amazon Bedrock:

1. Arahkan ke konsol IAM di akun AWS Anda.
2. Temukan peran eksekusi AgentCore Runtime untuk agen Anda.
3. Tambahkan izin berikut:

```
{
  "Effect": "Allow",
  "Action": "bedrock:Retrieve",
  "Resource": "arn:aws:bedrock:region:account-id:knowledge-base/knowledge-base-id"
}
```

Konsol IAM yang menampilkan StrandsRetrieveTool KBAccess kebijakan yang dilampirkan pada peran eksekusi AgentCore Runtime

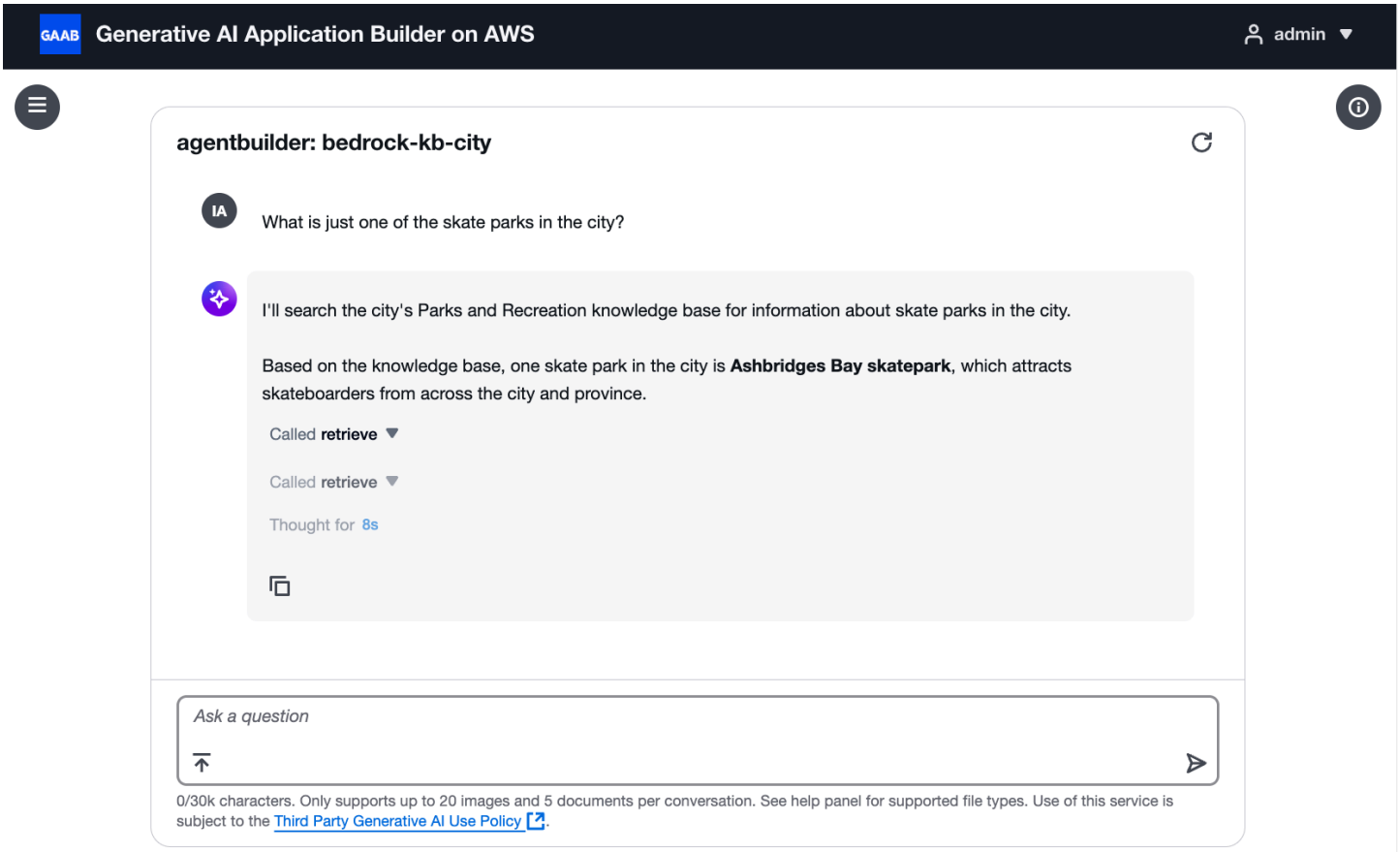


Izin khusus yang diperlukan akan bervariasi berdasarkan alat. Lihat dokumentasi alat dan dokumentasi layanan AWS untuk menentukan izin IAM yang sesuai.

