Membuat solusi Retrieval Augmented Generation untuk perawatan kesehatan AWS

# AWS Bimbingan Preskriptif

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Bimbingan Preskriptif: Membuat solusi Retrieval Augmented Generation untuk perawatan kesehatan AWS

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

# Membuat solusi Retrieval Augmented Generation untuk perawatan kesehatan AWS

Amazon Web Services, Accenture, dan Cadiem (kontributor)

Maret 2025 (sejarah dokumen)

Sebelum model bahasa besar (LLMs) dan AI generatif, tugas mengembangkan aplikasi otomatis dan presisi tinggi di industri perawatan kesehatan sangat menantang. Metode tradisional sangat bergantung pada entri dan analisis data manual. Kompleksitas menganalisis pencitraan medis dan catatan pasien memerlukan intervensi manusia yang ekstensif, yang sering mengakibatkan alur kerja yang terfragmentasi dan tidak efisien. Kemajuan teknologi AI membantu Anda membangun aplikasi yang sangat personal dalam skala besar. Aplikasi perawatan kesehatan sekarang dapat berintegrasi dengan basis pengetahuan medis, menafsirkan gambar diagnostik dengan akurasi yang meningkat, dan memperkirakan hasil pasien dengan menggunakan model prediktif.

Panduan ini mengeksplorasi bagaimana LLMs merevolusi perawatan kesehatan melalui aplikasi Retrieval Augmented Generation yang dapat Anda bangun. Layanan AWSRetrieval Augmented Generation (RAG) adalah teknologi AI generatif di mana LLM mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Aplikasi RAG membumikan keluaran model dalam pengetahuan dunia nyata, yang mengurangi halusinasi dan meningkatkan relevansi respons. Di sektor kesehatan, RAG dapat digunakan untuk memberikan informasi yang akurat dan up-to-date medis, memastikan bahwa penyedia layanan kesehatan memiliki akses ke penelitian terbaru dan pedoman klinis. Dengan mengubah data menjadi wawasan yang dapat ditindaklanjuti dan mengotomatiskan proses yang kompleks, teknologi ini membantu meningkatkan perawatan pasien, merampingkan operasi, dan meningkatkan produktivitas profesional perawatan kesehatan.

Di Amazon Bedrock, Anda dapat menyempurnakan LLMs dan mengintegrasikannya dengan agen cerdas untuk menciptakan solusi perawatan kesehatan tingkat lanjut. Menyoroti sinergi antara Amazon  OpenSearch Service dan Amazon Neptunus, panduan ini menunjukkan bagaimana layanan ini meningkatkan solusi RAG melalui peningkatan relevansi pencarian dan pengambilan data multi-sumber tingkat lanjut. Anda dapat mengatur solusi Amazon Bedrock komprehensif yang menggunakan agen Amazon Bedrock dan LangChainuntuk mengoordinasikan interaksi dengan mulus di berbagai repositori data. Integrasi ini menunjukkan kekuatan menggabungkan layanan khusus untuk menciptakan sistem berbasis AI yang lebih efektif dan efisien.

# Perawatan dan produktivitas pasien

Panduan ini menyajikan dua kasus penggunaan dunia nyata untuk perawatan dan produktivitas pasien: peningkatan data pasien dan memprediksi risiko masuk kembali. Ini memberikan cetak biru strategis untuk menerapkan solusi ini dalam skala besar, menawarkan organisasi perawatan kesehatan jalur yang jelas untuk mengindustrialisasi proses yang digerakkan oleh AI. Melalui wawasan ini, institusi kesehatan dapat menggunakan teknologi AI canggih untuk menciptakan alur kerja yang lebih efisien dan cerdas.

# Manajemen bakat

Panduan ini juga menguraikan strategi untuk melatih kembali dan memberdayakan petugas kesehatan untuk mengintegrasikan AI generatif secara mulus ke dalam rutinitas sehari-hari mereka. Hal ini dapat meningkatkan produktivitas dan kualitas perawatan pasien. Dengan melengkapi tenaga kerja mereka dengan keterampilan untuk secara efektif menggunakan alat AI canggih, organisasi perawatan kesehatan dapat memaksimalkan laba atas investasi mereka dan mendorong inovasi dalam perawatan pasien.

Solusi manajemen bakat bertenaga AI ini mencakup fitur-fitur utama berikut:

- Pengurai resume bakat cerdas — Dengan menggunakan lanjutan yang LLMs tersedia di Amazon Bedrock, alat ini secara efisien mengekstrak dan menganalisis keterampilan dan atribut bakat penting dari resume. Alat ini dapat merampingkan proses rekrutmen.
- Basis pengetahuan bakat — Didukung oleh Amazon Neptunus, database dinamis ini memberikan wawasan real-time tentang tingkat kepegawaian, distribusi keterampilan, dan tren industri. Ini membantu Anda membuat keputusan berbasis data tentang manajemen tenaga kerja.
- Mesin rekomendasi pembelajaran - Alat yang digerakkan oleh AI ini mengidentifikasi kesenjangan keterampilan dalam organisasi dan merekomendasikan program pelatihan yang dipersonalisasi untuk staf medis. Alat ini mempromosikan pengembangan profesional yang berkelanjutan dan membantu tenaga kerja Anda beradaptasi dengan teknologi perawatan kesehatan yang berkembang.

Bersama-sama, fitur berbasis AI ini membantu mengoptimalkan kinerja tenaga kerja, merevolusi manajemen bakat dengan peningkatan kecerdasan dan efisiensi.

# Peluang dan tantangan

Amazon Bedrock dapat memberikan peningkatan produktivitas, skalabilitas, efektivitas biaya, dan wawasan berbasis data. Amazon Bedrock memberdayakan organisasi perawatan kesehatan untuk menggunakan LLMs secara efektif di berbagai kasus penggunaan, mulai dari pembuatan konten dan analisis data hingga pengambilan keputusan otomatis. Panduan ini memberikan pendekatan untuk mengatasi tantangan AI generatif umum, seperti masalah kualitas data, skalabilitas infrastruktur, pemeliharaan kinerja model, dan persyaratan peningkatan berkelanjutan selama transisi dari bukti konsep ke produksi.

## Peluang untuk aplikasi AI generatif dalam perawatan kesehatan

Industri perawatan kesehatan siap untuk perubahan transformatif, didorong oleh peluang yang disajikan oleh aplikasi AI generatif. AI generatif memiliki potensi untuk meningkatkan perawatan pasien, merampingkan operasi, dan mempercepat penelitian medis. Dengan menggunakan model AI canggih, penyedia layanan kesehatan dapat mengotomatiskan augmentasi rekam medis. Riwayat komprehensif dan up-to-date pasien memfasilitasi diagnosis dan rencana perawatan yang lebih akurat. Analisis gambar berbasis AI, seperti menafsirkan sonogram dan pencitraan medis lainnya, dapat memberikan wawasan yang cepat dan tepat, mengurangi beban kerja pada profesional medis dan meminimalkan risiko kesalahan manusia.

Di luar diagnostik dan pengobatan, AI generatif dapat memainkan peran penting dalam analitik prediktif. Analisis prediktif membantu organisasi perawatan kesehatan mengantisipasi hasil pasien dan mempersonalisasi rencana perawatan yang sesuai. Teknologi ini juga dapat mengoptimalkan proses administrasi, mulai dari mengelola data pasien hingga merampingkan komunikasi antara penyedia dan pasien. Dengan mengintegrasikan solusi AI generatif dengan sistem perawatan kesehatan yang ada, institusi medis dapat mencapai efisiensi yang lebih besar, mengurangi biaya, dan pada akhirnya memberikan perawatan berkualitas lebih tinggi. Integrasi AI dengan perawatan kesehatan bukan hanya peningkatan tetapi perubahan mendasar menuju perawatan yang lebih cerdas, responsif, dan berpusat pada pasien.

## Analisis gambar tingkat lanjut

Menggabungkan Amazon Bedrock dengan penyimpanan data, seperti Amazon Neptunus OpenSearch dan Amazon Service, dapat membantu Anda mengatasi kompleksitas analisis gambar tingkat lanjut dalam perawatan kesehatan. Solusi pengambilan informasi dapat meningkatkan proses

penemuan penyakit dan meningkatkan akurasi interpretasi dengan menilai gambar diagnostik dan menafsirkan sonogram. Solusinya dapat mengintegrasikan data penilaian visual dan tekstual dengan tinjauan penilaian pasien manual oleh dokter.

# Tantangan dengan industrialisasi solusi

Hambatan utama yang harus diatasi ketika mengindustrialisasi solusi AI dalam perawatan kesehatan adalah kualitas dan ketersediaan data. Data kesehatan sering ada dalam format yang terfragmentasi dan tidak konsisten. Memastikan bahwa model AI memiliki akses ke data yang bersih, terstruktur, dan representatif sangat penting untuk mempertahankan kinerja dalam skenario dunia nyata. Skalabilitas infrastruktur dapat menjadi tantangan karena lingkungan produksi. Lingkungan ini perlu menangani volume besar data pasien real-time sambil memberikan waktu respons yang cepat dan menjaga kepatuhan terhadap peraturan privasi data, seperti Health Insurance Portability and Accountability Act (HIPAA). Selain itu, dengan informasi medis yang muncul dan data pasien yang berkembang dari waktu ke waktu, model AI perlu dilatih ulang dan diperbarui agar tetap relevan dan memberikan rekomendasi yang akurat. Akhirnya, mengintegrasikan solusi AI ini ke dalam sistem perawatan kesehatan yang ada dapat menjadi kompleks karena masalah interoperabilitas dan kebutuhan untuk penyelarasan dengan alur kerja klinis saat ini. Integrasi ini membutuhkan perubahan teknis dan operasional.

# Kasus penggunaan: Membangun aplikasi intelijen medis dengan data pasien yang ditambah

AI generatif dapat membantu meningkatkan perawatan pasien dan produktivitas staf dengan meningkatkan fungsi klinis dan administrasi. Analisis gambar berbasis AI, seperti menafsirkan sonogram, mempercepat proses diagnostik dan meningkatkan akurasi. Ini dapat memberikan wawasan kritis yang mendukung intervensi medis tepat waktu.

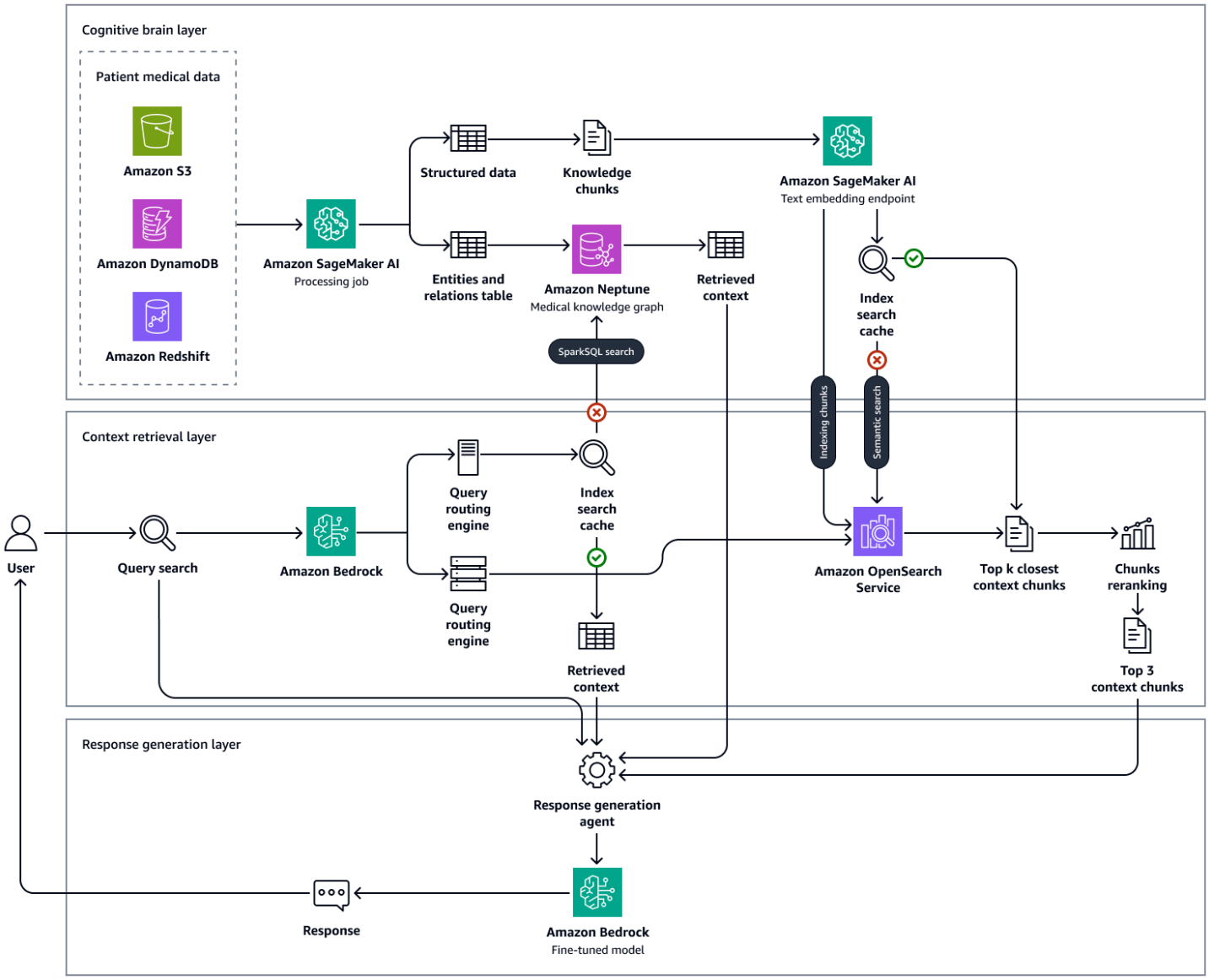
Saat Anda menggabungkan model AI generatif dengan grafik pengetahuan, Anda dapat mengotomatiskan organisasi kronologis catatan pasien elektronik. Ini membantu Anda mengintegrasikan data real-time dari interaksi dokter-pasien, gejala, diagnosis, hasil lab, dan analisis gambar. Ini melengkapi dokter dengan data pasien yang komprehensif. Data ini membantu dokter membuat keputusan medis yang lebih akurat dan tepat waktu, meningkatkan hasil pasien dan produktivitas penyedia layanan kesehatan.

## Ikhtisar solusi

AI dapat memberdayakan dokter dan dokter dengan mensintesis data pasien dan pengetahuan medis untuk memberikan wawasan yang berharga. Solusi Retrieval Augmented Generation (RAG) ini adalah mesin intelijen medis yang mengkonsumsi serangkaian data dan pengetahuan pasien yang komprehensif dari jutaan interaksi klinis. Ini memanfaatkan kekuatan AI generatif untuk menciptakan wawasan berbasis bukti untuk perawatan pasien yang lebih baik. Ini dirancang untuk meningkatkan alur kerja klinis, mengurangi kesalahan, dan meningkatkan hasil pasien.

Solusinya mencakup kemampuan pemrosesan gambar otomatis yang didukung oleh. LLMs Kemampuan ini mengurangi jumlah waktu yang harus dihabiskan tenaga medis secara manual untuk mencari gambar diagnostik serupa dan menganalisis hasil diagnostik.

Gambar berikut menunjukkan end-to-end-workflow solusi ini. Ini menggunakan Amazon Neptunus, SageMaker Amazon AI, OpenSearch Amazon Service, dan model fondasi di Amazon Bedrock. Untuk agen pengambilan konteks yang berinteraksi dengan grafik pengetahuan medis di Neptunus, Anda dapat memilih antara agen Amazon Bedrock dan LangChain agen.




Dalam percobaan kami dengan contoh pertanyaan medis, kami mengamati bahwa tanggapan akhir yang dihasilkan oleh pendekatan kami menggunakan grafik pengetahuan yang dipertahankan di Neptunus OpenSearch, basis data vektor yang menampung basis pengetahuan klinis, dan Amazon LLMs Bedrock didasarkan pada faktualitas dan jauh lebih akurat dengan mengurangi positif palsu dan meningkatkan positif sejati. Solusi ini dapat menghasilkan wawasan berbasis bukti tentang status kesehatan pasien dan bertujuan untuk meningkatkan alur kerja klinis, mengurangi kesalahan, dan meningkatkan hasil pasien.

Membangun solusi ini terdiri dari langkah-langkah berikut:

- Langkah 1: Menemukan data

- Langkah 2: Membangun grafik pengetahuan medis
- Langkah 3: Membangun agen pengambilan konteks untuk menanyakan grafik pengetahuan medis
- Langkah 4: Membuat basis pengetahuan data deskriptif real-time
- Langkah 5: Gunakan LLMs untuk menjawab pertanyaan medis

# Langkah 1: Menemukan data

Ada banyak kumpulan data medis open source yang dapat Anda gunakan untuk mendukung pengembangan solusi berbasis AI perawatan kesehatan. Salah satu dataset tersebut adalah dataset MIMIC-IV, yang merupakan dataset catatan kesehatan elektronik (EHR) yang tersedia untuk umum yang banyak digunakan dalam komunitas penelitian kesehatan. MIMIC-IV berisi informasi klinis terperinci, termasuk catatan pelepasan teks gratis dari catatan pasien. Anda dapat menggunakan catatan ini untuk bereksperimen dengan penjumlahan teks dan teknik ekstraksi entitas. Teknik-teknik ini membantu Anda mengekstrak informasi medis (seperti gejala pasien, obat yang diberikan, dan perawatan yang diresepkan) dari teks yang tidak terstruktur.

Anda juga dapat menggunakan kumpulan data yang menyediakan ringkasan pelepasan pasien beranotasi dan tidak teridentifikasi yang secara khusus dikuratori untuk tujuan penelitian. Kumpulan data ringkasan pelepasan dapat membantu Anda bereksperimen dengan ekstraksi entitas, memungkinkan Anda mengidentifikasi entitas medis utama (seperti kondisi, prosedur, dan obat-obatan) dari teks. Langkah 2: Membangun grafik pengetahuan medisdalam panduan ini menjelaskan bagaimana Anda dapat menggunakan data terstruktur yang diekstraksi dari MIMIC-IV dan kumpulan data ringkasan pelepasan untuk membuat grafik pengetahuan medis. Grafik pengetahuan medis ini berfungsi sebagai tulang punggung untuk kueri tingkat lanjut dan sistem pendukung keputusan untuk profesional perawatan kesehatan.

Selain kumpulan data berbasis teks, Anda dapat menggunakan kumpulan data gambar. Misalnya, kumpulan data Musculoskeletal Radiographs (MURA), yang merupakan database komprehensif gambar radiografi multi-tampilan tulang. Gunakan kumpulan data gambar tersebut untuk bereksperimen dengan penilaian diagnostik melalui teknik decoding gambar medis. Teknik decoding ini sangat penting untuk diagnosis dini penyakit, seperti penyakit muskuloskeletal, penyakit kardiovaskular, dan osteoporosis. Dengan menyempurnakan model fondasi visi dan bahasa pada kumpulan data gambar medis, Anda dapat mendeteksi kelainan pada gambar diagnostik. Ini membantu sistem memberikan wawasan diagnostik awal dan akurat kepada dokter. Dengan menggunakan kumpulan data gambar dan teks, Anda dapat membuat aplikasi perawatan kesehatan berbasis AI yang mampu memproses data teks dan gambar untuk meningkatkan perawatan pasien.