Langkah 5: Uji agen

Setelah menyelesaikan langkah-langkah konfigurasi, uji agen Anda untuk memverifikasi alat berfungsi dengan benar. Anda akan melihat pemanggilan alat di log eksekusi dan tanggapan agen.

Agen berhasil menggunakan alat pengambilan untuk menjawab pertanyaan tentang taman skate



> **Note**
>
> Untuk daftar lengkap alat Strands yang tersedia dan kemampuannya, lihat dokumentasi Strands Community Tools.

Memperluas basis pengetahuan yang didukung dan jenis memori percakapan

Untuk menambahkan implementasi memori percakapan atau basis pengetahuan Anda, tambahkan implementasi yang diperlukan di `shared` folder dan kemudian edit pabrik dan pencacahan yang sesuai untuk membuat instance kelas ini.

Saat Anda menyediakan konfigurasi LLM, yang disimpan di dalam penyimpanan parameter, memori percakapan dan basis pengetahuan yang sesuai akan dibuat untuk LLM Anda. Misalnya, ketika `ConversationMemoryType` ditentukan sebagai DynamoDB, sebuah instance `DynamoDBChatMessageHistory` dari (tersedia `shared_components/memory/ddb_enhanced_message_history.py` di dalam) dibuat. Ketika `KnowledgeBaseType` ditentukan sebagai Amazon Kendra, instance `KendraKnowledgeBase` (tersedia di dalam`shared_components/knowledge/kendra_knowledge_base.py`) dibuat.

Membangun dan menerapkan perubahan kode

Bangun program dengan `npm run build` perintah. Setelah kesalahan diselesaikan, jalankan `cdk synth` untuk menghasilkan file template dan semua aset Lambda.

1. Anda dapat menggunakan `0/stage-assets.sh` skrip untuk secara manual mementaskan aset apa pun yang dihasilkan ke bucket staging di akun Anda.
2. Gunakan perintah berikut untuk menyebarkan atau memperbarui platform:

   ```
   cdk deploy DeploymentPlatformStack --parameters AdminUserEmail='admin-email@amazon.com'
   ```

   CloudFormation Parameter AWS tambahan apa pun juga harus disertakan bersama dengan AdminUserEmailparameternya.

Panduan kustomisasi

Mengelola kumpulan pengguna Cognito

Saat dasbor Deployment di-deploy, kumpulan pengguna Amazon Cognito bersama dengan pengguna admin dibuat untuk menyediakan otentikasi aplikasi. Kumpulan pengguna ini dibagikan di dasbor Deployment dan semua kasus penggunaan. Pengguna admin yang dibuat saat penerapan dasbor secara otomatis diberikan akses ke semua kasus penggunaan yang digunakan menggunakan dasbor. Mekanisme ini disediakan melalui grup kumpulan pengguna Amazon Cognito.

Ketika kasus penggunaan disebarkan dari dasbor, jika email disediakan, pengguna akan dibuat di kumpulan pengguna bersama, bersama dengan grup pengguna yang diberi nama untuk kasus penggunaan tertentu. Pengguna yang baru dibuat kemudian ditambahkan ke grup, memberikan pengguna akses ke kasus penggunaan.

Jika Anda ingin menambahkan pengguna tambahan ke kasus penggunaan tertentu, ini dapat dicapai dengan membuat pengguna di kumpulan pengguna Cognito dan menambahkannya ke grup yang sesuai dengan kasus penggunaan yang Anda inginkan agar pengguna memiliki akses ke. Untuk step-by-step panduan, lihat Membuat pengguna baru di AWS Management Console.

Demikian pula, jika Anda ingin membuat pengguna admin tambahan, Anda harus membuat pengguna baru dan menambahkannya ke grup Admin di kumpulan pengguna.

Nama pengguna dibuat dengan mengambil bagian dari email yang disediakan sebelum@, dan menambahkan UUID kasus penggunaan yang dihasilkan (atau `-admin` dalam kasus pengguna admin).