# Langkah 2: Membangun grafik pengetahuan medis

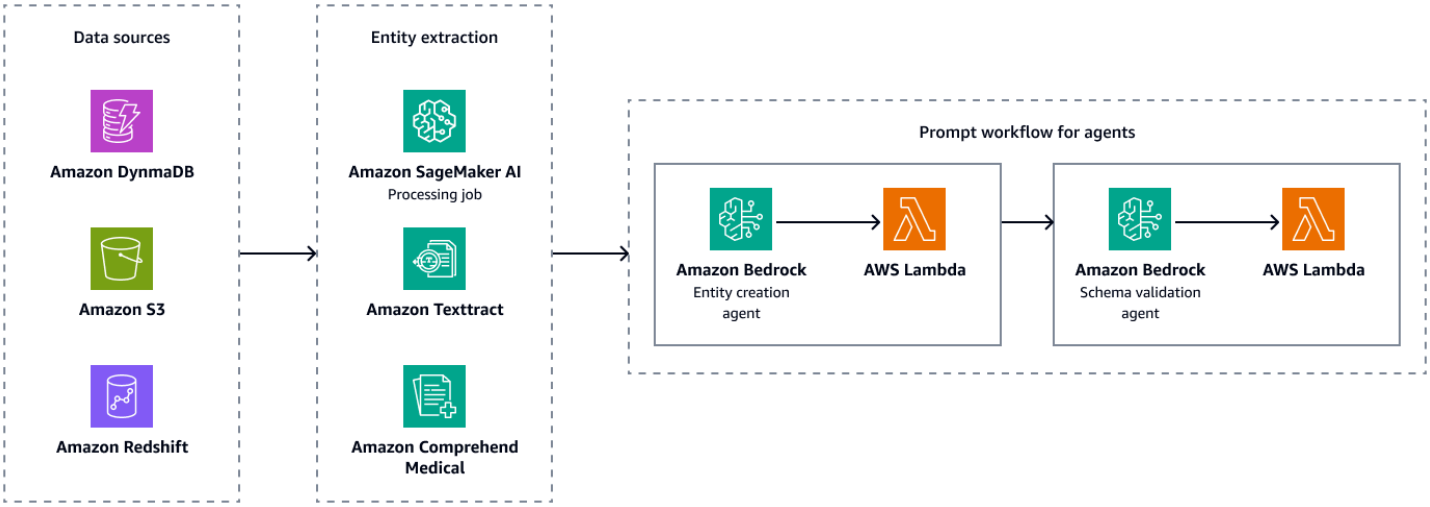
Untuk setiap organisasi perawatan kesehatan yang ingin membangun sistem pendukung keputusan berdasarkan basis pengetahuan yang besar, tantangan utama adalah menemukan dan mengekstrak entitas medis yang ada dalam catatan klinis, jurnal medis, ringkasan pelepasan, dan sumber data lainnya. Anda juga perlu menangkap hubungan temporal, subjek, dan penilaian kepastian dari catatan medis ini untuk secara efektif menggunakan entitas, atribut, dan hubungan yang diekstraksi.

Langkah pertama adalah mengekstrak konsep medis dari teks medis yang tidak terstruktur dengan menggunakan prompt beberapa tembakan untuk model pondasi, seperti Llama 3 di Amazon Bedrock. Permintaan beberapa tembakan adalah ketika Anda memberikan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Menggunakan ekstraktor entitas medis berbasis LLM, Anda dapat mengurai teks medis yang tidak terstruktur dan kemudian menghasilkan representasi data terstruktur dari entitas pengetahuan medis. Anda juga dapat menyimpan atribut pasien untuk analisis hilir dan otomatisasi. Proses ekstraksi entitas mencakup tindakan berikut:

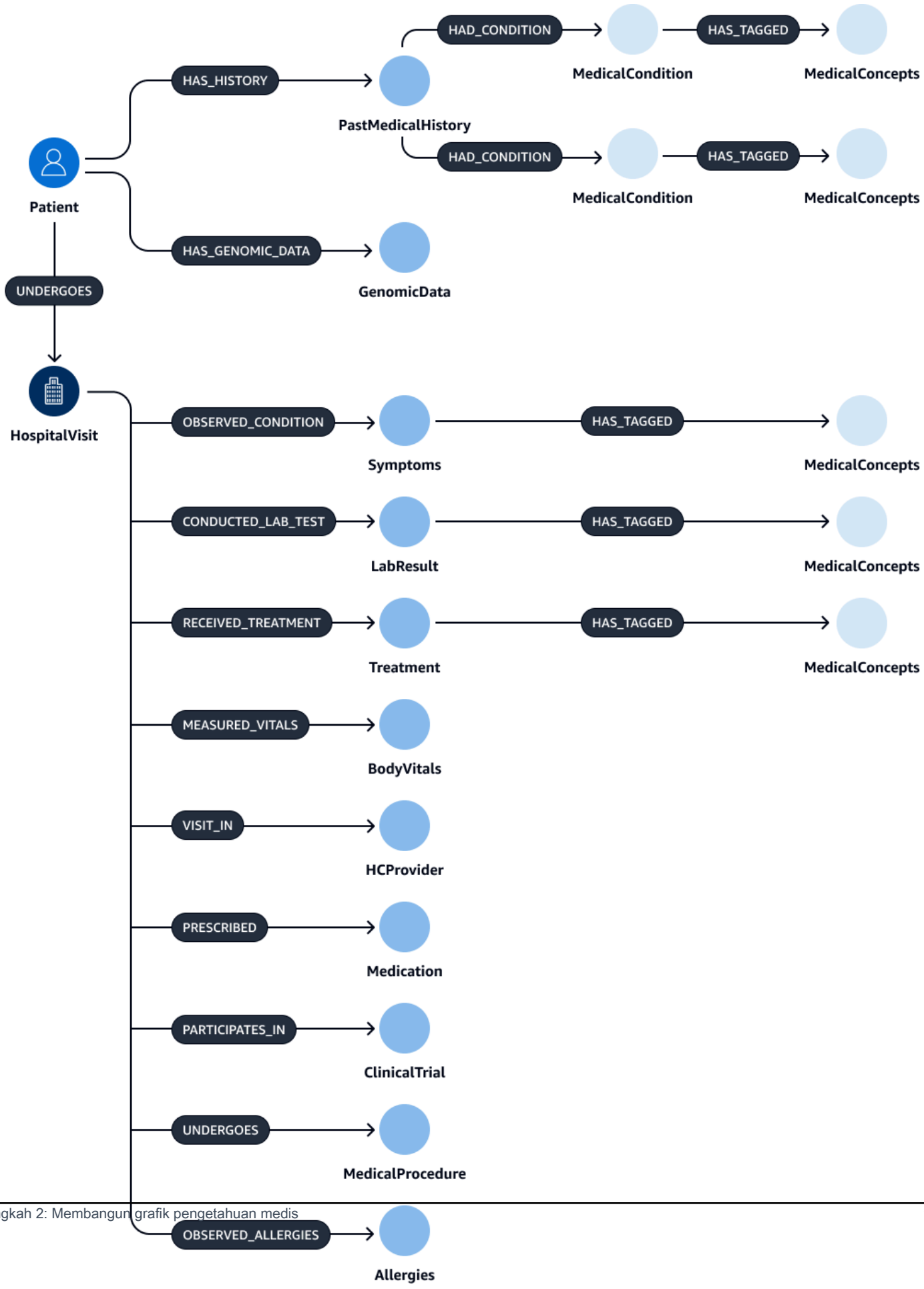
- Ekstrak informasi tentang konsep medis, seperti penyakit, obat-obatan, peralatan medis, dosis, frekuensi obat, durasi pengobatan, gejala, prosedur medis, dan atribut yang relevan secara klinis.
- Tangkap fitur fungsional, seperti hubungan temporal antara entitas yang diekstraksi, subjek, dan penilaian kepastian.
- Perluas kosakata medis standar, seperti berikut ini:
  - Pengidentifikasi konsep (RxCui) dari database RxNorm
  - Kode dari Klasifikasi Penyakit Internasional, Revisi ke-10, Modifikasi Klinis (ICD-10-CM)
  - Ketentuan dari Judul Subjek Medis (MeSH)
  - Konsep dari Nomenklatur Kedokteran Sistematisasi, Istilah Klinis (SNOMED CT)
  - Kode dari Unified Medical Language System (UMLS)
- Ringkas catatan pelepasan dan dapatkan wawasan medis dari transkrip.

Gambar berikut menunjukkan ekstraksi entitas dan langkah-langkah validasi skema untuk membuat kombinasi berpasangan yang valid dari entitas, atribut, dan hubungan. Anda dapat menyimpan data yang tidak terstruktur, seperti ringkasan debit atau catatan pasien, di Amazon Simple Storage Service (Amazon S3). Anda dapat menyimpan data terstruktur, seperti data perencanaan sumber daya perusahaan (ERP), catatan pasien elektronik, dan sistem informasi lab, di Amazon Redshift dan Amazon DynamoDB. Anda dapat membangun agen pembuatan entitas Amazon Bedrock. Agen

ini dapat mengintegrasikan layanan, seperti jalur ekstraksi data Amazon SageMaker AI, Amazon Textract, dan Amazon Comprehend Medical, untuk mengekstrak entitas, hubungan, dan atribut dari sumber data terstruktur dan tidak terstruktur. Terakhir, Anda menggunakan agen validasi skema Amazon Bedrock untuk memastikan bahwa entitas dan relasi yang diekstraksi sesuai dengan skema grafik yang telah ditentukan sebelumnya dan menjaga integritas koneksi tepi simpul dan properti terkait.




Setelah ekstraksi dan validasi entitas, relasi, dan atribut, Anda dapat menautkannya untuk membuat subject-object-predicate triplet. Anda menyerap data ini ke dalam database grafik Amazon Neptunus, seperti yang ditunjukkan pada gambar berikut. Database grafik dioptimalkan untuk menyimpan dan menanyakan hubungan antara item data.


HAD_CONDITION
HAS_TAGGED
MedicalCondition
MedicalConcepts
HAS_HISTORY
PastMedicalHistory
HAD_CONDITION
HAS_TAGGED
MedicalCondition
MedicalConcepts
Patient
HAS_GENOMIC_DATA
GenomicData
UNDERGOES
HospitalVisit
OBSERVED_CONDITION
HAS_TAGGED
Symptoms
MedicalConcepts
CONDUCTED_LAB_TEST
HAS_TAGGED
LabResult
MedicalConcepts
RECEIVED_TREATMENT
HAS_TAGGED
Treatment
MedicalConcepts
MEASURED_VITALS
BodyVitals
VISIT_IN
HCProvider
PRESCRIBED
Medication
PARTICIPATES_IN
ClinicalTrial
UNDERGOES
MedicalProcedure
OBSERVED_ALLERGIES
Allergies

Anda dapat membuat grafik pengetahuan yang komprehensif dengan data ini. Grafik pengetahuan membantu Anda mengatur dan menanyakan semua jenis informasi yang terhubung. Misalnya, Anda dapat membuat grafik pengetahuan yang memiliki node utama berikut:`HospitalVisit`,`PastMedicalHistory`,`Symptoms`, `MedicationMedicalProcedures`, dan`Treatment`.

Tabel berikut mencantumkan entitas dan atributnya yang mungkin Anda ekstrak dari catatan pelepasan.

| Entitas | Atribut |
| --- | --- |
| `Patient` | `PatientID` , `Name`, `Age`, `Gender`, `Address`, `ContactInformation` |
| `HospitalVisit` | `VisitDate` , `Reason`, `Notes` |
| `HealthcareProvider` | `ProviderID` , `Name`, `Specialty` , `ContactInformation` , `Address`, `AffiliatedInstitution` |
| `Symptoms` | `Description` , `RiskFactors` |
| `Allergies` | `AllergyType` , `Duration` |
| `Medication` | `MedicationID` , `Name`, `Description` , `Dosage`, `SideEffects` , `Manufacturer` |
| `PastMedicalHistory` | `ContinuingMedicines` |
| `MedicalCondition` | `ConditionName` , `Severity`, `Treatment Received` , `DoctorinCharge` , `HospitalName` , `MedicinesFollowed` |
| `BodyVitals` | `HeartRate` , `BloodPressure` , `RespiratoryRate` , `BodyTemperature` , `BMI` |
| `LabResult` | `LabResultID` , `PatientID` , `TestName`, `Result`, `Date` |

| Entitas | Atribut |
| --- | --- |
| `ClinicalTrial` | `TrialID`, `Name`, `Description` , `Phase`, `Status`, `StartDate` , `EndDate` |
| `GenomicData` | `GenomicDataID` , `PatientID` , `SequenceData` , `VariantInformation` |
| `Treatment` | `TreatmentID` , `Name`, `Description` , `Type`, `SideEffects` |
| `MedicalProcedure` | `ProcedureID` , `Name`, `Description` , `Risks`, `Outcomes` |
| `MedicalConcepts` | `UMLSCodes` , `MedicalVocabularies` |

Tabel berikut mencantumkan hubungan yang mungkin dimiliki entitas dan atribut yang sesuai. Misalnya, `Patient` entitas mungkin terhubung ke `HospitalVisit` entitas dengan `[UNDERGOES]` hubungan. Atribut untuk hubungan ini adalah`VisitDate`.

| Entitas subjek | Hubungan | Entitas objek | Atribut |
| --- | --- | --- | --- |
| `Patient` | `[UNDERGOES]` | `HospitalVisit` | `VisitDate` |
| `HospitalVisit` | `[VISIT_IN]` | `HealthcareProvider` | `ProviderName` , `Location`, `ProviderID` , `VisitDate` |
| `HospitalVisit` | `[OBSERVED_CONDITION]` | `Symptoms` | `Severity`, `CurrentStatus` , `VisitDate` |
| `HospitalVisit` | `[RECEIVED_TREATMENT]` | `Treatment` | `Duration`, `Dosage`, `VisitDate` |

| Entitas subjek | Hubungan | Entitas objek | Atribut |
|---|---|---|---|
| HospitalVisit | [PRESCRIBED] | Medication | Duration, Dosage, Adherence , VisitDate |
| Patient | [HAS_HISTORY] | PastMedicalHistory | Tidak ada |
| PastMedicalHistory | [HAD_CONDITION] | MedicalCondition | DiagnosisDate , CurrentStatus |
| HospitalVisit | [PARTICIPATES_IN] | ClinicalTrial | VisitDate , Status, Outcomes |
| Patient | [HAS_GENOMIC_DATA] | GenomicData | CollectionDate |
| HospitalVisit | [OBSERVED_ALLERGIES] | Allergies | VisitDate |
| HospitalVisit | [CONDUCTED_LAB_TEST] | LabResult | VisitDate , AnalysisDate , Interpretation |
| HospitalVisit | [UNDERGOES] | MedicalProcedure | VisitDate , Outcome |
| MedicalCondition | [HAS_TAGGED] | MedicalConcepts | Tidak ada |
| LabResult | [HAS_TAGGED] | MedicalConcepts | Tidak ada |
| Treatment | [HAS_TAGGED] | MedicalConcepts | Tidak ada |
| Symptoms | [HAS_TAGGED] | MedicalConcepts | Tidak ada |

# Langkah 3: Membangun agen pengambilan konteks untuk menanyakan grafik pengetahuan medis

Setelah Anda membangun database grafik medis, langkah selanjutnya adalah membangun agen untuk interaksi grafik. Agen mengambil konteks yang benar dan diperlukan untuk kueri yang dimasukkan oleh dokter atau dokter. Ada beberapa opsi untuk mengonfigurasi agen ini yang mengambil konteks dari grafik pengetahuan:

- Agen Amazon Bedrock
- LangChain pelaku

## Agen Amazon Bedrock untuk interaksi grafik

Agen Amazon Bedrock bekerja dengan mulus dengan database grafik Amazon Neptunus. Anda dapat melakukan interaksi lanjutan melalui grup tindakan Amazon Bedrock. Grup tindakan memulai proses dengan memanggil AWS Lambda fungsi, yang menjalankan kueri Neptunus OpenCypher.

Untuk menanyakan grafik pengetahuan, Anda dapat menggunakan dua pendekatan berbeda: eksekusi kueri langsung atau kueri dengan penyematan konteks. Pendekatan ini dapat diterapkan secara independen atau digabungkan, tergantung pada kasus penggunaan spesifik dan kriteria peringkat Anda. Dengan menggabungkan kedua pendekatan, Anda dapat memberikan konteks yang lebih komprehensif untuk LLM, yang dapat meningkatkan hasil. Berikut ini adalah dua pendekatan eksekusi query:

- Eksekusi kueri Direct Cypher tanpa embeddings — Fungsi Lambda mengeksekusi kueri langsung terhadap Neptunus tanpa pencarian berbasis penyematan. Berikut ini adalah contoh dari pendekatan ini:

```
MATCH (p:Patient)-[u:UNDERGOES]->(h:HospitalVisit) WHERE h.Reason = 'Acute Diabetes'
 AND date(u.VisitDate) > date('2024-01-01')
RETURN p.PatientID, p.Name, p.Age, p.Gender, p.Address, p.ContactInformation
```

- Eksekusi kueri Cypher langsung menggunakan pencarian penyematan - Fungsi Lambda menggunakan pencarian penyematan untuk meningkatkan hasil kueri. Pendekatan ini meningkatkan eksekusi query dengan menggabungkan embeddings, yang merupakan representasi vektor padat data. Embeddings sangat berguna ketika kueri membutuhkan kesamaan semantik atau pemahaman yang lebih luas di luar kecocokan yang tepat. Anda dapat

menggunakan model pra-terlatih atau terlatih khusus untuk menghasilkan embeddings untuk setiap kondisi medis. Berikut ini adalah contoh dari pendekatan ini:

```
CALL { WITH "Acute Diabetes" AS query_term RETURN search_embedding(query_term) AS
 similar_reasons }

MATCH (p:Patient)-[u:UNDERGOES]->(h:HospitalVisit) WHERE h.Reason IN similar reasons
 AND date(u.VisitDate) > date('2024-01-01')
RETURN p.PatientID, p.Name, p.Age, p.Gender, p.Address, p.ContactInformation
```

Dalam contoh ini, `search_embedding("Acute Diabetes")` fungsi mengambil kondisi yang secara semantik dekat dengan “Diabetes Akut.” Ini membantu pertanyaan untuk juga menemukan pasien yang memiliki kondisi seperti pra-diabetes atau sindrom metabolik.

Gambar berikut menunjukkan bagaimana agen Amazon Bedrock berinteraksi dengan Amazon Neptunus untuk melakukan kueri Cypher dari grafik pengetahuan medis.

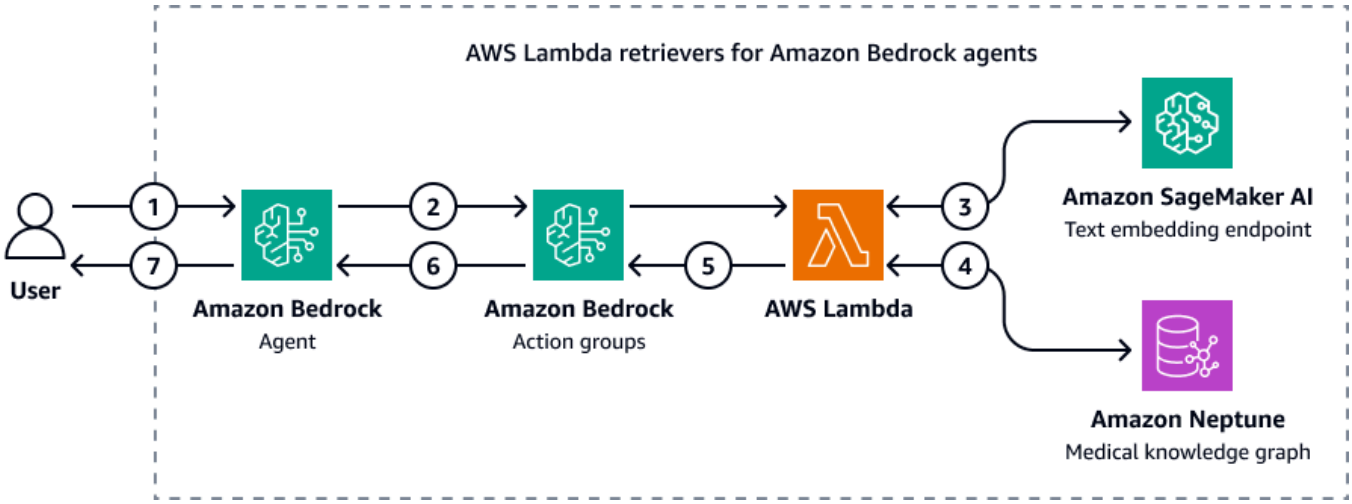



Diagram menunjukkan alur kerja berikut:

1. Pengguna mengirimkan pertanyaan ke agen Amazon Bedrock.
2. Agen Amazon Bedrock meneruskan variabel filter pertanyaan dan input ke grup tindakan Amazon Bedrock. Grup tindakan ini berisi AWS Lambda fungsi yang berinteraksi dengan titik akhir penyematan teks Amazon SageMaker AI dan grafik pengetahuan medis Amazon Neptunus.
3. Fungsi Lambda terintegrasi dengan titik akhir penyematan teks SageMaker AI untuk melakukan pencarian semantik dalam kueri OpenCypher. Ini mengubah kueri bahasa alami menjadi kueri OpenCypher dengan menggunakan dasar LangChain agen.

4. Fungsi Lambda menanyakan grafik pengetahuan medis Neptunus untuk dataset yang benar dan menerima output dari grafik pengetahuan medis Neptunus.
5. Fungsi Lambda mengembalikan hasil dari Neptunus ke grup aksi Amazon Bedrock.
6. Grup aksi Amazon Bedrock mengirim konteks yang diambil ke agen Amazon Bedrock.
7. Agen Amazon Bedrock menghasilkan respons dengan menggunakan kueri pengguna asli dan konteks yang diambil dari grafik pengetahuan.

## LangChain agen untuk interaksi grafik

Anda dapat mengintegrasikan LangChain dengan Neptunus untuk mengaktifkan kueri dan pengambilan berbasis grafik. Pendekatan ini dapat meningkatkan alur kerja berbasis AI dengan menggunakan kemampuan database grafik di Neptunus. Kebiasaan LangChain retriever bertindak sebagai perantara. Model dasar di Amazon Bedrock dapat berinteraksi dengan Neptunus dengan menggunakan kueri Cypher langsung dan algoritma grafik yang lebih kompleks.

Anda dapat menggunakan retriever kustom untuk menyempurnakan bagaimana LangChain agen berinteraksi dengan algoritma grafik Neptunus. Misalnya, Anda dapat menggunakan beberapa bidikan bidikan, yang membantu Anda menyesuaikan respons model fondasi berdasarkan pola atau contoh tertentu. Anda juga dapat menerapkan filter yang diidentifikasi LLM untuk menyempurnakan konteks dan meningkatkan ketepatan respons. Ini dapat meningkatkan efisiensi dan akurasi proses pengambilan keseluruhan saat berinteraksi dengan data grafik yang kompleks.

Gambar berikut menunjukkan bagaimana kustom LangChain agen mengatur interaksi antara model yayasan Amazon Bedrock dan grafik pengetahuan medis Amazon Neptunus.

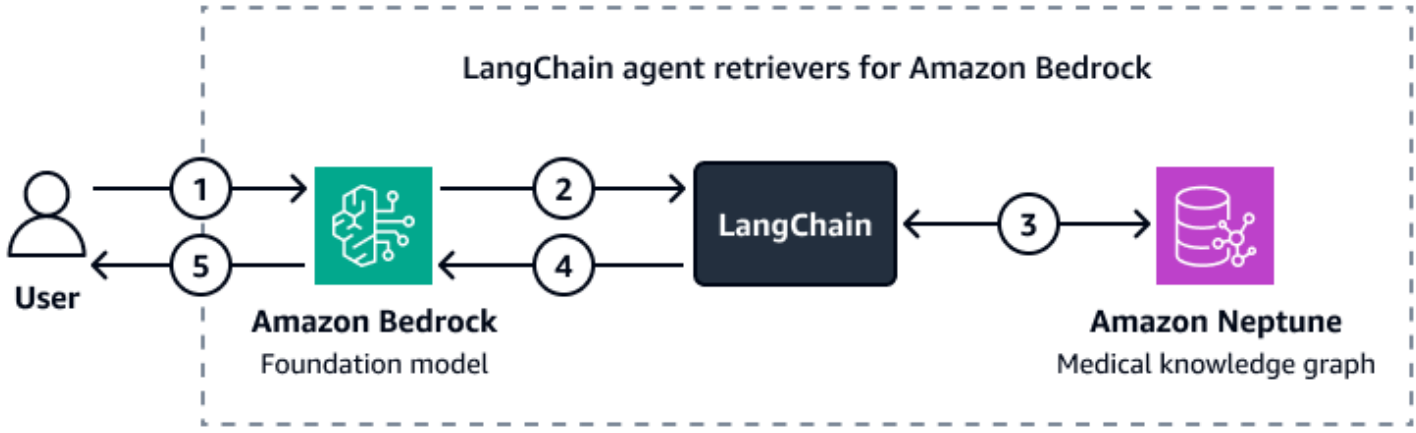



Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna mengirimkan pertanyaan ke Amazon Bedrock dan LangChain agen.
2. Model fondasi Amazon Bedrock menggunakan skema Neptunus, yang disediakan oleh LangChain agen, untuk menghasilkan kueri untuk pertanyaan pengguna.
3. Bagian LangChain agen menjalankan kueri terhadap grafik pengetahuan medis Amazon Neptunus.
4. Bagian LangChain agen mengirimkan konteks yang diambil ke model yayasan Amazon Bedrock.
5. Model dasar Amazon Bedrock menggunakan konteks yang diambil untuk menghasilkan jawaban atas pertanyaan pengguna.

# Langkah 4: Membuat basis pengetahuan data deskriptif real-time

Selanjutnya, Anda membuat basis pengetahuan catatan interaksi dokter-pasien deskriptif waktu nyata, penilaian gambar diagnostik, dan laporan analisis lab. Basis pengetahuan ini adalah database vektor. Dengan menggunakan database vektor, yang dapat menyimpan pengetahuan medis deskriptif dalam bentuk vektor yang diindeks, penyedia layanan kesehatan dapat secara efisien meminta dan mengakses informasi yang relevan dari repositori yang luas. Representasi vektor ini membantu Anda mengambil data yang serupa secara semantik. Penyedia perawatan dapat dengan cepat menavigasi melalui catatan klinis, gambar medis, dan hasil laboratorium. Ini mempercepat pengambilan keputusan berdasarkan informasi dengan menawarkan akses instan ke informasi yang relevan secara kontekstual, meningkatkan akurasi dan kecepatan diagnosis dan rencana perawatan.

## Menggunakan basis pengetahuan medis OpenSearch Layanan

Amazon OpenSearch Service dapat mengelola volume besar data medis berdimensi tinggi. Ini adalah layanan terkelola yang memfasilitasi pencarian berkinerja tinggi dan analitik real-time. Ini sangat cocok sebagai database vektor untuk aplikasi RAG. OpenSearch Layanan bertindak sebagai alat backend untuk mengelola sejumlah besar data tidak terstruktur atau semi-terstruktur, seperti catatan medis, artikel penelitian, dan catatan klinis. Kemampuan pencarian semantik canggihnya membantu Anda mengambil informasi yang relevan secara kontekstual. Ini membuatnya sangat berguna dalam aplikasi seperti sistem pendukung keputusan klinis, alat resolusi kueri pasien, dan sistem manajemen pengetahuan perawatan kesehatan. Misalnya, seorang dokter dapat dengan cepat menemukan data pasien yang relevan atau studi penelitian yang sesuai dengan gejala atau protokol pengobatan tertentu. Ini membantu dokter membuat keputusan yang diinformasikan oleh informasi yang paling up-to-date dan relevan.

OpenSearch Layanan dapat menskalakan dan menangani pengindeksan dan kueri data secara real-time. Ini membuatnya ideal untuk lingkungan perawatan kesehatan yang dinamis di mana akses tepat waktu ke informasi yang akurat sangat penting. Selain itu, ia memiliki kemampuan pencarian multi-modal yang optimal untuk pencarian yang memerlukan banyak input, seperti gambar medis dan catatan dokter. Saat menerapkan OpenSearch Layanan untuk aplikasi perawatan kesehatan, penting bagi Anda untuk menentukan bidang dan pemetaan yang tepat untuk mengoptimalkan pengindeksan dan pengambilan data. Bidang mewakili potongan data individu, seperti catatan pasien, riwayat medis, dan kode diagnostik. Pemetaan menentukan bagaimana bidang ini disimpan (dalam bentuk penyematan atau bentuk asli) dan ditanyakan. Untuk aplikasi perawatan kesehatan, penting untuk membuat pemetaan yang mengakomodasi berbagai tipe data, termasuk data terstruktur (seperti hasil tes numerik), data semi-terstruktur (seperti catatan pasien), dan data tidak terstruktur (seperti gambar medis)

Di OpenSearch Layanan, Anda dapat melakukan kueri penelusuran saraf teks lengkap melalui petunjuk yang dikuratori untuk mencari melalui catatan medis, catatan klinis, atau makalah penelitian untuk dengan cepat menemukan informasi yang relevan tentang gejala, perawatan, atau riwayat pasien tertentu. Kueri pencarian saraf secara otomatis menangani penyematan prompt input dan gambar dengan menggunakan model jaringan saraf bawaan. Ini membantunya memahami dan menangkap hubungan semantik yang lebih dalam dalam data multi-modal, menawarkan hasil pencarian yang lebih sadar konteks dan tepat dibandingkan dengan algoritme kueri penelusuran lainnya, seperti pencarian K-Nearest Neighbor (K-nN).

## Membuat arsitektur RAG

Anda dapat menerapkan solusi RAG khusus yang menggunakan agen Amazon Bedrock untuk menanyakan basis pengetahuan medis di Layanan. OpenSearch Untuk mencapai hal ini, Anda membuat AWS Lambda fungsi yang dapat berinteraksi dengan dan query OpenSearch Service. Fungsi Lambda menyematkan pertanyaan masukan pengguna dengan mengakses titik akhir penyematan teks SageMaker AI. Agen Amazon Bedrock meneruskan parameter kueri tambahan sebagai input ke fungsi Lambda. Fungsi ini menanyakan basis pengetahuan medis di OpenSearch Layanan, yang mengembalikan konten medis yang relevan. Setelah Anda mengatur fungsi Lambda, tambahkan sebagai grup tindakan dalam agen Amazon Bedrock. Agen Amazon Bedrock mengambil input pengguna, mengidentifikasi variabel yang diperlukan, meneruskan variabel dan pertanyaan ke fungsi Lambda, dan kemudian memulai fungsi. Fungsi mengembalikan konteks yang membantu model dasar memberikan jawaban yang lebih akurat untuk pertanyaan pengguna.

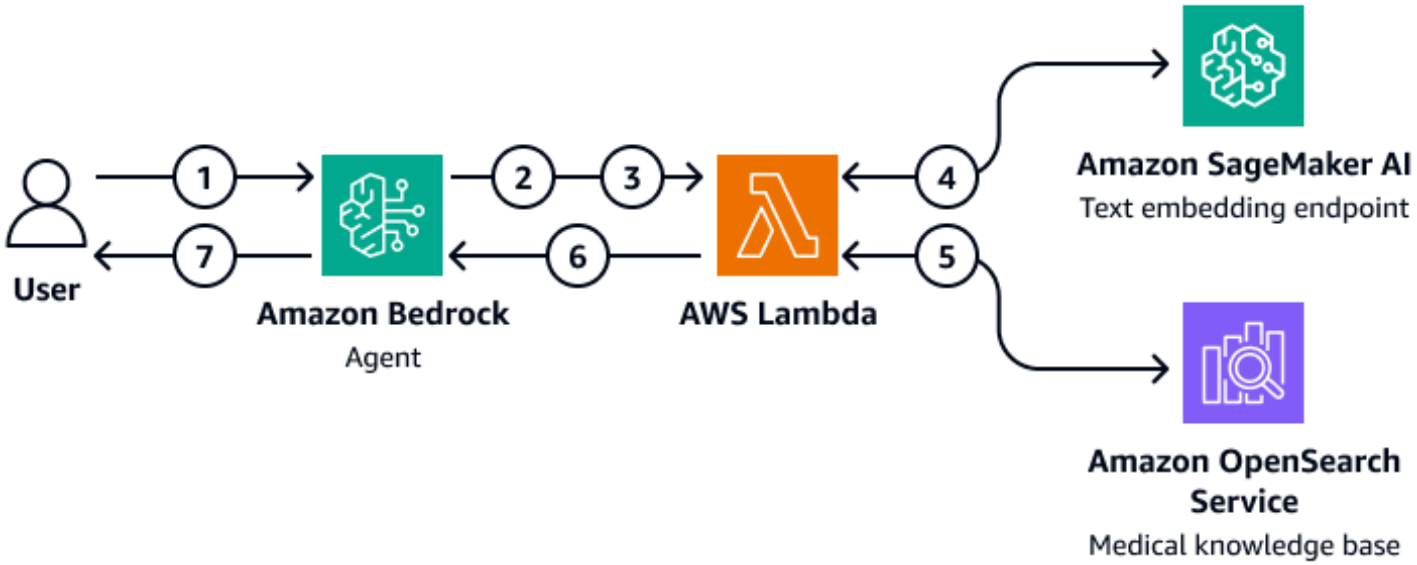



Diagram menunjukkan alur kerja berikut:

1. Seorang pengguna mengirimkan pertanyaan ke agen Amazon Bedrock.
2. Agen Amazon Bedrock memilih grup tindakan mana yang akan dimulai.
3. Agen Amazon Bedrock memulai AWS Lambda fungsi dan meneruskan parameter ke sana.
4. Fungsi Lambda memulai model penyematan teks Amazon SageMaker AI untuk menyematkan pertanyaan pengguna.
5. Fungsi Lambda meneruskan teks yang disematkan dan parameter serta filter tambahan ke Layanan Amazon OpenSearch . Amazon OpenSearch Service menanyakan basis pengetahuan medis dan mengembalikan hasilnya ke fungsi Lambda.
6. Fungsi Lambda meneruskan hasilnya kembali ke agen Amazon Bedrock.
7. Model dasar di agen Amazon Bedrock menghasilkan respons berdasarkan hasil dan mengembalikan respons kepada pengguna.

Untuk situasi di mana penyaringan yang lebih kompleks terlibat, Anda dapat menggunakan kustom LangChain retriever. Buat retriever ini dengan menyiapkan klien pencarian vektor OpenSearch Layanan yang dimuat langsung ke LangChain. Arsitektur ini memungkinkan Anda untuk melewati lebih banyak variabel untuk membuat parameter filter. Setelah retriever diatur, gunakan model Amazon Bedrock dan retriever untuk menyiapkan rantai penjawab pertanyaan pengambilan. Rantai ini mengatur interaksi antara model dan retriever dengan meneruskan input pengguna dan filter potensial ke retriever. Retriever mengembalikan konteks yang relevan yang membantu model dasar menjawab pertanyaan pengguna.

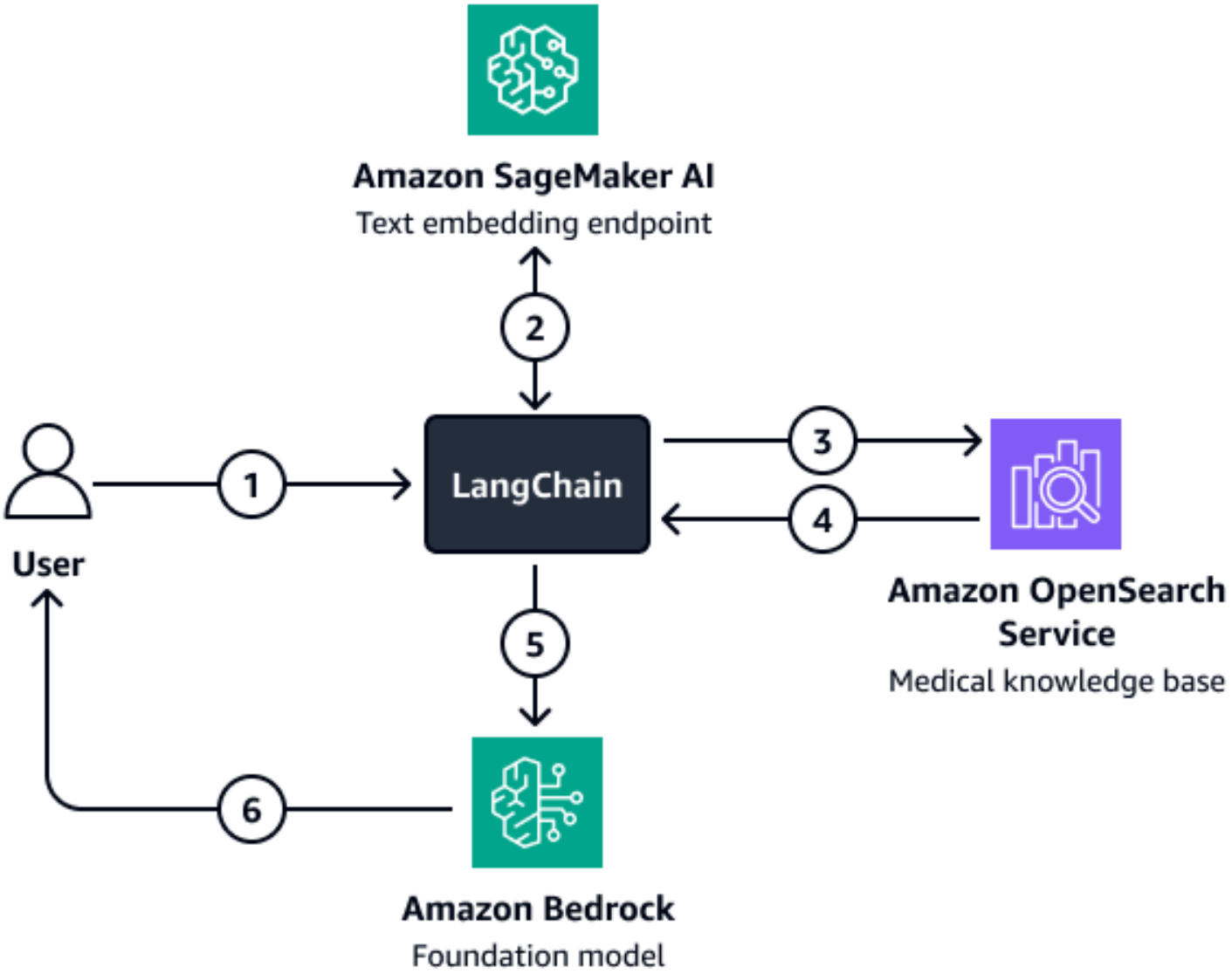



Diagram menunjukkan alur kerja berikut:

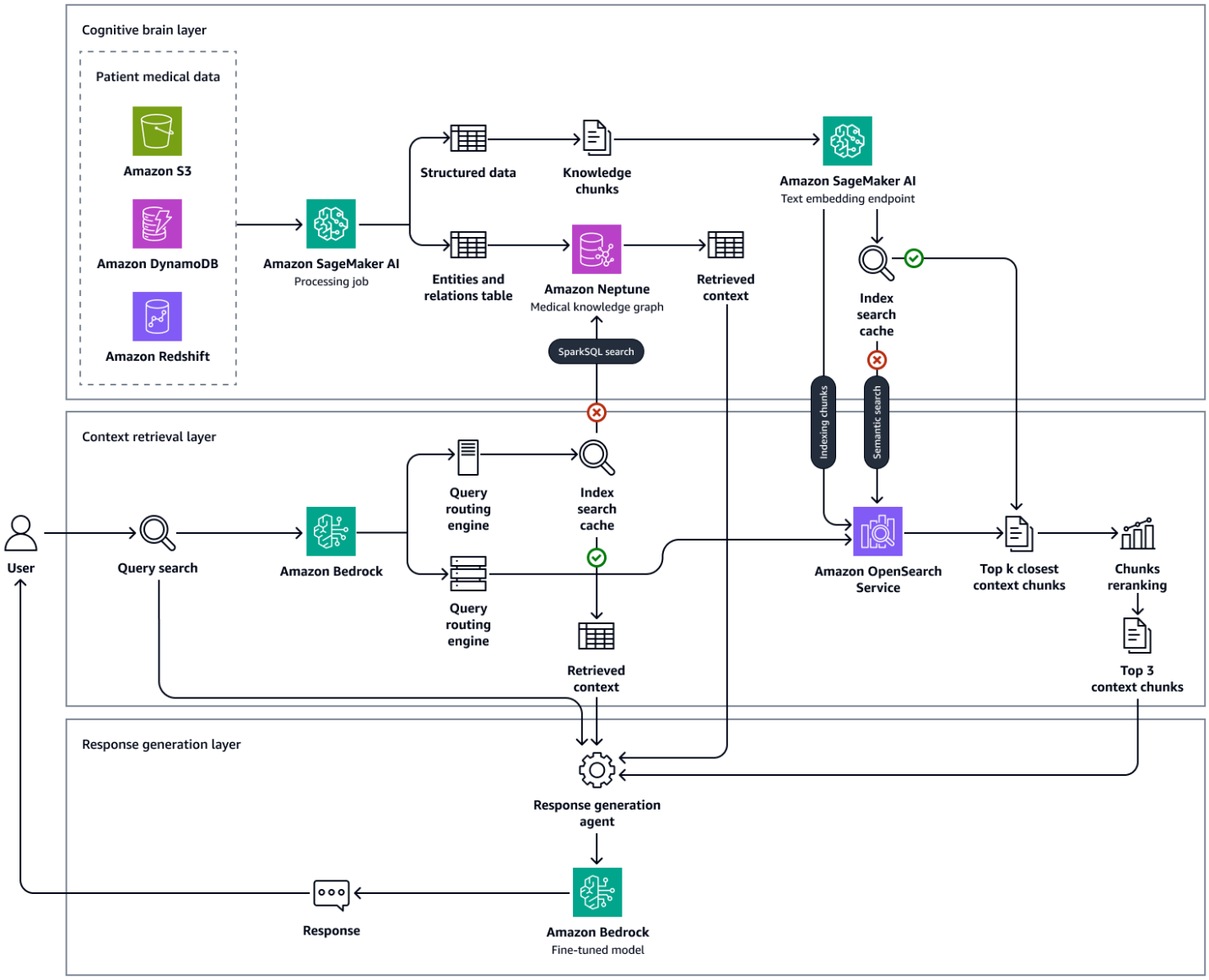
1. Seorang pengguna mengajukan pertanyaan ke LangChain agen retriever.
2. Bagian LangChain agen retriever mengirimkan pertanyaan ke titik akhir penyematan teks Amazon SageMaker AI untuk menyematkan pertanyaan.
3. Bagian LangChain agen retriever meneruskan teks yang disematkan ke Amazon OpenSearch Service.
4. Amazon OpenSearch Service mengembalikan dokumen yang diambil ke LangChain agen retriever.
5. Bagian LangChain agen retriever meneruskan pertanyaan pengguna dan mengambil konteks ke model dasar Amazon Bedrock.

6. Model foundation menghasilkan respons dan mengirimkannya ke pengguna.

# Langkah 5: Gunakan LLMs untuk menjawab pertanyaan medis

Langkah-langkah sebelumnya membantu Anda membangun aplikasi intelijen medis yang dapat mengambil catatan medis pasien dan merangkum obat-obatan yang relevan dan diagnosis potensial. Sekarang, Anda membangun layer generasi. Lapisan ini menggunakan kemampuan generatif LLM di Amazon Bedrock, seperti Llama 3, untuk menambah output aplikasi.

Ketika seorang dokter memasukkan kueri, lapisan pengambilan konteks aplikasi melakukan proses pengambilan dari grafik pengetahuan dan mengembalikan catatan teratas yang berkaitan dengan riwayat, demografi, gejala, diagnosis, dan hasil pasien. Dari database vektor, ia juga mengambil catatan interaksi dokter-pasien deskriptif waktu nyata, wawasan penilaian gambar diagnostik, ringkasan laporan analisis laboratorium, dan wawasan dari kumpulan besar penelitian medis dan buku akademik. Hasil teratas yang diambil ini, kueri dokter, dan petunjuknya (yang disesuaikan untuk menyusun jawaban berdasarkan sifat kueri), kemudian diteruskan ke model dasar di Amazon Bedrock. Ini adalah lapisan generasi respons. LLM menggunakan konteks yang diambil untuk menghasilkan respons terhadap permintaan dokter. Gambar berikut menunjukkan end-to-end alur kerja langkah-langkah dalam solusi ini.




Anda dapat menggunakan model dasar pra-terlatih di Amazon Bedrock, seperti Llama 3, untuk berbagai kasus penggunaan yang harus ditangani oleh aplikasi intelijen medis. LLM yang paling efektif untuk tugas tertentu bervariasi tergantung pada kasus penggunaan. Misalnya, model pra-pelatihan mungkin cukup untuk meringkas percakapan pasien-dokter, mencari melalui obat-obatan dan riwayat pasien, dan mengambil wawasan dari kumpulan data medis internal dan badan pengetahuan ilmiah. Namun, LLM yang disetel dengan baik mungkin diperlukan untuk kasus penggunaan kompleks lainnya, seperti evaluasi laboratorium waktu nyata, rekomendasi prosedur medis, dan prediksi hasil pasien. Anda dapat menyempurnakan LLM dengan melatihnya pada kumpulan data domain medis. Persyaratan perawatan kesehatan dan ilmu hayati yang spesifik atau kompleks mendorong pengembangan model yang disetel dengan baik ini.

Untuk informasi lebih lanjut tentang menyempurnakan LLM atau memilih LLM yang ada yang telah dilatih pada data domain medis, lihat Menggunakan model bahasa besar untuk perawatan kesehatan dan kasus penggunaan ilmu hayati.

## Penyelarasan dengan Kerangka AWS Well-Architected

Solusinya sejalan dengan keenam pilar Kerangka AWS Well-Architected sebagai berikut:

- Keunggulan operasional — Arsitektur dipisahkan untuk pemantauan dan pembaruan yang efisien. Agen Amazon Bedrock dan AWS Lambda membantu Anda menyebarkan dan memutar kembali alat dengan cepat.
- Keamanan — Solusi ini dirancang untuk mematuhi peraturan perawatan kesehatan, seperti HIPAA. Anda juga dapat menerapkan enkripsi, kontrol akses berbutir halus, dan pagar pembatas Amazon Bedrock untuk membantu melindungi data pasien.
- Keandalan - layanan AWS terkelola, seperti Amazon OpenSearch Service dan Amazon Bedrock, menyediakan infrastruktur untuk interaksi model berkelanjutan.
- Efisiensi kinerja — Solusi RAG mengambil data yang relevan dengan cepat menggunakan pencarian semantik yang dioptimalkan dan kueri Cypher, sementara router agen mengidentifikasi rute optimal untuk kueri pengguna.
- Optimalisasi biaya — pay-per-token Model dalam arsitektur Amazon Bedrock dan RAG mengurangi biaya inferensi dan pra-pelatihan.
- Keberlanjutan — Menggunakan infrastruktur tanpa server dan pay-per-token komputasi meminimalkan penggunaan sumber daya dan meningkatkan keberlanjutan.

# Kasus penggunaan: Memprediksi hasil pasien dan tingkat penerimaan ulang

Analisis prediktif bertenaga AI menawarkan manfaat lebih lanjut dengan meramalkan hasil pasien dan memungkinkan rencana perawatan yang dipersonalisasi. Hal ini dapat meningkatkan kepuasan pasien dan hasil kesehatan. Dengan mengintegrasikan kemampuan AI ini dengan Amazon Bedrock dan teknologi lainnya, penyedia layanan kesehatan dapat mencapai peningkatan produktivitas yang signifikan, mengurangi biaya, dan meningkatkan kualitas perawatan pasien secara keseluruhan.

Anda dapat menyimpan data medis, seperti riwayat pasien, catatan klinis, obat-obatan, dan perawatan, dalam grafik pengetahuan. Dengan menggabungkan pemahaman kontekstual yang mendalam LLMs dengan data temporal terstruktur dalam grafik pengetahuan medis, penyedia layanan kesehatan dapat memperoleh wawasan tambahan tentang pola pasien individu. Dengan menggunakan analitik prediktif, Anda dapat mengidentifikasi potensi ketidakpatuhan atau komplikasi pengobatan sejak dini dan menghasilkan skor kecenderungan masuk kembali yang dipersonalisasi.

Solusi ini membantu Anda memprediksi kemungkinan masuk kembali. Prediksi ini dapat meningkatkan hasil pasien dan mengurangi biaya perawatan kesehatan. Solusi ini juga dapat membantu dokter dan administrator rumah sakit memusatkan perhatian mereka pada pasien dengan risiko masuk kembali yang lebih tinggi. Ini juga membantu mereka memulai intervensi proaktif dengan pasien tersebut melalui tindakan peringatan, layanan mandiri, dan berbasis data.

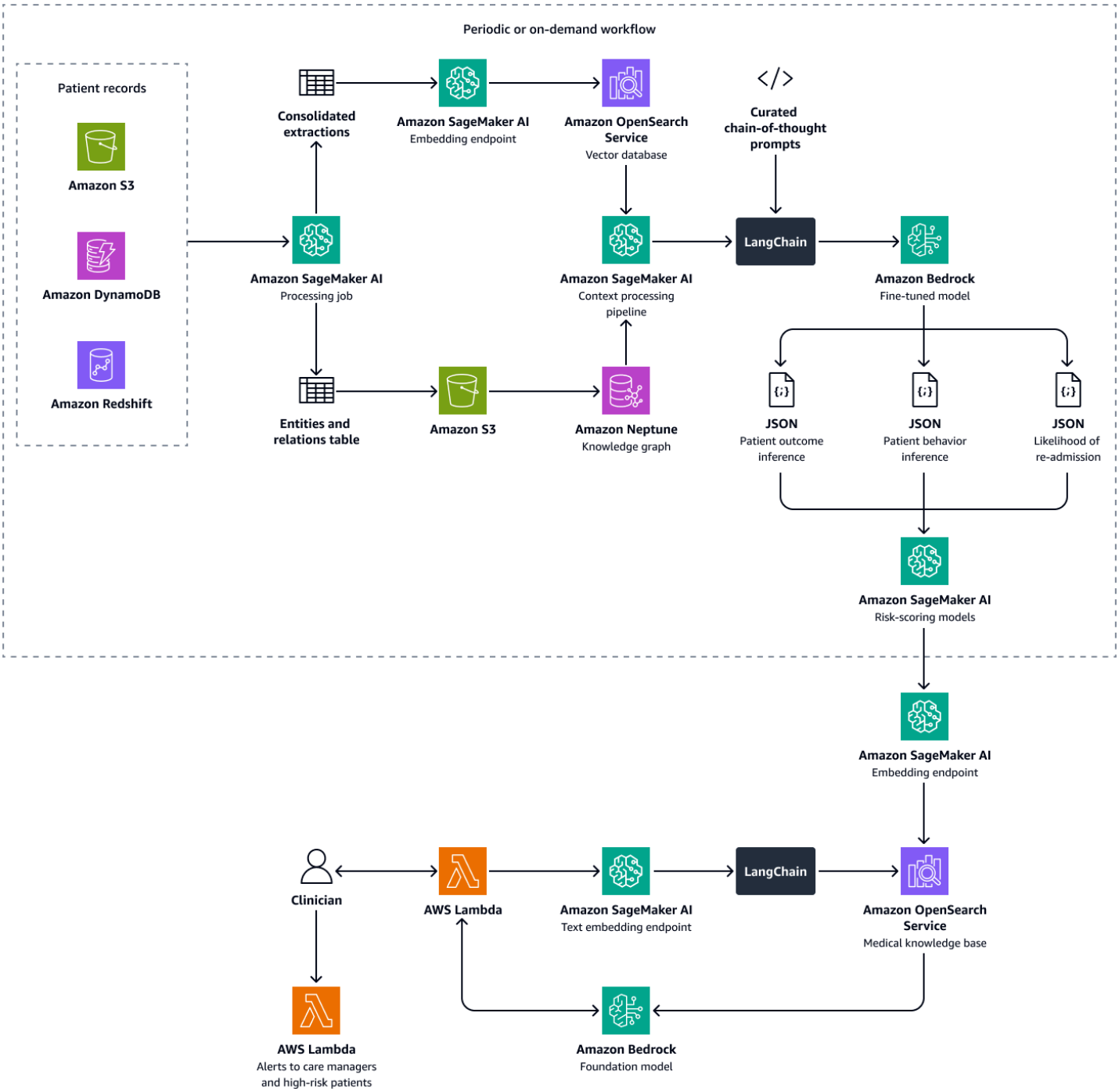
## Ikhtisar solusi

Solusi ini menggunakan kerangka kerja multi-retriever Retrieval Augmented Generation (RAG) untuk menganalisis data pasien. Ini memprediksi kemungkinan masuk kembali rumah sakit untuk masing-masing pasien dan membantu Anda menghitung skor kecenderungan masuk kembali tingkat rumah sakit. Solusi ini mengintegrasikan fitur-fitur berikut:

- Grafik pengetahuan — Menyimpan data pasien kronologis yang terstruktur, seperti pertemuan rumah sakit, penerimaan ulang sebelumnya, gejala, hasil laboratorium, perawatan yang ditentukan, dan riwayat kepatuhan pengobatan
- Database vektor - Menyimpan data klinis yang tidak terstruktur, seperti ringkasan pelepasan, catatan dokter, dan catatan janji yang terlewat atau efek samping obat yang dilaporkan

- LLM yang disetel dengan baik - Mengkonsumsi data terstruktur dari grafik pengetahuan dan data tidak terstruktur dari database vektor untuk menghasilkan kesimpulan tentang perilaku pasien, kepatuhan pengobatan, dan kemungkinan masuk kembali

Model penilaian risiko mengukur kesimpulan dari LLM menjadi skor numerik. Anda dapat menggabungkan skor ke dalam skor kecenderungan masuk kembali tingkat rumah sakit. Skor ini menentukan paparan risiko setiap pasien, dan Anda dapat menghitungnya secara berkala atau sesuai kebutuhan. Semua kesimpulan dan skor risiko diindeks dan disimpan di OpenSearch Layanan Amazon sehingga manajer perawatan dan dokter dapat mengambilnya. Dengan mengintegrasikan agen AI percakapan dengan database vektor ini, dokter dan manajer perawatan dapat dengan mulus mengekstrak wawasan pada tingkat pasien individu, tingkat fasilitas, atau dengan spesialisasi medis. Anda juga dapat mengatur peringatan otomatis berdasarkan skor risiko, yang mendorong intervensi proaktif.




Membangun solusi ini terdiri dari langkah-langkah berikut:

- Langkah 1: Memprediksi hasil pasien dengan menggunakan grafik pengetahuan medis
- Langkah 2: Memprediksi perilaku pasien terhadap obat atau perawatan yang diresepkan
- Langkah 3: Memprediksi kemungkinan masuk kembali pasien

# Langkah 1: Memprediksi hasil pasien dengan menggunakan grafik pengetahuan medis

Di Amazon Neptunus, Anda dapat menggunakan grafik pengetahuan untuk menyimpan pengetahuan temporal tentang kunjungan dan hasil pasien dari waktu ke waktu. Cara paling efektif untuk membangun dan menyimpan grafik pengetahuan adalah dengan menggunakan model grafik dan database grafik. Database grafik dibuat khusus untuk menyimpan dan menavigasi hubungan. Database grafik membuatnya lebih mudah untuk memodelkan dan mengelola data yang sangat terhubung dan memiliki skema yang fleksibel.

Grafik pengetahuan membantu Anda melakukan analisis deret waktu. Berikut ini adalah elemen kunci dari database grafik yang digunakan untuk prediksi temporal hasil pasien:

- Data historis — Diagnosis sebelumnya, pengobatan lanjutan, obat yang sebelumnya digunakan, dan hasil laboratorium untuk pasien
- Kunjungan pasien (kronologis) - Tanggal kunjungan, gejala, alergi yang diamati, catatan klinis, diagnosis, prosedur, perawatan, obat yang diresepkan, dan hasil laboratorium
- Gejala dan parameter klinis — Informasi klinis dan berbasis gejala, termasuk tingkat keparahan, pola perkembangan, dan respons pasien terhadap obat

Anda dapat menggunakan wawasan dari grafik pengetahuan medis untuk menyempurnakan LLM di Amazon Bedrock, seperti Llama 3. Anda menyempurnakan LLM dengan data pasien berurutan tentang respons pasien terhadap serangkaian obat atau perawatan dari waktu ke waktu. Gunakan kumpulan data berlabel yang mengklasifikasikan satu set obat atau perawatan dan data interaksi pasien-klinik ke dalam kategori yang telah ditentukan sebelumnya yang menunjukkan status kesehatan pasien. Contoh dari kategori ini adalah penurunan kesehatan, peningkatan, atau kemajuan yang stabil. Ketika dokter memasukkan konteks baru tentang pasien dan gejalanya, LLM yang disetel dengan baik dapat menggunakan pola dari kumpulan data pelatihan untuk memprediksi potensi hasil pasien.

Gambar berikut menunjukkan langkah-langkah berurutan yang terlibat dalam menyempurnakan LLM di Amazon Bedrock dengan menggunakan kumpulan data pelatihan khusus perawatan kesehatan. Data ini mungkin termasuk kondisi medis pasien dan respons terhadap perawatan dari waktu ke

waktu. Dataset pelatihan ini akan membantu model untuk membuat prediksi umum tentang hasil pasien.