Di tab Grup, Anda dapat melihat bahwa grup Admin dan grup untuk setiap kasus penggunaan telah dibuat secara otomatis menggunakan nama kasus penggunaan (seperti yang disediakan dalam wizard) dan UUID kasus penggunaan.

Referensi API

Bagian ini menyediakan referensi API untuk solusinya.

Dasbor penyebaran

API REST	Metode HTTP	Fungsionalitas	Penelepon resmi
`/deployments`	`GET`	Dapatkan semua penerapan.	Token JWT yang diautentikasi Amazon Cognito
`/deployments`	`POST`	Membuat penerapan kasus penggunaan baru.	Token JWT yang diautentikasi Amazon Cognito
`/deployments/{useCaseId}`	`GET`	Mendapat detail penerapan untuk satu penerapan.	Token JWT yang diautentikasi Amazon Cognito
`/deployments/{useCaseId}`	`PATCH`	Memperbarui penerapan yang diberikan.	Token JWT yang diautentikasi Amazon Cognito

API REST	Metode HTTP	Fungsionalitas	Penelepon resmi
`/deployments/{useCaseId}`	`DELETE`	Menghapus penerapan yang diberikan.	Token JWT yang diautentikasi Amazon Cognito
`/model-info/use-case-types`	`GET`	Mendapatkan tipe kasus penggunaan yang tersedia untuk penerapan	Token JWT yang diautentikasi Amazon Cognito
`/model-info/{useCaseType}/providers`	`GET`	Mendapatkan penyedia model yang tersedia untuk jenis kasus penggunaan yang diberikan	Token JWT yang diautentikasi Amazon Cognito
`/model-info/{useCaseType}/{providerName}`	`GET`	Mendapatkan IDs model yang tersedia untuk penyedia tertentu dan tipe kasus penggunaan	Token JWT yang diautentikasi Amazon Cognito
`/model-info/{useCaseType}/{providerName}/{modelId}`	`GET`	Mendapat info tentang model yang diberikan, termasuk parameter default.	Token JWT yang diautentikasi Amazon Cognito

> **Note**
>
> File OpenAPI dan Swagger juga dapat diekspor dari API Gateway untuk integrasi yang lebih mudah dengan API. Lihat Mengekspor REST API dari API Gateway.