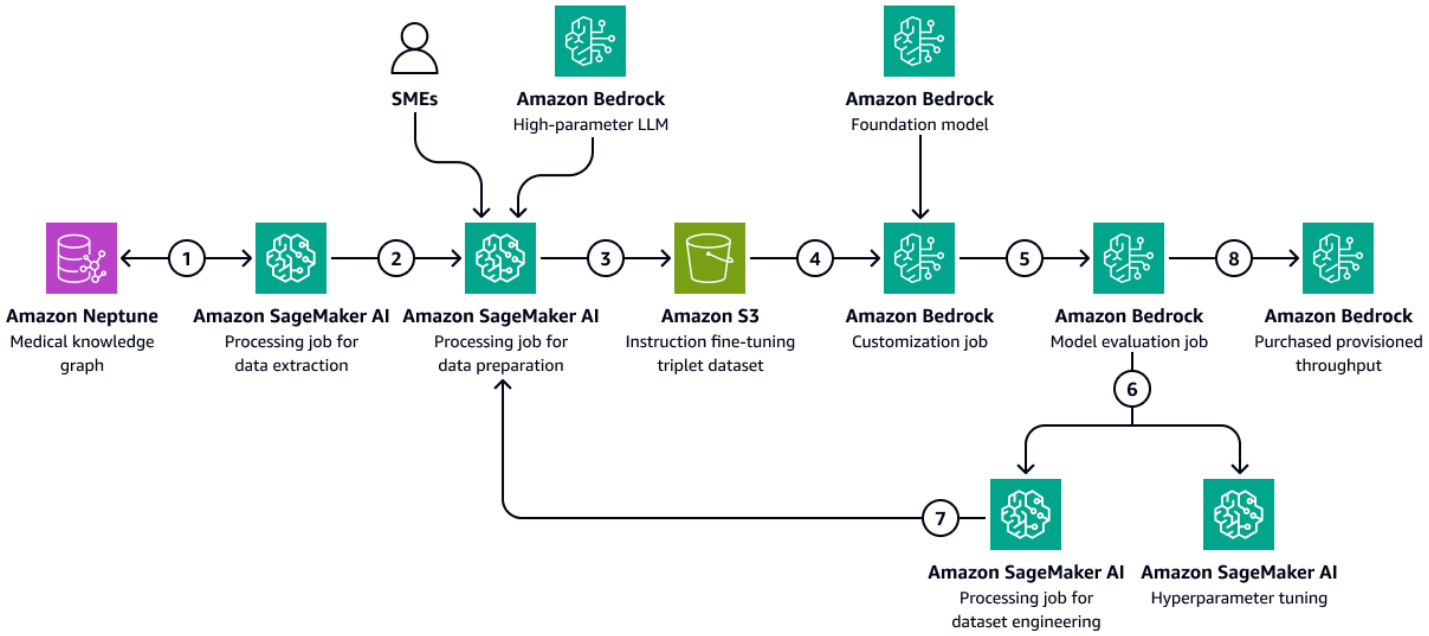



Diagram menunjukkan alur kerja berikut:

1. Pekerjaan ekstraksi data SageMaker AI Amazon menanyakan grafik pengetahuan untuk mengambil data kronologis tentang respons pasien yang berbeda terhadap serangkaian obat atau perawatan dari waktu ke waktu.
2. Pekerjaan persiapan data SageMaker AI mengintegrasikan Amazon Bedrock LLM dan masukan dari pakar materi pelajaran (). SMEs Pekerjaan mengklasifikasikan data yang diambil dari grafik pengetahuan ke dalam kategori yang telah ditentukan (seperti penurunan kesehatan, peningkatan, atau kemajuan yang stabil) yang menunjukkan status kesehatan setiap pasien.
3. Pekerjaan membuat kumpulan data fine-tuning yang mencakup informasi yang diekstrak dari grafik pengetahuan, chain-of-thought petunjuk, dan kategori hasil pasien. Ini mengunggah kumpulan data pelatihan ini ke bucket Amazon S3.
4. Pekerjaan kustomisasi Amazon Bedrock menggunakan kumpulan data pelatihan ini untuk menyempurnakan LLM.
5. Pekerjaan kustomisasi Amazon Bedrock mengintegrasikan model dasar pilihan Amazon Bedrock dalam lingkungan pelatihan. Ini memulai pekerjaan fine-tuning dan menggunakan dataset pelatihan dan hyperparameter pelatihan yang Anda konfigurasikan.
6. Pekerjaan evaluasi Amazon Bedrock mengevaluasi model yang disetel dengan menggunakan kerangka evaluasi model yang telah dirancang sebelumnya.

7. Jika model membutuhkan perbaikan, pekerjaan pelatihan berjalan lagi dengan lebih banyak data setelah mempertimbangkan kumpulan data pelatihan dengan cermat. Jika model tidak menunjukkan peningkatan kinerja tambahan, pertimbangkan juga untuk memodifikasi hiperparameter pelatihan.
8. Setelah evaluasi model memenuhi standar yang ditentukan oleh pemangku kepentingan bisnis, Anda meng-host model yang disetel dengan baik ke throughput yang disediakan Amazon Bedrock.

## Langkah 2: Memprediksi perilaku pasien terhadap obat atau perawatan yang diresepkan

Fine-tuned LLMs dapat memproses catatan klinis, ringkasan pelepasan, dan dokumen khusus pasien lainnya dari grafik pengetahuan medis temporal. Mereka dapat menilai apakah pasien cenderung mengikuti obat atau perawatan yang diresepkan.

Langkah ini menggunakan grafik pengetahuan yang dibuat di[Langkah 1: Memprediksi hasil pasien dengan menggunakan grafik pengetahuan medis](). Grafik pengetahuan berisi data dari profil pasien, termasuk kepatuhan historis pasien sebagai simpul. Ini juga mencakup contoh ketidakpatuhan terhadap obat-obatan atau perawatan, efek samping terhadap obat-obatan, kurangnya akses atau hambatan biaya untuk obat-obatan, atau rejimen dosis kompleks sebagai atribut dari node tersebut.

Fine-tuned LLMs dapat mengkonsumsi data pemenuhan resep sebelumnya dari grafik pengetahuan medis dan ringkasan deskriptif dari catatan klinis dari database vektor Layanan Amazon. OpenSearch Catatan klinis ini mungkin menyebutkan janji yang sering terlewatkan atau ketidakpatuhan terhadap perawatan. LLM dapat menggunakan catatan ini untuk memprediksi kemungkinan ketidakpatuhan di masa depan.

1. Siapkan data input sebagai berikut:
   - Data terstruktur — Ekstrak data pasien terbaru, seperti tiga kunjungan terakhir dan hasil lab, dari grafik pengetahuan medis.
   - Data tidak terstruktur - Ambil catatan klinis terbaru dari database vektor OpenSearch Layanan Amazon.
2. Buat prompt input yang mencakup riwayat pasien dan konteks saat ini. Berikut ini adalah contoh prompt:

```
You are a highly specialized AI model trained in healthcare predictive analytics.
 Your task is to analyze a patient's historical medical records, adherence patterns,
```

```
 and clinical context to predict the **likelihood of future non-adherence** to
 prescribed medications or treatments.

### **Patient Details**
- **Patient ID:** {patient_id}
- **Age:** {age}
- **Gender:** {gender}
- **Medical Conditions:** {medical_conditions}
- **Current Medications:** {current_medications}
- **Prescribed Treatments:** {prescribed_treatments}

### **Chronological Medical History**
- **Visit Dates & Symptoms:** {visit_dates_symptoms}
- **Diagnoses & Procedures:** {diagnoses_procedures}
- **Prescribed Medications & Treatments:** {medications_treatments}
- **Past Adherence Patterns:** {historical_adherence}
- **Instances of Non-Adherence:** {past_non_adherence}
- **Side Effects Experienced:** {side_effects}
- **Barriers to Adherence (e.g., Cost, Access, Dosing Complexity):** {barriers}

### **Patient-Specific Insights**
- **Clinical Notes & Discharge Summaries:** {clinical_notes}
- **Missed Appointments & Non-Compliance Patterns:** {missed_appointments}

### **Let's think Step-by-Step to predict the patient behaviour**
1. You should first analyze past adherence trends and patterns of non-adherence.
2. Identify potential barriers, such as financial constraints, medication side
 effects, or complex dosing regimens.
3. Thoroughly examine clinical notes and documented patient behaviors that may hint
 at non-adherence.
4. Correlate adherence history with prescribed treatments and patient conditions.
5. Finally predict the likelihood of non-adherence based on these contextual
 insights.

### **Output Format (JSON)**
Return the prediction in the following structured format:
```json
{
  "patient_id": "{patient_id}",
  "likelihood_of_non_adherence": "{low | moderate | high}",
  "reasoning": "{detailed_explanation_based_on_patient_history}"
}
```
```

3. Lulus prompt ke LLM yang disetel dengan baik. LLM memproses prompt dan memprediksi hasilnya. Berikut ini adalah contoh respon dari LLM:

```
{
  "patient_id": "P12345",
  "likelihood_of_non_adherence": "high",
  "reasoning": "The patient has a history of missed appointments, has reported side
 effects to previous medications. Additionally, clinical notes indicate difficulty
 following complex dosing schedules."
}
```

4. Urai respons model untuk mengekstrak kategori hasil yang diprediksi. Misalnya, kategori untuk respons contoh pada langkah sebelumnya mungkin memiliki kemungkinan ketidakpatuhan yang tinggi.
5. (Opsional) Gunakan log model atau metode tambahan untuk menetapkan skor kepercayaan. Logit adalah probabilitas yang tidak dinormalisasi dari item yang termasuk dalam kelas atau kategori tertentu.

## Langkah 3: Memprediksi kemungkinan masuk kembali pasien

Penerimaan ulang rumah sakit menjadi perhatian utama karena tingginya biaya administrasi perawatan kesehatan dan karena dampaknya terhadap kesejahteraan pasien. Menghitung tingkat penerimaan ulang rumah sakit adalah salah satu cara untuk mengukur kualitas perawatan pasien dan kinerja penyedia layanan kesehatan.

Untuk menghitung tingkat penerimaan ulang, Anda menentukan indikator, seperti tarif masuk ulang 7 hari. Indikator ini adalah persentase pasien rawat inap yang kembali ke rumah sakit untuk kunjungan yang tidak direncanakan dalam waktu tujuh hari setelah keluar. Untuk memprediksi kemungkinan masuk kembali untuk pasien, LLM yang disetel dengan baik dapat mengkonsumsi data temporal dari grafik pengetahuan medis yang Anda buat. Langkah 1: Memprediksi hasil pasien dengan menggunakan grafik pengetahuan medis Grafik pengetahuan ini menyimpan catatan kronologis pertemuan pasien, prosedur, pengobatan, dan gejala. Catatan data ini berisi yang berikut:

- Durasi waktu sejak keputihan terakhir pasien
- Respon pasien terhadap perawatan dan pengobatan masa lalu
- Perkembangan gejala atau kondisi dari waktu ke waktu

Anda dapat memproses peristiwa deret waktu ini untuk memprediksi kemungkinan masuk kembali pasien melalui prompt sistem yang dikuratori. Prompt menanamkan logika prediksi ke LLM yang disetel dengan baik.

1. Siapkan data input sebagai berikut:
   - Riwayat kepatuhan - Ekstrak tanggal pengambilan obat, frekuensi isi ulang obat, diagnosis dan detail pengobatan, riwayat medis kronologis, dan informasi lain dari grafik pengetahuan medis.
   - Indikator perilaku - Ambil dan sertakan catatan klinis tentang janji yang terlewat dan efek samping yang dilaporkan pasien.
2. Buat prompt input yang mencakup riwayat kepatuhan dan indikator perilaku. Berikut ini adalah contoh prompt:

```
You are a highly specialized AI model trained in healthcare predictive analytics.
 Your task is to analyze a patient's historical medical records, clinical events, and
 adherence patterns to predict the **likelihood of hospital readmission** within the
 next few days.

### **Patient Details**
- **Patient ID:** {patient_id}
- **Age:** {age}
- **Gender:** {gender}
- **Primary Diagnoses:** {diagnoses}
- **Current Medications:** {current_medications}
- **Prescribed Treatments:** {prescribed_treatments}

### **Chronological Medical History**
- **Recent Hospital Encounters:** {encounters}
- **Time Since Last Discharge:** {time_since_last_discharge}
- **Previous Readmissions:** {past_readmissions}
- **Recent Lab Results & Vital Signs:** {recent_lab_results}
- **Procedures Performed:** {procedures_performed}
- **Prescribed Medications & Treatments:** {medications_treatments}
- **Past Adherence Patterns:** {historical_adherence}
- **Instances of Non-Adherence:** {past_non_adherence}

### **Patient-Specific Insights**
- **Clinical Notes & Discharge Summaries:** {clinical_notes}
- **Missed Appointments & Non-Compliance Patterns:** {missed_appointments}
- **Patient-Reported Side Effects & Complications:** {side_effects}

### **Reasoning Process – You have to analyze this use case step-by-step.**
```

```
1. First assess **time since last discharge** and whether recent hospital encounters
 suggest a pattern of frequent readmissions.
2. Second examine **recent lab results, vital signs, and procedures performed** to
 identify clinical deterioration.
3. Third analyze **adherence history**, checking if past non-adherence to medications
 or treatments correlates with readmissions.
4. Then identify **missed appointments, self-reported side effects, or symptoms
 worsening** from clinical notes.
5. Finally predict the **likelihood of readmission** based on these contextual
 insights.

### **Output Format (JSON)**
Return the prediction in the following structured format:
```json
{
  "patient_id": "{patient_id}",
  "likelihood_of_readmission": "{low | moderate | high}",
  "reasoning": "{detailed_explanation_based_on_patient_history}"
}
```

3. Lulus prompt ke LLM yang disetel dengan baik. LLM memproses prompt dan memprediksi kemungkinan dan alasan masuk kembali. Berikut ini adalah contoh respon dari LLM:

```
{
  "patient_id": "P67890",
  "likelihood_of_readmission": "high",
  "reasoning": "The patient was discharged only 5 days ago, has a history of more
 than two readmissions to hospitals where the patient received treatment. Recent
 lab results indicate abnormal kidney function and high liver enzymes. These factors
 suggest a medium risk of readmission."
}
```

4. Kategorikan prediksi ke dalam skala standar, seperti rendah, sedang, atau tinggi.
5. Tinjau alasan yang diberikan oleh LLM, dan identifikasi faktor-faktor kunci yang berkontribusi pada prediksi.
6. Petakan output kualitatif ke skor kuantitatif. Misalnya, sangat tinggi mungkin sama dengan probabilitas 0,9.
7. Gunakan kumpulan data validasi untuk mengkalibrasi keluaran model terhadap tingkat penerimaan ulang aktual.
```

# Langkah 4: Menghitung skor kecenderungan masuk kembali rumah sakit

Selanjutnya, Anda menghitung skor kecenderungan masuk kembali rumah sakit per pasien. Skor ini mencerminkan dampak bersih dari tiga analisis yang dilakukan pada langkah-langkah sebelumnya: hasil pasien potensial, perilaku pasien terhadap pengobatan dan perawatan, dan kemungkinan masuk kembali pasien. Dengan menggabungkan skor kecenderungan masuk kembali tingkat pasien ke tingkat khusus dan kemudian di tingkat rumah sakit, Anda dapat memperoleh wawasan kepada dokter, manajer perawatan, dan administrator. Skor kecenderungan masuk kembali rumah sakit membantu Anda menilai kinerja keseluruhan berdasarkan fasilitas, spesialisasi, atau berdasarkan kondisi. Kemudian, Anda dapat menggunakan skor ini untuk menerapkan intervensi proaktif.

1. Tetapkan bobot untuk masing-masing faktor yang berbeda (prediksi hasil, kemungkinan kepatuhan, penerimaan kembali). Berikut ini adalah contoh bobot:
   - Berat Prediksi Hasil: 0.4
   - Berat Prediksi Kepatuhan: 0.3
   - Berat Kemungkinan Masuk Kembali: 0,3
2. Gunakan perhitungan berikut untuk menghitung skor komposit:

   `ReadadmissionPropensityScore` = (`OutcomeScore` × `OutcomeWeight`) + (`AdherenceScore` × `AdherenceWeight`) + (`ReadmissionLikelihoodScore` × `ReadmissionLikelihoodWeight`)
3. Pastikan bahwa semua skor individu berada pada skala yang sama, seperti 0 hingga 1.
4. Tentukan ambang batas untuk tindakan. Misalnya, skor di atas 0,7 memulai peringatan.

Berdasarkan analisis di atas dan skor kecenderungan masuk kembali pasien, dokter atau manajer perawatan dapat mengatur peringatan untuk memantau pasien masing-masing berdasarkan skor yang dihitung. Jika berada di atas ambang batas yang telah ditentukan sebelumnya, mereka akan diberi tahu ketika ambang batas itu tercapai. Ini membantu manajer perawatan untuk menjadi proaktif daripada reaktif saat membuat rencana perawatan keputihan untuk pasien mereka. Simpan hasil pasien, perilaku, dan skor kecenderungan masuk kembali dalam bentuk yang diindeks dalam database vektor OpenSearch Layanan Amazon sehingga manajer perawatan dapat mengambilnya dengan mulus menggunakan agen AI percakapan.

Diagram berikut menunjukkan alur kerja agen AI percakapan yang dapat digunakan oleh dokter atau manajer perawatan untuk mengambil wawasan tentang hasil pasien, perilaku yang diharapkan, dan kecenderungan masuk kembali. Pengguna dapat mengambil wawasan di tingkat pasien, tingkat departemen, atau tingkat rumah sakit. Agen AI mengambil wawasan ini, yang disimpan dalam bentuk yang diindeks dalam database vektor Layanan Amazon OpenSearch . Agen menggunakan kueri untuk mengambil data yang relevan dan memberikan tanggapan yang disesuaikan, termasuk tindakan yang disarankan untuk pasien yang memiliki risiko tinggi masuk kembali. Berdasarkan tingkat risikonya, agen juga dapat mengatur pengingat untuk pasien dan pemberi perawatan.

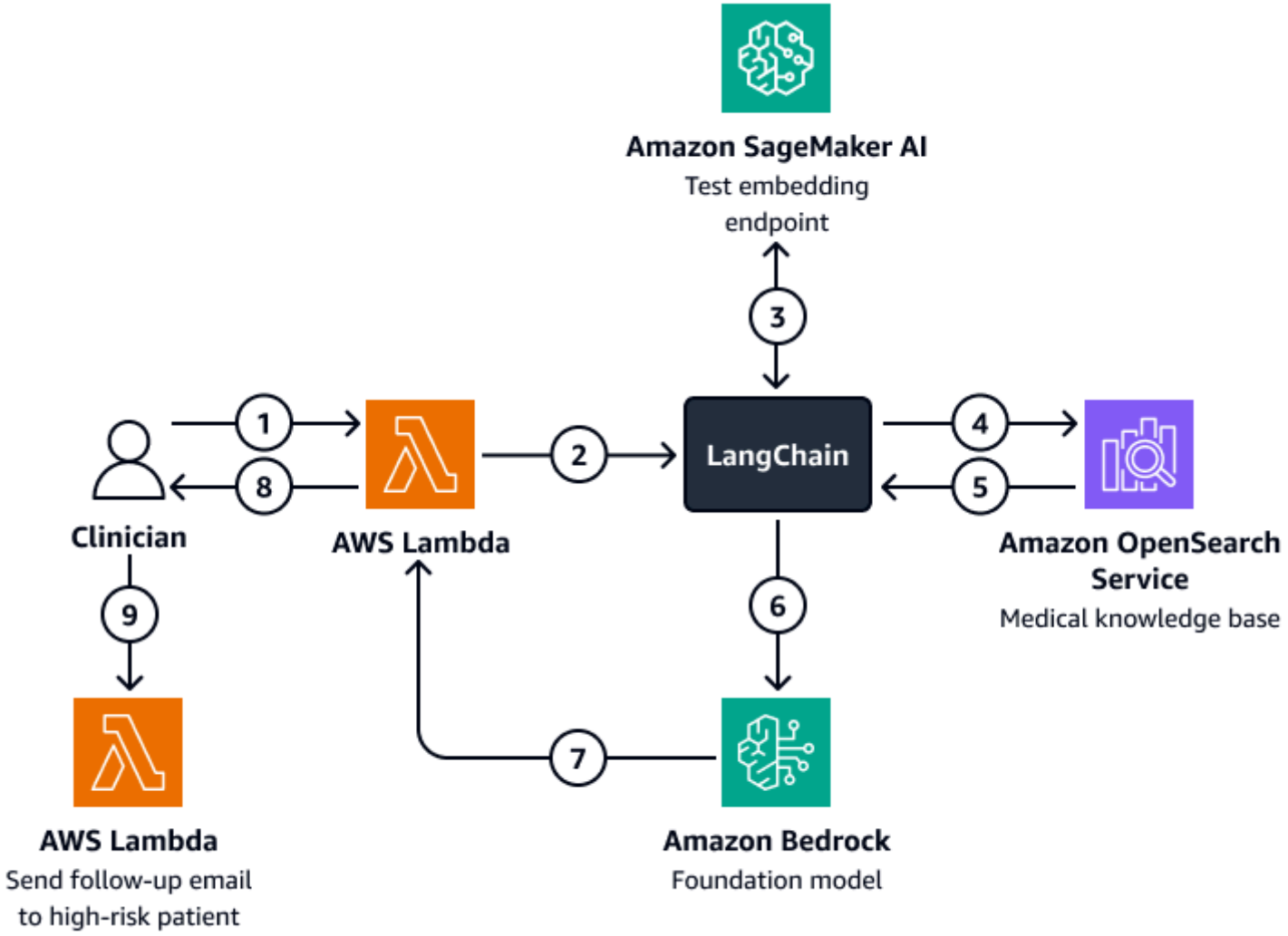



Diagram menunjukkan alur kerja berikut:

1. Dokter mengajukan pertanyaan kepada agen AI percakapan, yang menampung suatu AWS Lambda fungsi.
2. Fungsi Lambda memulai a LangChain agen.

3. Bagian LangChain agen mengirimkan pertanyaan pengguna ke titik akhir penyematan teks Amazon SageMaker AI. Titik akhir menyematkan pertanyaan.
4. Bagian LangChain agen meneruskan pertanyaan tertanam ke basis pengetahuan medis di Amazon OpenSearch Service.
5. Amazon OpenSearch Service mengembalikan wawasan spesifik yang paling relevan dengan kueri pengguna ke LangChain agen.
6. Bagian LangChain agen mengirimkan kueri dan konteks yang diambil dari basis pengetahuan ke model dasar Amazon Bedrock.
7. Model dasar Amazon Bedrock menghasilkan respons dan mengirimkannya ke fungsi Lambda.
8. Fungsi Lambda mengembalikan respons ke dokter.
9. Dokter memulai fungsi Lambda yang mengirimkan email tindak lanjut ke pasien yang memiliki risiko tinggi masuk kembali.

## Penyelarasan dengan Kerangka AWS Well-Architected

Arsitektur untuk melacak perilaku pasien dan memprediksi tingkat penerimaan ulang rumah sakit terintegrasi Layanan AWS, grafik pengetahuan medis, dan LLMs untuk meningkatkan hasil perawatan kesehatan sambil menyelaraskan dengan enam pilar Kerangka Kerja Well-Architected:AWS

- Keunggulan operasional — Solusinya adalah sistem otomatis terpisah yang menggunakan Amazon Bedrock dan AWS Lambda untuk peringatan waktu nyata.
- Keamanan — Solusi ini dirancang untuk mematuhi peraturan perawatan kesehatan, seperti HIPAA. Anda juga dapat menerapkan enkripsi, kontrol akses berbutir halus, dan pagar pembatas Amazon Bedrock untuk membantu melindungi data pasien.
- Keandalan — Arsitektur menggunakan toleran kesalahan, tanpa server. Layanan AWS
- Efisiensi kinerja - Amazon OpenSearch Service dan fine-tuned LLMs dapat memberikan prediksi yang cepat dan akurat.
- Optimalisasi biaya — Teknologi dan pay-per-inference model tanpa server membantu meminimalkan biaya. Meskipun LLM yang menggunakan fine-tuned dapat menimbulkan biaya tambahan, model menggunakan pendekatan RAG yang mengurangi data dan waktu komputasi yang diperlukan untuk proses fine-tuning.

- Keberlanjutan — Arsitektur meminimalkan konsumsi sumber daya melalui penggunaan infrastruktur tanpa server. Ini juga mendukung operasi perawatan kesehatan yang efisien dan terukur.

# Kasus penggunaan: Mengelola dan meningkatkan keterampilan staf perawatan kesehatan Anda

Menerapkan strategi transformasi bakat dan peningkatan keterampilan membantu tenaga kerja tetap mahir dalam menggunakan teknologi dan praktik baru dalam layanan medis dan perawatan kesehatan. Inisiatif peningkatan keterampilan proaktif memastikan bahwa profesional perawatan kesehatan dapat memberikan perawatan pasien berkualitas tinggi, mengoptimalkan efisiensi operasional, dan tetap mematuhi standar peraturan. Selain itu, transformasi bakat menumbuhkan budaya pembelajaran berkelanjutan. Ini sangat penting untuk beradaptasi dengan lanskap perawatan kesehatan yang berubah dan mengatasi tantangan kesehatan masyarakat yang muncul. Pendekatan pelatihan tradisional, seperti pelatihan berbasis kelas dan modul pembelajaran statis, menawarkan konten yang seragam kepada khalayak luas. Mereka sering tidak memiliki jalur pembelajaran yang dipersonalisasi, yang sangat penting untuk memenuhi kebutuhan spesifik dan tingkat kemahiran praktisi individu. one-size-fits-allStrategi ini dapat mengakibatkan pelepasan dan retensi pengetahuan yang kurang optimal.

Akibatnya, organisasi perawatan kesehatan harus merangkul solusi inovatif, terukur, dan berbasis teknologi yang dapat menentukan kesenjangan untuk setiap karyawan mereka dalam keadaan mereka saat ini dan keadaan masa depan yang potensial. Solusi ini harus merekomendasikan jalur pembelajaran yang sangat personal dan kumpulan konten pembelajaran yang tepat. Ini secara efektif mempersiapkan tenaga kerja untuk masa depan perawatan kesehatan.

Dalam industri perawatan kesehatan, Anda dapat menerapkan AI generatif untuk membantu Anda memahami dan meningkatkan tenaga kerja Anda. Melalui koneksi model bahasa besar (LLMs) dan retriever tingkat lanjut, organisasi dapat memahami keterampilan apa yang mereka miliki saat ini dan mengidentifikasi keterampilan kunci yang mungkin diperlukan di masa depan. Informasi ini membantu Anda menjembatani kesenjangan dengan mempekerjakan pekerja baru dan meningkatkan keterampilan tenaga kerja saat ini. Menggunakan Amazon Bedrock dan grafik pengetahuan, organisasi perawatan kesehatan dapat mengembangkan aplikasi khusus domain yang memfasilitasi pembelajaran berkelanjutan dan pengembangan keterampilan.

Pengetahuan yang diberikan oleh solusi ini membantu Anda mengelola bakat secara efektif, mengoptimalkan kinerja tenaga kerja, mendorong kesuksesan organisasi, mengidentifikasi keterampilan yang ada, dan menyusun strategi bakat. Solusi ini dapat membantu Anda melakukan tugas-tugas ini dalam beberapa minggu, bukan bulan.

# Ikhtisar solusi

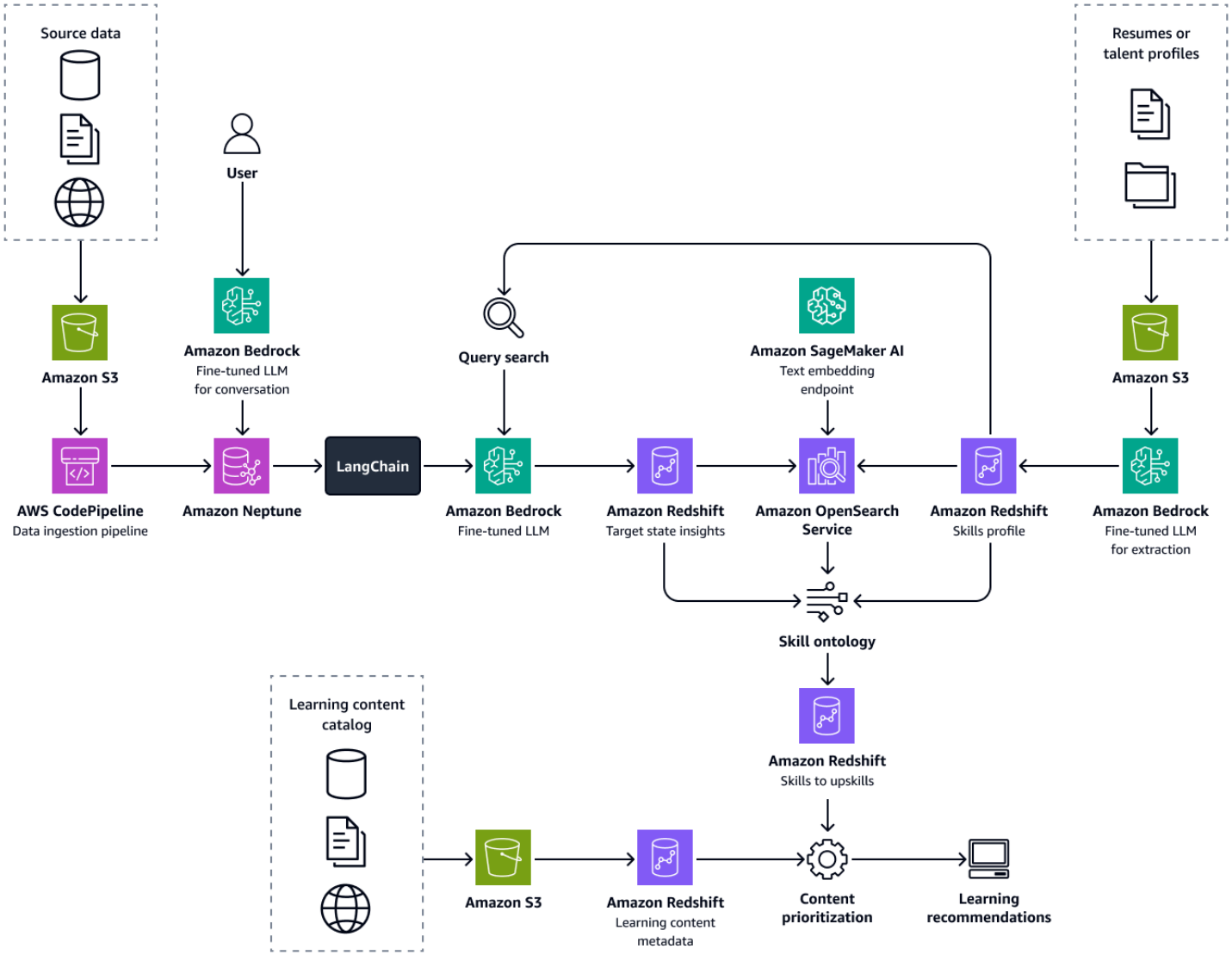
Solusi ini adalah kerangka transformasi bakat kesehatan yang terdiri dari komponen-komponen berikut:

- Parser resume cerdas - Komponen ini dapat membaca resume kandidat dan secara tepat mengekstrak informasi kandidat, termasuk keterampilan. Solusi ekstraksi informasi cerdas yang dibuat menggunakan model Llama 2 yang disetel dengan baik di Amazon Bedrock pada kumpulan data pelatihan eksklusif yang mencakup resume dan profil bakat dari lebih dari 19 industri. Proses berbasis LLM ini menghemat ratusan jam dengan mengotomatiskan proses peninjauan manual resume dan mencocokkan kandidat teratas untuk membuka peran.
- Grafik pengetahuan — Grafik pengetahuan yang dibangun di Amazon Neptunus, gudang informasi bakat terpadu termasuk taksonomi peran dan keterampilan organisasi serta industri, menangkap semantik bakat perawatan kesehatan menggunakan definisi keterampilan, peran dan propertinya, hubungan, dan kendala logis.
- Ontologi keterampilan — Penemuan kedekatan keterampilan antara keterampilan kandidat dan keadaan ideal saat ini atau keterampilan keadaan masa depan (diambil menggunakan grafik pengetahuan) dicapai melalui algoritma ontologi yang mengukur kesamaan semantik antara keterampilan kandidat dan keterampilan status target.
- Jalur dan konten pembelajaran — Komponen ini adalah mesin rekomendasi pembelajaran yang dapat merekomendasikan konten pembelajaran yang tepat dari katalog materi pembelajaran dari vendor mana pun berdasarkan kesenjangan keterampilan yang diidentifikasi. Mengidentifikasi jalur peningkatan keterampilan yang paling optimal untuk setiap kandidat dengan menganalisis kesenjangan keterampilan dan merekomendasikan konten pembelajaran yang diprioritaskan, untuk memungkinkan pengembangan profesional yang mulus dan berkelanjutan untuk setiap kandidat selama transisi ke peran baru.

Solusi otomatis berbasis cloud ini didukung oleh layanan pembelajaran mesin, grafik pengetahuan LLMs, dan Retrieval Augmented Generation (RAG). Ini dapat menskalakan untuk memproses puluhan atau ribuan resume dalam jumlah waktu minimum, membuat profil kandidat instan, mengidentifikasi kesenjangan dalam keadaan masa depan mereka saat ini atau potensial, dan kemudian secara efisien merekomendasikan konten pembelajaran yang tepat untuk menutup kesenjangan ini.

Gambar berikut menunjukkan end-to-end aliran kerangka kerja. Solusinya dibangun di atas fine-tuned di LLMs Amazon Bedrock. Ini LLMs mengambil data dari basis pengetahuan bakat perawatan

kesehatan di Amazon Neptunus. Algoritma berbasis data membuat rekomendasi untuk jalur pembelajaran yang optimal untuk setiap kandidat.



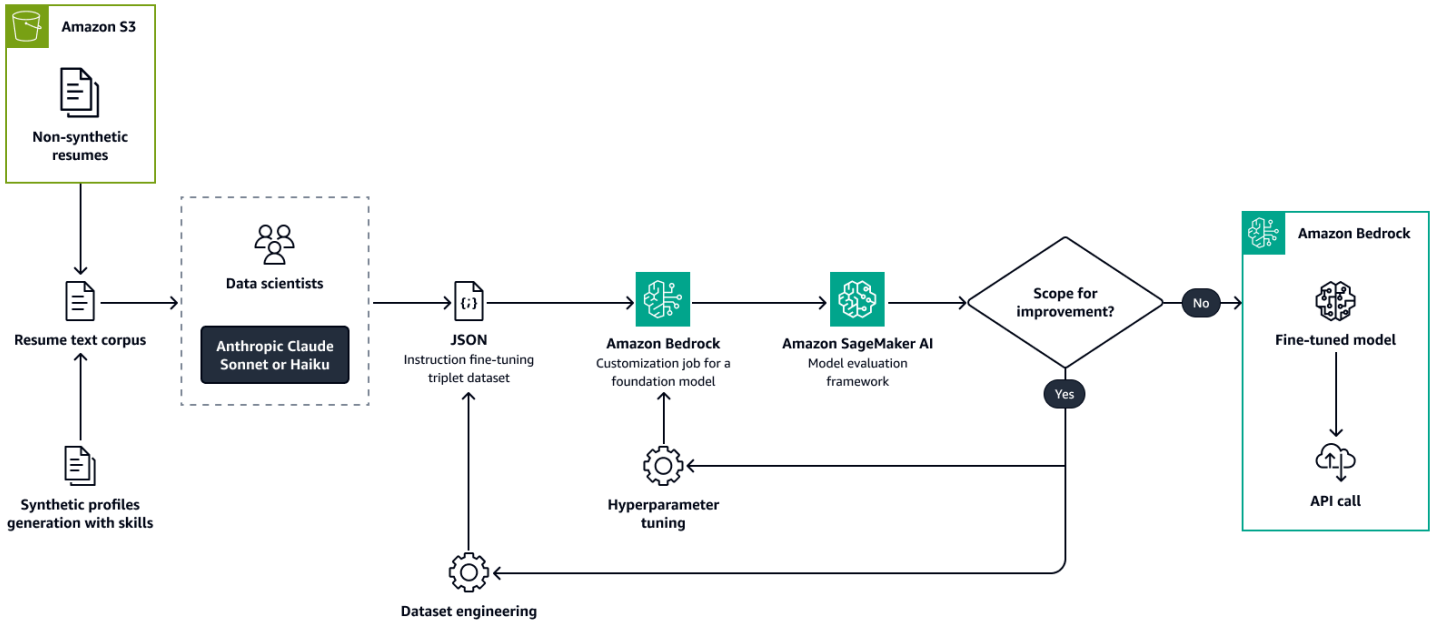

Membangun solusi ini terdiri dari langkah-langkah berikut:

- Langkah 1: Mengekstrak informasi bakat dan membangun profil keterampilan
- Langkah 2: Menemukan role-to-skill relevansi dari grafik pengetahuan
- Langkah 3: Mengidentifikasi kesenjangan keterampilan dan merekomendasikan pelatihan

# Langkah 1: Mengekstrak informasi bakat dan membangun profil keterampilan

Pertama, Anda menyempurnakan model bahasa besar, seperti Llama 2, di Amazon Bedrock dengan kumpulan data khusus. Ini menyesuaikan LLM untuk kasus penggunaan. Selama pelatihan, Anda secara akurat dan konsisten mengekstrak atribut bakat utama dari resume kandidat atau profil bakat serupa. Atribut bakat ini termasuk keterampilan, judul peran saat ini, judul pengalaman dengan rentang tanggal, pendidikan, dan sertifikasi. Untuk informasi selengkapnya, lihat Menyesuaikan model Anda untuk meningkatkan kinerjanya untuk kasus penggunaan Anda di dokumentasi Amazon Bedrock.

Gambar berikut menunjukkan proses untuk menyempurnakan model resume-parsing dengan menggunakan Amazon Bedrock. Resume nyata dan sintetis dibuat diteruskan ke LLM untuk mengekstrak informasi kunci. Sekelompok ilmuwan data memvalidasi informasi yang diekstraksi terhadap teks mentah asli. Informasi yang diekstraksi kemudian digabungkan dengan menggunakan chain-of-thoughtprompt dan teks asli untuk mendapatkan dataset pelatihan untuk fine-tuning. Dataset ini kemudian diteruskan ke pekerjaan kustomisasi Amazon Bedrock, yang menyempurnakan model. Pekerjaan batch Amazon SageMaker AI menjalankan kerangka evaluasi model yang mengevaluasi model yang disetel dengan baik. Jika model membutuhkan perbaikan, pekerjaan berjalan lagi dengan lebih banyak data atau hiperparameter yang berbeda. Setelah evaluasi memenuhi standar, Anda meng-host Model kustom melalui throughput yang disediakan Amazon Bedrock.