POST dan PATCH Muatan

Lihat di bawah untuk contoh muatan POST ke `/deployments` titik akhir, yang akan membuat kasus penggunaan baru.

```
{
 "UseCaseName": "usecase1",
 "UseCaseDescription": "Description of the use case to be deployed. For display
 purposes", // optional
 "DefaultUserEmail": "placeholder@example.com", // optional, if not provided, the
 Cognito Group and User will not be created
 "DeployUI": true, // optional
 "VpcParams": {
 "VpcEnabled": true,
 "CreateNewVpc": false,
 // provide these if not creating new vpc
 "ExistingVpcId": "vpc-id",
 "ExistingPrivateSubnetIds": ["subnet-1", "subnet-2"],
 "ExistingSecurityGroupIds": ["sg-1", "sg-2"]
 },
 "ConversationMemoryParams": {
 "ConversationMemoryType": "DynamoDB",
 "HumanPrefix": "user", // optional
 "AiPrefix": "ai", // optional
 "ChatHistoryLength": 10 // optional
 },
 "KnowledgeBaseParams": {
 "KnowledgeBaseType": "Bedrock",
 // one of the following based on selected provider
 "BedrockKnowledgeBaseParams": {
 "BedrockKnowledgeBaseId": "my-bedrock-kb",
 "RetrievalFilter": {}, // optional
 "OverrideSearchType": "HYBRID" // optional
 },
 "KendraKnowledgeBaseParams": {
 "AttributeFilter": {}, // optional
 "RoleBasedAccessControlEnabled": true, // optional
 "ExistingKendraIndexId": "12345678-abcd-1234-abcd-1234567890ab",
 // provide the following in place of ExistingKendraIndexId if you want the solution to
 deploy an index for you
 "KendraIndexName": "index",
 "QueryCapacityUnits": 1, // optional
 "StorageCapacityUnits": 1, // optional
 "KendraIndexEdition": "DEVELOPER" // optional
 },
 "NoDocsFoundResponse": "Sorry, I couldn't find any relevant information for your
 query.", // optional
 "NumberOfDocs": 3, // optional
```

```
 "ScoreThreshold": 0.7, // optional
 "ReturnSourceDocs": true // optional
 },
 "LlmParams": {
 "ModelProvider": "Bedrock | SAGEMAKER",
 // one of the following based on selected provider
 "BedrockLlmParams": {
 "ModelId": "model-id", // use this for on demand models. Can't use with ModelArn
 "ModelArn": "model-arn", // use this for provisioned/custom models. Can't use with
 ModelId,
 "InferenceProfileId": "profile-id"
 "GuardrailIdentifier": "arn:aws:bedrock:us-east-1:123456789012:guardrail/my-
guardrail", // optional
 "GuardrailVersion": "1" // optional. Required if GuardrailIdentifier provided.
 },
 "SageMakerLlmParams": {
 "EndpointName": "some-endpoint",
 "ModelInputPayloadSchema": {},
 "ModelOutputJSONPath": "$."
 },
 // optional. Passes on arbitrary params to the underlying LLM.
 "ModelParams": {
 "param1": {
 "Value": "value1",
 "Type": "string"
 },
 "param2": {
 "Value": 1,
 "Type": "integer"
 }
 },
 // optional
 "PromptParams": {
 "PromptTemplate": "some template",
 "UserPromptEditingEnabled": true,
 "MaxPromptTemplateLength": 1000,
 "MaxInputTextLength": 1000,
 "DisambiguationPromptTemplate": "some disambiguation template",
 "DisambiguationEnabled": true
 },
 "Temperature": 1.0, // optional
 "Streaming": true, // optional
 "RAGEnabled": true, // optional. Must be true if providing KnowledgeBaseParams above.
 "Verbose": false // optional
```

```
 },
 "AgentParams": {
 "AgentType": "Bedrock",
 "BedrockAgentParams": {
 "AgentId": "agent-id",
 "AgentAliasId": "alias-id",
 "EnableTrace": true
 }
 },
 // optional
 "AuthenticationParams": {
 "AuthenticationProvider": "Cognito",
 "CognitoParams": {
 "ExistingUserPoolId": "user-pool-id",
 "ExistingUserPoolClientId": "client-id" // optional. If not provided, the solution
 will create a client for you in the provided pool
 }
 }

}
```

Untuk pembaruan, strukturnya sama seperti di atas dengan beberapa peringatan:

- Nama use case tidak dapat diubah
- Kasus penggunaan hanya dapat mengubah grup keamanan dan subnet setelah digunakan di VPC. VPC itu sendiri tidak dapat diubah.
- Jika indeks Kendra dibuat untuk Anda sebagai basis pengetahuan, Anda tidak dapat mengubah konfigurasi indeks tersebut (misalnya,, KendraIndexName) QueryCapacityUnits

Kasus Penggunaan Bersama APIs

Titik akhir REST API berikut tersedia untuk kasus penggunaan Teks dan Agen Batuan Dasar:

API REST	Metode HTTP	Fungsionalitas	Penelepon resmi
`/details/{useCaseConfigKey}`	`GET`	Mendapat detail konfigurasi untuk kasus penggunaan tertentu.	Token JWT yang diautentikasi Amazon Cognito

WebSocket API	Fungsionalitas	Penelepon resmi
`/$connect`	Memulai WebSocket koneksi dan mengautentikasi pengguna.	Token JWT yang diautentikasi Amazon Cognito
`/$disconnect`	Endpoint dipanggil ketika WebSocket koneksi telah terputus.	Token JWT yang diautentikasi Amazon Cognito

Gunakan API Detail Kasus

Titik akhir API detail mengambil informasi tentang kasus penggunaan tertentu:

```
GET /details/{useCaseConfigKey}
```

Titik akhir ini mengembalikan detail konfigurasi untuk kasus penggunaan tertentu, termasuk parameter model, setelan basis pengetahuan, dan informasi penerapan lainnya. Ini membutuhkan token JWT yang diautentikasi Amazon Cognito untuk otorisasi.

Kasus penggunaan teks

WebSocket API	Fungsionalitas	Penelepon resmi
`/sendMessage`	Mengirim pesan obrolan pengguna ke WebSocket untuk diproses dengan pengalaman LLM yang dikonfigurasi.	Token JWT yang diautentikasi Amazon Cognito

API REST	Metode HTTP	Fungsionalitas	Penelepon resmi
`/feedback/{useCaseId}`	`POST`	Mengirimkan umpan balik pengguna untuk	Token JWT yang diautentikasi Amazon Cognito

API REST	Metode HTTP	Fungsionalitas	Penelepon resmi
		kasus penggunaan tertentu.	