# Langkah 2: Menemukan role-to-skill relevansi dari grafik pengetahuan

Selanjutnya, Anda membangun grafik pengetahuan yang merangkum keterampilan dan peran taksonomi organisasi Anda dan organisasi lain di industri perawatan kesehatan. Basis pengetahuan yang diperkaya ini bersumber dari bakat gabungan dan data organisasi di Amazon Redshift. Anda dapat mengumpulkan data bakat dari berbagai penyedia data pasar tenaga kerja dan dari sumber data terstruktur dan tidak terstruktur khusus organisasi, seperti sistem perencanaan sumber daya perusahaan (ERP), sistem informasi sumber daya manusia (HRIS), resume karyawan, deskripsi pekerjaan, dan dokumen arsitektur bakat.

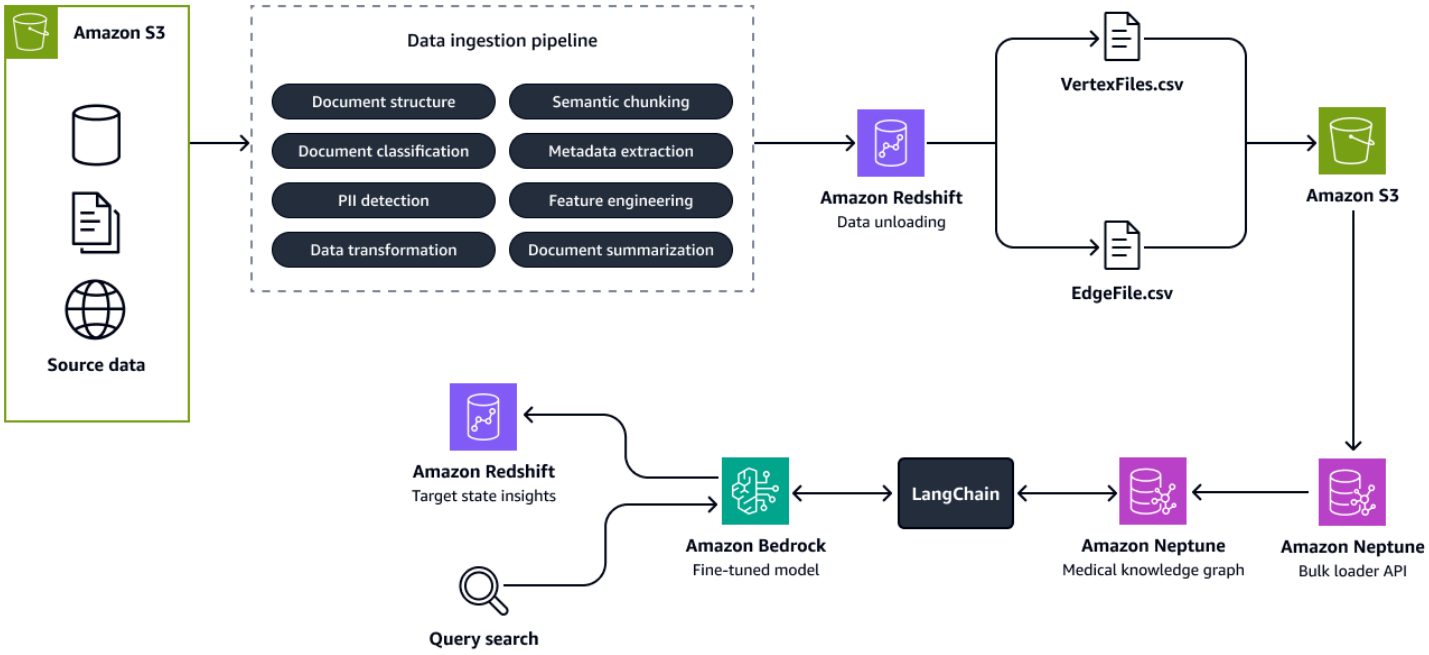
Bangun grafik pengetahuan di Amazon Neptunus. Node mewakili keterampilan dan peran, dan tepi mewakili hubungan di antara mereka. Perkaya grafik ini dengan metadata untuk menyertakan detail seperti nama organisasi, industri, keluarga pekerjaan, jenis keterampilan, tipe peran, dan tag industri.

Selanjutnya, Anda mengembangkan aplikasi Graph Retrieval Augmented Generation (Graph RAG). Graph RAG adalah pendekatan RAG yang mengambil data dari database grafik. Berikut ini adalah komponen aplikasi Graph RAG:

- Integrasi dengan LLM di Amazon Bedrock - Aplikasi ini menggunakan LLM di Amazon Bedrock untuk pemahaman bahasa alami dan pembuatan kueri. Pengguna dapat berinteraksi dengan sistem dengan menggunakan bahasa alami. Ini membuatnya dapat diakses oleh pemangku kepentingan non-teknis.
- Orkestrasi dan pengambilan informasi — Gunakan atau LlamaIndexLangChainorkestrator untuk memfasilitasi integrasi antara LLM dan grafik pengetahuan Neptunus. Mereka mengelola proses mengubah kueri bahasa alami menjadi kueri OpenCypher. Kemudian, mereka menjalankan kueri pada grafik pengetahuan. Gunakan teknik cepat untuk menginstruksikan LLM tentang praktik terbaik untuk membangun kueri OpenCypher. Ini membantu mengoptimalkan kueri untuk mengambil subgraf yang relevan, yang berisi semua entitas dan hubungan terkait tentang peran dan keterampilan yang ditanyakan.
- Pembuatan wawasan - LLM di Amazon Bedrock memproses data grafik yang diambil. Ini menghasilkan wawasan terperinci tentang keadaan saat ini dan memproyeksikan status masa depan untuk peran yang ditanyakan dan keterampilan terkait.

Gambar berikut menunjukkan langkah-langkah untuk membangun grafik pengetahuan dari sumber data. Anda meneruskan data sumber terstruktur dan tidak terstruktur ke pipeline konsumsi data.

Pipeline mengekstrak dan mengubah informasi menjadi formasi beban massal CSV yang kompatibel dengan Amazon Neptunus. API pemuat massal mengunggah file CSV yang disimpan dalam bucket Amazon S3 ke grafik pengetahuan Neptunus. Untuk kueri pengguna yang terkait dengan status talent future, peran yang relevan, atau keterampilan, LLM yang disetel dengan baik di Amazon Bedrock berinteraksi dengan grafik pengetahuan melalui a LangChain orkestrator. Orkestrator mengambil konteks yang relevan dari grafik pengetahuan dan mendorong respons ke tabel wawasan di Amazon Redshift. Bagian LangChain orkestrator, seperti Graph QAChain, mengonversi kueri bahasa alami dari pengguna ke kueri OpenCypher untuk menanyakan grafik pengetahuan. Model fine-tuned Amazon Bedrock menghasilkan respons berdasarkan konteks yang diambil.




## Langkah 3: Mengidentifikasi kesenjangan keterampilan dan merekomendasikan pelatihan

Pada langkah ini, Anda secara akurat menghitung kedekatan antara keadaan profesional kesehatan saat ini dan peran negara masa depan yang potensial. Untuk melakukan ini, Anda melakukan analisis afinitas keterampilan dengan membandingkan set keterampilan individu dengan peran pekerjaan. Dalam database vektor OpenSearch Layanan Amazon, Anda menyimpan informasi taksonomi keterampilan dan metadata keterampilan, seperti deskripsi keterampilan, jenis keterampilan, dan kelompok keterampilan. Gunakan model penyematan Amazon Bedrock, seperti model Amazon Titan Text Embeddings, untuk menyematkan keterampilan kunci yang diidentifikasi ke dalam vektor. Melalui pencarian vektor, Anda mengambil deskripsi keterampilan keadaan saat

ini dan keterampilan status target dan melakukan analisis ontologi. Analisis ini memberikan skor kedekatan antara pasangan keterampilan status saat ini dan target. Untuk setiap pasangan, Anda menggunakan skor ontologi yang dihitung untuk mengidentifikasi kesenjangan dalam afinitas keterampilan. Kemudian, Anda merekomendasikan jalur optimal untuk peningkatan keterampilan, yang dapat dipertimbangkan kandidat selama transisi peran.

Untuk setiap peran, merekomendasikan konten pembelajaran yang benar untuk meningkatkan keterampilan atau reskilling melibatkan pendekatan sistematis yang dimulai dengan membuat katalog konten pembelajaran yang komprehensif. Katalog ini, yang Anda simpan dalam database Amazon Redshift, menggabungkan konten dari berbagai penyedia dan menyertakan metadata, seperti durasi konten, tingkat kesulitan, dan mode pembelajaran. Langkah selanjutnya adalah mengekstrak keterampilan kunci yang ditawarkan oleh setiap konten dan kemudian memetakannya ke keterampilan individu yang diperlukan untuk peran target. Anda mencapai pemetaan ini dengan menganalisis cakupan yang disediakan oleh konten melalui analisis kedekatan keterampilan. Analisis ini menilai seberapa dekat keterampilan yang diajarkan oleh konten selaras dengan keterampilan yang diinginkan untuk peran tersebut. Metadata memainkan peran penting dalam memilih konten yang paling tepat untuk setiap keterampilan, memastikan bahwa peserta didik menerima rekomendasi khusus yang sesuai dengan kebutuhan belajar mereka. Gunakan LLMs di Amazon Bedrock untuk mengekstrak keterampilan dari metadata konten, melakukan rekayasa fitur, dan memvalidasi rekomendasi konten. Ini meningkatkan akurasi dan relevansi dalam proses peningkatan keterampilan atau reskilling.

## Penyelarasan dengan Kerangka AWS Well-Architected

Solusinya sejalan dengan keenam pilar dari AWS Well-Architected Framework:

- Keunggulan operasional - Pipa modular dan otomatis meningkatkan keunggulan operasional. Komponen utama dari pipa dipisahkan dan otomatis, memungkinkan pembaruan model yang lebih cepat dan pemantauan yang lebih mudah. Selain itu, jaringan pipa pelatihan otomatis mendukung rilis model yang disetel dengan lebih cepat.
- Keamanan — Solusi ini memproses informasi yang sensitif dan dapat diidentifikasi secara pribadi (PII), seperti data dalam resume dan profil bakat. Dalam AWS Identity and Access Management (IAM), terapkan kebijakan kontrol akses berbutir halus dan pastikan bahwa hanya personel yang berwenang yang memiliki akses ke data ini.
- Keandalan — Penggunaan solusi Layanan AWS, seperti Neptunus, Amazon Bedrock, dan Service OpenSearch , yang memberikan toleransi kesalahan, ketersediaan tinggi, dan akses tanpa gangguan ke wawasan bahkan selama permintaan tinggi.

- Efisiensi kinerja - Disesuaikan dalam basis data vektor LLMs Amazon Bedrock dan OpenSearch Service dirancang untuk memproses kumpulan data besar dengan cepat dan akurat untuk memberikan rekomendasi pembelajaran yang dipersonalisasi secara tepat waktu.
- Optimalisasi biaya — Solusi ini menggunakan pendekatan RAG, yang mengurangi kebutuhan akan pra-pelatihan model yang berkelanjutan. Alih-alih menyempurnakan seluruh model berulang kali, sistem hanya menyempurnakan proses tertentu, seperti mengekstraksi informasi dari resume dan menyusun output. Ini menghasilkan penghematan biaya yang signifikan. Dengan meminimalkan frekuensi dan skala pelatihan model intensif sumber daya dan dengan menggunakan layanan pay-per-use cloud, organisasi kesehatan dapat mengoptimalkan biaya operasional mereka sambil mempertahankan kinerja tinggi.
- Keberlanjutan — Solusi ini menggunakan layanan cloud-native yang dapat diskalakan yang mengalokasikan sumber daya komputasi secara dinamis. Ini mengurangi konsumsi energi dan dampak lingkungan sambil tetap mendukung inisiatif transformasi bakat berskala besar dan intensif data.

# Mengembangkan dan mengatur solusi AI generatif untuk perawatan kesehatan

Untuk membangun solusi dalam panduan ini, Anda harus membangun arsitektur RAG yang menggunakan fine-tuned LLMs untuk memberikan data pasien yang ditambah, wawasan klinis dan diagnostik, dan hasil pasien yang diprediksi kepada penyedia layanan kesehatan. Ini membutuhkan integrasi beberapa Layanan AWS dan alat untuk menciptakan alur kerja yang kohesif dan efisien. Bagian ini membahas hal-hal berikut:

- Amazon Q Developer— Gunakan Pengembang Amazon Q untuk menjawab pertanyaan teknik dan kesalahan kode selama proses pengembangan.
- Desain RAG multi-retriever— Merancang dan menerapkan solusi RAG yang menggunakan beberapa retriever untuk mengambil konteks medis yang benar untuk pertanyaan pengguna.
- ReAct agen— Menerapkan agen yang menggabungkan penalaran dengan tindakan dinamis.

## Amazon Q Developer

Saat membangun solusi AI generatif, mungkin sulit untuk membuat agen AI dan layanan kunci penghubung. Namun, Amazon Q Developer membantu ilmuwan data dan insinyur AI dengan menyediakan akses ke asisten AI generatif tingkat lanjut. Amazon Q dapat dengan cepat dan akurat menjawab pertanyaan pengguna dan kesalahan kode, yang dapat membantu Anda mengoptimalkan proses pengembangan LLM. Amazon Q menawarkan keuntungan signifikan bagi pengembang yang membuat aplikasi yang menggunakan model dasar Amazon Bedrock. Ini dapat merampingkan alur kerja dan meningkatkan kualitas kode. Ini mengotomatiskan pembuatan skrip Python dan infrastruktur sebagai konfigurasi kode (IAc), secara signifikan mengurangi waktu dan upaya pengembangan. Melalui kemampuan refactoring tingkat lanjut, Amazon Q dapat meningkatkan kinerja kode, mengidentifikasi kerentanan keamanan, dan memastikan pengembang mematuhi praktik terbaik. Selain itu, ini memfasilitasi pembelajaran dan adopsi untuk pemula dengan memberikan saran dan penjelasan yang sadar konteks, membuat tugas pengkodean yang kompleks lebih mudah diakses dan efisien.

# Desain RAG multi-retriever

Dalam aplikasi AI generatif, pipa RAG multi-retriever dapat secara efisien mengambil informasi dari berbagai sumber data untuk membantu penyedia layanan kesehatan dan dokter menjawab pertanyaan medis. Pipeline ini menggunakan berbagai jenis retriever untuk menarik data yang relevan dari basis pengetahuan yang berbeda. Setiap retriever mengkhususkan diri dalam mengambil jenis informasi tertentu, seperti riwayat pasien, wawasan diagnostik, catatan klinis, atau konten dari penelitian medis dan teks akademik.

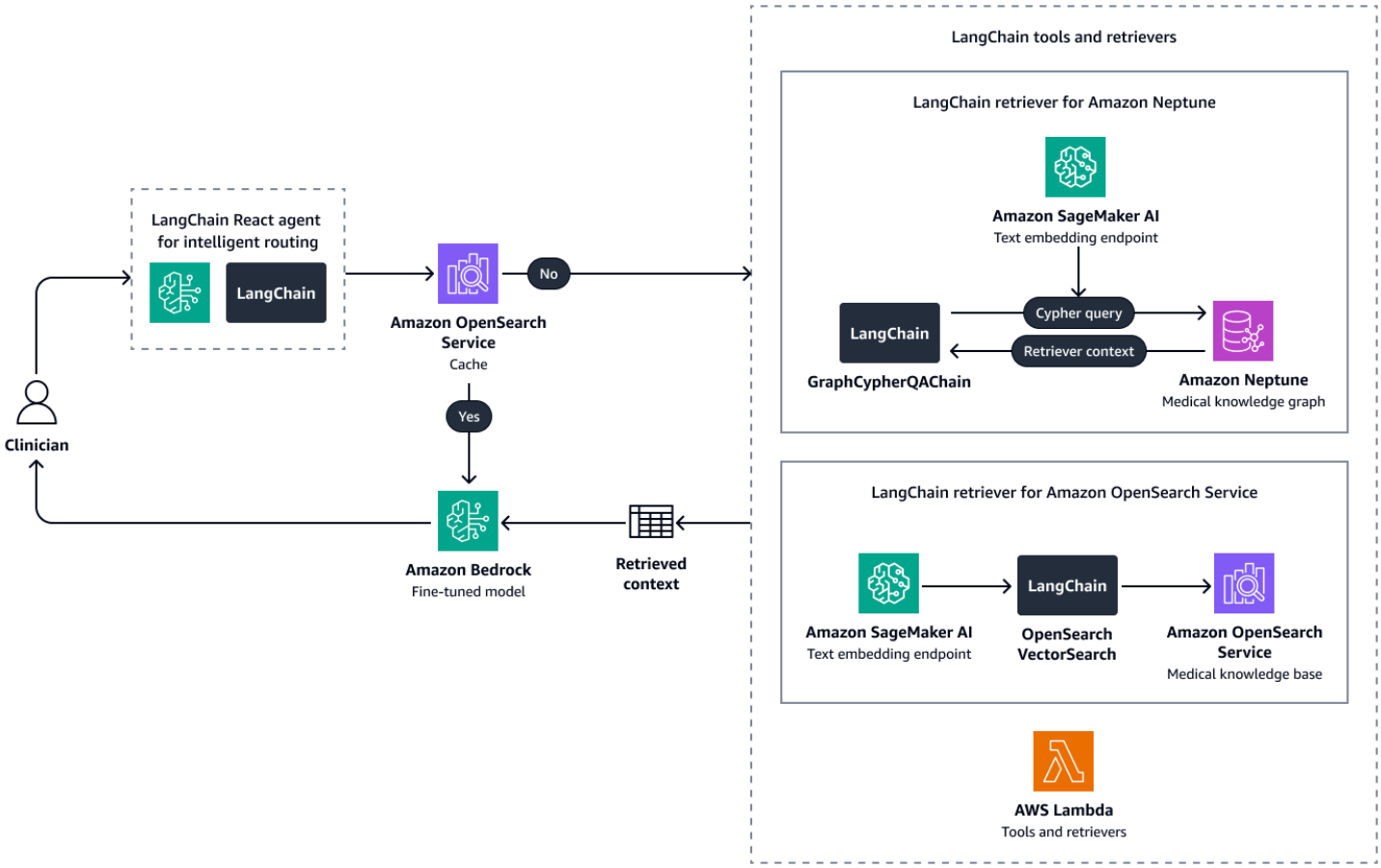
Gunakan sifat data dan persyaratan aplikasi khusus untuk menentukan basis pengetahuan backend yang benar yang benar untuk kasus penggunaan Anda. Basis data vektor OpenSearch Layanan Amazon sangat cocok untuk volume besar data perawatan kesehatan tidak terstruktur atau semi-terstruktur, termasuk ringkasan penilaian diagnosis gambar, ringkasan pelepasan, laporan klinis, penelitian medis, dan konten teks akademik. Di sisi lain, layanan database grafik, seperti Amazon Neptunus, dapat ideal untuk kasus penggunaan perawatan kesehatan yang memerlukan eksplorasi mendalam tentang hubungan temporal antara entitas, seperti pasien, riwayat pasien, penyedia layanan kesehatan, obat-obatan, gejala, dan perawatan.