Muatan SendMessage

Jika Anda langsung berintegrasi dengan `/sendMessage` API, Anda harus mematuhi format payload permintaan dan respons berikut.

Minta Muatan

```
{
 "action": "sendMessage",
 "question": "the message to send to the api",
 "conversationId": "", // If not provided, a new conversation will be created, with the
 conversationId returned in the response. All subsequent messages in that conversation
 (where history is retained), should provide the conversationId there.
 "promptTemplate": "", // Optional. Overrides the configured prompt
 "authToken": "XXXX" // Optional. accessToken from cognito flow. Required for RAG with
 RBAC
}
```

Nama Parameter	Tipe	Deskripsi
aksi	`String`	Saat ini kami hanya mendukung tindakan “SendMessage” di WebSocket
pertanyaan	`String`	Masukan pengguna untuk mengirim ke LLM
ConversationID	`String`	UUID yang mengidentifikasi percakapan. Jika tidak disediakan, percakapan baru akan dibuat, dengan ConversationId dikembalikan dalam respons. Semua pesan berikutnya dalam percakapa

Nama Parameter	Tipe	Deskripsi
		n itu (di mana Anda history/c ontext ingin disimpan), harus menyediakan ConversationId di sana.
PromptTemplate	`String`[Opsional]	Mengganti template prompt untuk pesan ini. Jika kosong atau tidak disediakan, akan default ke prompt yang disetel pada waktu penerapan. Harus memiliki placeholder yang tepat yang ditentuka n untuk konfigurasi yang diberikan (yaitu {history} dan {input} untuk penerapan AI Sagemaker non-RAG, dengan penambahan {context} jika menggunakan RAG untuk semua penerapan.

Nama Parameter	Tipe	Deskripsi
AuthToken	`String`[Opsional]	AccessToken seperti yang diperoleh dari aliran autentikasi cognito. Ini diperlukan saat menjalankan endpoint websocket obrolan yang dikonfigurasi untuk RAG dengan Role Based Access Control (RBAC). Daftar klaim cognito:groups dalam token JWT ini digunakan untuk mengontrol akses ke dokumen dalam indeks Kendra. Parameter ini tidak diperlukan untuk kasus penggunaan non-RAG. Hal ini juga tidak diperlukan untuk kasus penggunaan RAG yang memiliki RBAC dinonaktifkan.

Muatan Respon

Respon Pertanyaan

WebSocket API akan merespons dengan 1 (jika streaming dinonaktifkan) atau banyak (jika streaming diaktifkan) objek JSON terstruktur sebagai berikut untuk setiap kueri.

```
{
 "data": "some data",
 "conversationId": "id",
}
```

Nama Parameter	Tipe	Deskripsi
data	`String`	Sepotong respons dari LLM jika streaming diaktifkan,

Nama Parameter	Tipe	Deskripsi
		atau seluruh respons. Jika menggunakan streaming, respons format ini dengan konten data END_CONVERSATION akan dikirim untuk menunjukkan akhir respons terhadap satu pertanyaan.
ConversationID	`String`	ID percakapan yang dimiliki oleh respons SourceDocument ini.

Tanggapan Dokumen Sumber

Jika Anda telah mengonfigurasi kasus penggunaan RAG untuk mengembalikan dokumen sumber, Anda juga akan menerima muatan berikut di akhir setiap respons untuk setiap dokumen sumber yang digunakan untuk membuat respons.

```
{
 "sourceDocument": {
 "excerpt": "some excerpt from the",
 "location": "s3://fake-bucket/test.txt",
 "score": 0.500,
 "document_title": null,
 "document_id": null,
 "additional_attributes": null
 },
 "conversationId": "some-id"
}
```

Nama Parameter	Tipe	Deskripsi
kutipan	`String`	Kutipan dari dokumen sumber.
lokasi	`String`	Lokasi dokumen sumber. Ini akan tergantung pada sumber data yang digunakan dan jenis

Nama Parameter	Tipe	Deskripsi
		basis pengetahuan, tetapi bisa berupa hal-hal seperti s3 URIs atau situs web.
skor	`Number` | `String`	Keyakinan bahwa dokumen tersebut sesuai dengan pertanyaan yang diajukan. Ini akan menjadi float dari 0 ke 1 untuk Bedrock, dan string (misalnya TINGGI, RENDAH, dll.) untuk Kendra.
document_title	`String`	Judul dokumen sumber yang dikembalikan. Hanya tersedia saat menggunakan Kendra.
document_id	`String`	ID dari dokumen sumber yang dikembalikan. Hanya tersedia saat menggunakan Kendra.
additional_attributes	`String`	Bidang ini akan berisi semua atribut tambahan pada dokumen yang disesuaikan pada basis pengetahuan Anda saat konsumsi.
ConversationID	`String`	ID percakapan yang dimiliki oleh respons SourceDoc ument ini.

Umpan Balik API Payload

Di bawah ini adalah contoh payload POST ke `/feedback/{useCaseId}` titik akhir, yang akan mengirimkan umpan balik pengguna untuk kasus penggunaan tertentu:

```
{
```

```
  "useCaseRecordKey": "12345678-12345678",
  "conversationId": "12345678-1234-1234-1234-123456789012",
  "messageId": "12345678-1234-1234-1234-123456789012",
  "feedback": "positive",
  "feedbackReason": ["accurate", "helpful"],
  "comment": "This response was very helpful.",
  "rephrasedQuery": "What are the key features of Amazon Bedrock?",
  "sourceDocuments": [
    "s3://bucket-name/document1.pdf",
    "s3://bucket-name/document2.pdf"
  ]
}
```

Kasus penggunaan Agen Batuan Dasar

WebSocket API	Fungsionalitas	Penelepon resmi
`/invokeAgent`	Mengirim pesan pengguna ke WebSocket untuk diproses dengan agen yang dikonfigu rasi.	Token JWT yang diautentikasi Amazon Cognito

Muatan InvokeAgent

Jika Anda langsung berintegrasi dengan`/invokeAgent API`, Anda harus mematuhi format payload permintaan dan respons berikut.

Meminta muatan

```
{
"action": "invokeAgent",
"inputText": "User query to the agent",
"conversationId": "", // Optional. Empty conversationId implies a new conversation.
 When not provided, a new conversationId will be created and returned with the
 response. All subsequent messages in the same conversation should provide the same
 conversationId (i.e. chat memory/history is maintained).
"authToken": "XXXX" // Optional. accessToken from cognito flow. If provided, it needs
 to be a valid JWT token associated with the user
}
```

Nama parameter	Tipe	Deskripsi
aksi	`String`	Kami hanya mendukung `invokeAgent` aksi di WebSocket.
InputTeks	`String`	Masukan pengguna untuk dikirim ke LLM.
ConversationID	`String[Optional]`	UUID yang secara unik mengidentifikasi percakapan. Jika Anda tidak memberikan nilai ini, solusi akan membuat percakapan baru dan menampilkan ConversationId dalam respons. Semua pesan berikutnya dalam percakapan itu (di mana Anda ingin menyimpan riwayat dan konteks) menyediakan ConversationId di sana.
AuthToken	`String[Optional]`	AccessToken seperti yang diperoleh dari aliran autentikasi Amazon Cognito. Parameter ini tidak diperlukan. Jika Anda memberikannya, token JWT akan divalidasi. Ini membantu mempermudah solusi ini untuk diperpanjang.

Muatan respons

Tanggapan pertanyaan

WebSocket API akan merespons dengan satu (jika streaming dinonaktifkan) atau banyak (jika streaming diaktifkan) objek JSON terstruktur sebagai berikut untuk setiap kueri.

```
{
 "data" "some data",
 "conversationId": "id",
 }
```

Nama parameter	Tipe	Deskripsi
data	`String`	Tanggapan dari doa agen.
ConversationID	`String`	ID percakapan.

Referensi

Bagian ini mencakup informasi tentang pengumpulan data untuk solusi ini, petunjuk ke sumber daya terkait, dan daftar pembangun yang berkontribusi pada solusi ini.

Penyedia LLM yang didukung

Solusinya dapat diintegrasikan dengan penyedia LLM berikut:

1. Amazon Bedrock
   - Dokumentasi: https://aws.amazon.com/bedrock/
   - Model yang didukung:
     - Amazon
       - Nova Lite
       - Nova Mikro
       - Nova Pro
     - AI21 Lab
       - Jamba 1.5 Mini
       - Jamba 1.5 Besar
     - Antropik
       - Claude v3 Haiku
       - Claude v3.5 Soneta
       - Claude v3.7 Soneta (melalui penggunaan profil inferensi)
     - Cohere
       - Perintah R
       - Perintah R +
     - Deepseek
       - Deepseek-R1 (melalui penggunaan profil inferensi)
     - Meta
       - Llama 3
       - Llama 3.2 (melalui penggunaan profil inferensi)
     - Mistral AI

        - Instruksi Mistral 7B
        - Instruksi Mistral 8x7B
    - Inferensi lintas wilayah
        - Kemampuan untuk menggunakan profil inferensi yang ditentukan di Wilayah yang sama dengan dasbor Deployment
2. Amazon SageMaker AI
    - Dokumentasi: https://aws.amazon.com/sagemaker/
    - Model yang didukung: Model Teks ke Teks

Untuk parameter model terbaru, praktik terbaik, dan penggunaan yang direkomendasikan, lihat dokumentasi dari penyedia model.

Pengumpulan data

Solusi ini mengirimkan metrik operasional ke AWS (“Data”) tentang penggunaan solusi ini. Kami menggunakan Data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini serta layanan serta produk terkait. Pengumpulan AWS atas Data ini tunduk pada Pemberitahuan Privasi AWS.

Kontributor

- Tarek Abdunabi
- Majd Arbash
- George Bearden
- Mukit Bin Momin
- Michael Connor
- Johny Duval
- Nihit Kasabwala
- Ahern Knox
- Simon Krol
- Michael Lin
- Tim Mekari

- Ibrahim Muhammad
- Omar Radwan Mohsen
- James Nixon
- Dekshitha Ravikumar
- Jae Shim
- Ajay Swamy
- Mohammed Taha
- Reet Takkar
- Dimitri Tchikatilov
- Karangan Bunga Jason
- Kamyar Ziabari

Revisi

Tanggal publikasi: Oktober 2023 (pembaruan terakhir: Januari 2025)

Periksa file ChangelOG.md di GitHub repositori untuk melihat semua perubahan dan pembaruan penting pada perangkat lunak. Changelog memberikan catatan perbaikan dan perbaikan yang jelas untuk setiap versi.

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, pernyataan, atau ketentuan dalam bentuk apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

Generative AI Application Builder on AWS dilisensikan berdasarkan ketentuan Lisensi Apache Versi 2.0.

Important

Generative AI Application Builder di AWS memungkinkan Anda untuk membangun dan menerapkan aplikasi kecerdasan buatan generatif di AWS dengan melibatkan model AI generatif pilihan Anda, termasuk model AI generatif pihak ketiga yang dapat Anda pilih untuk digunakan yang tidak dimiliki AWS atau memiliki kendali atas (“Model AI Generatif Pihak Ketiga”).

Penggunaan Anda atas Model AI Generatif Pihak Ketiga diatur oleh persyaratan yang diberikan kepada Anda oleh penyedia Model AI Generatif Pihak Ketiga saat Anda memperoleh lisensi untuk menggunakannya (misalnya, ketentuan layanan, perjanjian lisensi, kebijakan penggunaan yang dapat diterima, dan kebijakan privasi).

Anda bertanggung jawab untuk memastikan bahwa penggunaan Anda atas Model AI Generatif Pihak Ketiga mematuhi persyaratan yang mengaturnya, dan hukum, aturan, peraturan, kebijakan, atau standar apa pun yang berlaku untuk Anda.

Anda juga bertanggung jawab untuk membuat penilaian independen Anda sendiri terhadap Model AI Generatif Pihak Ketiga yang Anda gunakan, termasuk outputnya dan bagaimana penyedia Model AI Generatif Pihak Ketiga menggunakan data apa pun yang mungkin dikirimkan kepada mereka berdasarkan penerapan Anda. AWS tidak membuat pernyataan, jaminan, atau jaminan apa pun terkait Model AI Generatif Pihak Ketiga, yang merupakan “Konten Pihak Ketiga” berdasarkan perjanjian Anda dengan AWS. Generative AI Application Builder on AWS ditawarkan kepada Anda sebagai “Konten AWS” berdasarkan perjanjian Anda dengan AWS.

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