Komponen penting dari pipeline ini adalah prediksi maksud kueri pengguna. Ini memastikan bahwa sistem merutekan kueri ke rantai retriever yang benar. Misalnya, jika seorang dokter bertanya tentang riwayat perawatan pasien, gejala, interaksi dengan rumah sakit, kemungkinan masuk kembali ke rumah sakit, atau hasil pasien potensial, maka modul prediksi maksud kueri mengidentifikasi maksud ini. Ini mengarahkan permintaan ke rantai retriever yang dapat mengambil catatan pasien atau data perawatan kronologis dari grafik pengetahuan medis. Atau, jika pertanyaannya adalah tentang penemuan penyakit, penilaian diagnostik spesifik, atau detail prosedur klinis spesifik dari buku teks akademik, maka kueri diarahkan ke rantai retriever yang dapat mengambil informasi ini dari database vektor Layanan. OpenSearch Anda dapat menggunakan fungsi pemanggilan alat LangChain untuk mengikat alat khusus ke Amazon Bedrock LLM yang dapat mengklasifikasikan pertanyaan pengguna ke dalam maksud yang telah ditentukan.

Sistem RAG multi-retriever ini mencakup LangChain agen yang dirancang untuk mengelola akses ke basis pengetahuan tertentu. Anda dapat menggunakan LangChain untuk mengatur interaksi antara Amazon Bedrock LLM, retriever yang berbeda, dan alat. LangChain menyertakan kelas pemanggilan alat yang membantu Anda membuat alat kustom, seperti pengklasifikasi maksud, retriever untuk Neptunus, retriever untuk OpenSearch Layanan, atau alat lain apa pun yang dapat dikembangkan untuk mengklasifikasikan maksud pengguna dan mengakses data dari basis pengetahuan tertentu dalam format terstruktur. Anda kemudian memasukkan alat-alat ini ke kelas

untuk membuat agen Penalaran dan Akting (ReAct). ReAct Agen memproses pertanyaan pengguna, merencanakan langkah-langkah berurutan untuk menjawab pertanyaan, dan kemudian secara berulang mengeksekusi alat yang tersedia dan memproses respons alat untuk akhirnya menjawab kueri pengguna.

Gambar berikut menunjukkan bagaimana sistem RAG multi-retriever yang dirancang untuk pengambilan pengetahuan yang efisien dan resolusi kueri cerdas bekerja. A LangChain ReAct agen menganalisis maksud pengguna, merumuskan rencana terstruktur untuk eksekusi, dan memilih alat pengambilan yang paling relevan. Sistem menanyakan cache pertanyaan sebelumnya dan memeriksa kueri serupa berdasarkan atribut kunci, seperti ID pasien, kondisi medis, dan tanggal kunjungan. Jika pertanyaan yang sangat mirip ditemukan, jawaban yang sesuai diambil secara langsung. Jika tidak, agen mengeksekusi retriever yang sesuai. Untuk mengambil informasi yang berpusat pada pasien, seperti riwayat pengobatan, gejala, interaksi rumah sakit, atau kemungkinan masuk kembali, sistem menggunakan grafik retriever. Untuk penilaian diagnostik, prosedur klinis, dan temuan medis terstruktur, agen menggunakan retriever database vektor. Dalam skenario yang membutuhkan kombinasi pengetahuan kontekstual dari kedua penyimpanan data, untuk menghasilkan respons yang komprehensif, sistem menggunakan strategi pengambilan hibrida yang mengintegrasikan hasil dari grafik pengetahuan dan basis data vektor.




# ReAct agen

Agen Reasoning and Acting (ReAct) dirancang untuk aplikasi RAG multi-segi. Agen ini memberikan kombinasi yang kuat antara penalaran dan tindakan dinamis, terutama untuk aplikasi kompleks yang melibatkan step-by-step, alur kerja pengambilan informasi logis. Untuk informasi lebih lanjut, lihat ReAct: Mensinergikan Penalaran dan Akting dalam Model Bahasa.

Dalam konteks medis dan perawatan kesehatan, pertanyaan dari dokter atau dokter seringkali memiliki banyak segi. Misalnya, seorang dokter mungkin bertanya “Perawatan apa yang diberikan kepada pasien serupa dengan hipertensi dan diabetes tipe 2?” Setelah mengidentifikasi maksud pengguna, yaitu untuk mengambil perawatan untuk hipertensi dan diabetes tipe 2, agen AI perlu membagi kueri ini menjadi subtugas dan kemudian memilih strategi pengambilan yang paling efisien. Dalam hal ini, agen AI harus mengidentifikasi node yang paling relevan (seperti usia pasien, jenis kelamin, kondisi, perawatan, dan obat-obatan) dan kemudian menanyakan grafik untuk entitas ini dan atribut serta hubungannya. ReAct agen sangat membantu karena mereka menggabungkan

kemampuan penalaran (inferensi logis) LLM dengan tindakan (query atau berinteraksi dengan sumber daya eksternal atau basis pengetahuan).

Untuk menjawab pertanyaan pengguna “Perawatan apa yang diberikan kepada pasien serupa dengan hipertensi dan diabetes tipe 2?” , contoh berikut menggambarkan bagaimana ReAct agen bekerja:

1. Penalaran agen — ReAct Agen menyimpulkan bahwa pertanyaan tersebut melibatkan pengambilan informasi tentang kondisi (diabetes dan hipertensi). Ini mempertimbangkan usia pasien, perawatan, obat-obatan, dan periode untuk menganalisis.
2. Tindakan agen — Agen menggunakan OpenCypher untuk menanyakan grafik pengetahuan untuk perawatan yang khusus untuk diabetes tipe 2 dan hipertensi. Ini juga mengambil obat yang diberikan, tanggal kunjungan rumah sakit, efek samping obat, hasil pasien yang diketahui, dan data referensi silang untuk pasien serupa (seperti pasien dengan jenis kelamin dan usia yang sama).
3. Observasi agen — Dari grafik pengetahuan, agen mengambil enam bulan terakhir data tabular tentang perawatan yang diberikan kepada pasien yang memiliki hipertensi dan diabetes tipe 2.
4. Penalaran agen — Untuk menentukan peringkat hasil dari catatan yang diambil, agen mengidentifikasi atribut penting, seperti kebaruan, efek samping obat, atau hasil pasien yang diketahui.
5. Tindakan agen — Agen memberi peringkat ulang catatan berdasarkan atribut yang diidentifikasi dan logika yang telah ditentukan sebelumnya yang diberikan melalui prompt sistem.
6. Generasi respons - LLM di Amazon Bedrock menghasilkan respons berdasarkan konteks yang disiapkan ReAct agen.

# Mengevaluasi solusi AI generatif untuk perawatan kesehatan

Mengevaluasi solusi AI perawatan kesehatan yang Anda bangun sangat penting untuk memastikan bahwa solusi tersebut efektif, andal, dan dapat diskalakan di lingkungan medis dunia nyata. Gunakan pendekatan sistematis untuk mengevaluasi kinerja setiap komponen solusi. Berikut ini adalah ringkasan metodologi dan metrik yang dapat Anda gunakan untuk mengevaluasi solusi Anda.

Topik



## Mengevaluasi ekstraksi informasi

Mengevaluasi kinerja solusi ekstraksi informasi, seperti parser resume cerdas dan ekstraktor entitas khusus. Anda dapat mengukur keselarasan respons solusi ini dengan menggunakan kumpulan data pengujian. Jika Anda tidak memiliki kumpulan data yang mencakup profil bakat perawatan kesehatan serbaguna dan catatan medis pasien, Anda dapat membuat kumpulan data tes khusus dengan menggunakan kemampuan penalaran LLM. Misalnya, Anda dapat menggunakan model parameter besar, seperti Anthropic Claude model, untuk menghasilkan dataset uji.

Berikut ini adalah tiga metrik utama yang dapat Anda gunakan untuk mengevaluasi model ekstraksi informasi:

- Akurasi dan kelengkapan — Metrik ini mengevaluasi sejauh mana output menangkap informasi yang benar dan lengkap yang ada dalam data kebenaran dasar. Ini melibatkan memeriksa kebenaran informasi yang diekstraksi dan keberadaan semua detail yang relevan dalam informasi yang diekstraksi.
- Kesamaan dan relevansi — Metrik ini menilai kesamaan semantik, struktural, dan kontekstual antara output dan data kebenaran dasar (kesamaan) dan sejauh mana output selaras dengan dan membahas konten, konteks, dan maksud dari data kebenaran dasar (relevansi).
- Tingkat penarikan atau penangkapan yang disesuaikan — Tingkat ini secara empiris menentukan berapa banyak nilai saat ini dalam data kebenaran dasar yang diidentifikasi dengan benar oleh model. Tarif harus mencakup hukuman untuk semua nilai palsu yang diekstrak model.

- Skor presisi — Skor presisi membantu Anda menentukan berapa banyak positif palsu yang ada dalam prediksi, dibandingkan dengan positif sebenarnya. Misalnya, Anda dapat menggunakan metrik presisi untuk mengukur kebenaran kemahiran keterampilan yang diekstraksi.

## Mengevaluasi solusi RAG dengan beberapa retriever

Untuk menilai seberapa baik sistem mengambil informasi yang relevan dan seberapa efektif menggunakan informasi tersebut untuk menghasilkan respons yang akurat dan sesuai kontekstual, Anda dapat menggunakan metrik berikut:

- Relevansi respons - Ukur seberapa relevan respons yang dihasilkan, yang menggunakan konteks yang diambil, dengan kueri asli.
- Ketepatan konteks — Dari total hasil yang diambil, evaluasi proporsi dokumen atau cuplikan yang diambil yang relevan dengan kueri. Ketepatan konteks yang lebih tinggi menunjukkan bahwa mekanisme pengambilan efektif dalam memilih informasi yang relevan.
- Kesetiaan — Menilai seberapa akurat respons yang dihasilkan mencerminkan informasi dalam konteks yang diambil. Dengan kata lain, ukur apakah respons tetap benar terhadap informasi sumber.

## Mengevaluasi solusi dengan menggunakan LLM

Anda dapat menggunakan teknik yang disebut LLM- as-a-judge untuk mengevaluasi respons teks dari solusi AI generatif Anda. Ini melibatkan penggunaan LLMs untuk mengevaluasi dan menilai kinerja output model. Teknik ini menggunakan kemampuan Amazon Bedrock untuk memberikan penilaian pada berbagai atribut, seperti kualitas respons, koherensi, kepatuhan, akurasi, dan kelengkapan preferensi manusia atau data kebenaran dasar. Anda menggunakan chain-of-thought (CoT) dan beberapa teknik bidikan untuk evaluasi komprehensif. Prompt menginstruksikan LLM untuk mengevaluasi respons yang dihasilkan dengan rubrik penilaian, dan sampel beberapa tembakan dalam prompt menunjukkan proses evaluasi yang sebenarnya. Prompt ini juga mencakup pedoman untuk diikuti oleh evaluator LLM. Misalnya, Anda dapat mempertimbangkan untuk menggunakan satu atau lebih teknik evaluasi berikut yang menggunakan LLM untuk menilai tanggapan yang dihasilkan:

- Perbandingan berpasangan - Berikan evaluator LLM pertanyaan medis dan beberapa tanggapan yang dihasilkan oleh versi berulang yang berbeda dari sistem RAG yang Anda buat. Minta

evaluator LLM untuk menentukan respons terbaik berdasarkan kualitas respons, koherensi, dan kepatuhan terhadap pertanyaan awal.

- Penilaian jawaban tunggal — Teknik ini sangat cocok untuk kasus penggunaan di mana Anda perlu mengevaluasi keakuratan kategorisasi, seperti klasifikasi hasil pasien, kategorisasi perilaku pasien, kemungkinan masuk kembali pasien, dan kategorisasi risiko. Gunakan evaluator LLM untuk menganalisis kategorisasi atau klasifikasi individu secara terpisah, dan mengevaluasi alasan yang diberikannya terhadap data kebenaran dasar.
- Penilaian yang dipandu referensi - Berikan evaluator LLM dengan serangkaian pertanyaan medis yang memerlukan jawaban deskriptif. Buat contoh tanggapan untuk pertanyaan-pertanyaan ini, seperti jawaban referensi atau tanggapan ideal. Minta evaluator LLM untuk membandingkan respons yang dihasilkan LLM dengan jawaban referensi atau tanggapan ideal, dan minta evaluator LLM untuk menilai respons yang dihasilkan untuk akurasi, kelengkapan, kesamaan, relevansi, atau atribut lainnya. Teknik ini membantu Anda mengevaluasi apakah respons yang dihasilkan selaras dengan standar yang terdefinisi dengan baik atau jawaban teladan.

# Sumber daya

## AWS dokumentasi

- Dokumentasi Amazon Bedrock
- Dokumentasi Amazon Neptunus
- Dokumentasi OpenSearch Layanan Amazon
- Menerapkan Kerangka AWS Kerja Well-Architected untuk Amazon Neptunus (Panduan Preskriptif)AWS
- Praktik terbaik operasional untuk OpenSearch Layanan Amazon (Dokumentasi OpenSearch layanan)
- Menggunakan Amazon Comprehend Medical LLMs dan untuk perawatan kesehatan dan ilmu kehidupan (Prescriptive Guidance)AWS

## AWS posting blog

- Buat RAG dan aplikasi AI generatif berbasis agen dengan model Amazon Titan Text Premier baru, tersedia di Amazon Bedrock
- Lengkapi Kecerdasan Komersil dengan Membangun Grafik Pengetahuan dari Gudang Data dengan Amazon Neptunus
- Menggunakan grafik pengetahuan untuk membangun aplikasi GraphRag dengan Amazon Bedrock dan Amazon Neptunus

## Sumber daya lainnya

- Mengintegrasikan Generasi Retrieval-Augmented dengan Model Bahasa Besar dalam Nefrologi: Memajukan Aplikasi Praktis (Pusat, Perpustakaan Kedokteran Nasional) PubMed
- Pengantar LangChain (LangChain dokumentasi)

# Kontributor

## Mengotorisasi

- Nitu Nivedita, Direktur Pelaksana — Pimpinan Kecerdasan Buatan, Data & AI, Accenture
- Manoj Appully, Pendiri dan CTO, Cadiem
- Conor Folan, Konsultan — Data & AI, Accenture
- Deepak Krishna AR, Konsultan — Data & AI, Accenture
- Almore Cato, Manajer — Data & AI, Accenture
- Soonam Kurian, Arsitek Solusi Utama, AWS

## Meninjau

- Sally Lin, Manajer Senior Ilmu Data — Data & AI, Accenture
- Terry Huang, Manajer Ilmu Data — Data & AI, Accenture
- William Lorenz, Arsitek Solusi Mitra, AWS

## Penulisan teknis

- Lilly AbouHarb, Penulis Teknis Senior, AWS

# Riwayat dokumen

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

| Perubahan | Deskripsi | Tanggal |
| --- | --- | --- |
| Publikasi awal | — | Maret 14, 2025 |

# AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

## Nomor

7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instance EC2 di. AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

# A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

Metode migrasi database di mana database sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi aktif-pasif.

migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi `SUM` dan`MAX`.

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk penemuan portofolio dan proses analisis dan membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat Apa itu Kecerdasan Buatan?

operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat panduan integrasi operasi.

enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat ABAC untuk AWS dokumentasi AWS Identity and Access Management (IAM).

sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan memproses atau memodifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat situs web AWS CAF dan whitepaper AWS CAF.

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool ()AWS SCT. Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

# B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat Data dalam grafik perilaku di dokumentasi Detektif.

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga endianness.

klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

deployment biru/hijau

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

botnet

Jaringan bot yang terinfeksi oleh malware dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat Tentang cabang (GitHub dokumentasi).

akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator Implementasikan prosedur break-glass dalam panduan Well-Architected AWS .

strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian Terorganisir di sekitar kemampuan bisnis dari Menjalankan layanan mikro kontainer di whitepaper. AWS

perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

# C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

rekayasa kekacauan

Dengan sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan AWS Fault Injection Service (AWS FIS) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat posting CCo E di Blog Strategi AWS Cloud Perusahaan.

komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi edge computing.

model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat Membangun Model Operasi Cloud Anda.

tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCo E, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog The Journey Toward Cloud-First & the Stages of Adoption di blog Strategi Perusahaan. AWS Cloud Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat panduan kesiapan migrasi.

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Region, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat Paket kesesuaian dalam dokumentasi. AWS Config

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD is commonly described as a pipeline. CI/CDdapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

# D

data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisan dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat Klasifikasi data.

penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat Membangun perimeter data pada AWS.

prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

DDL

Lihat bahasa definisi database.

ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

defense-in-depth

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, defense-in-depth pendekatan mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat Layanan yang berfungsi dengan AWS Organizations AWS Organizations dokumentasi.

deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

tabel dimensi

Dalam skema bintang, tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh bencana. Untuk informasi selengkapnya, lihat Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk mendeteksi penyimpangan dalam sumber daya sistem, atau Anda dapat menggunakannya AWS Control Tower untuk mendeteksi perubahan di landing zone yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

DVSM

Lihat pemetaan aliran nilai pengembangan.

# E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan komputasi awan, komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat Apa itu Pertukaran Data Elektronik.

enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat layanan titik akhir di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, MES, dan manajemen proyek) untuk suatu perusahaan.

enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat Enkripsi amplop dalam dokumentasi AWS Key Management Service (AWS KMS).

lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam pipa CI/CD, lingkungan produksi adalah lingkungan penyebaran terakhir.

- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat panduan implementasi program.

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

# F

tabel fakta

Tabel tengah dalam skema bintang. Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat Batas Isolasi AWS Kesalahan.

cabang fitur

Lihat cabang.

fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal “2021-05-27 00:15:37" menjadi “2021", “Mei”, “Kamis”, dan “15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

beberapa tembakan mendorong

Menyediakan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga bidikan nol.

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui pengambilan data perubahan untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih

menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat model pondasi.

model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FMs mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat Apa itu Model Foundation.

# G

AI generatif

Subset model AI yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat Apa itu AI Generatif.

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat Membatasi distribusi geografis konten Anda dalam dokumentasi. CloudFront

Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan alur kerja berbasis batang adalah pendekatan modern yang lebih disukai.

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur,

gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi ()OUs. Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

# H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. AWS menyediakan AWS SCT yang membantu dengan konversi skema.

ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran mesin. Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

## I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah. Untuk informasi selengkapnya, lihat praktik terbaik Deploy using immutable infrastructure di AWS Well-Architected Framework.

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

Industri 4.0

Sebuah istilah yang diperkenalkan oleh Klaus Schwab pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML.

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAc dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi lebih lanjut, lihat Membangun strategi transformasi digital Internet of Things (IIoT) industri.

inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPCs (dalam yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat Apa itu IoT?

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan integrasi operasi.

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

# L

kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan.

model bahasa besar (LLM)

Model AI pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke dalam bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat Apa itu LLMs.

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga endianness.

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

# M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

PETA

Lihat Program Percepatan Migrasi.

mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi lebih lanjut, lihat Membangun mekanisme di AWS Well-Architected Framework.

akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

Protokol komunikasi ringan machine-to-machine (M2M), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi dengan jelas APIs dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan ringan. APIs Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat Menerapkan layanan mikro di AWS.

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik dan pelajaran terbaik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari strategi AWS migrasi.

pabrik migrasi

Tim lintas fungsi yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat diskusi tentang pabrik migrasi dan panduan Pabrik Migrasi Cloud di kumpulan konten ini.

metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 dengan Layanan Migrasi AWS Aplikasi.

Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat panduan kesiapan migrasi. MRA adalah tahap pertama dari strategi AWS migrasi.

strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke file. AWS Cloud Untuk informasi lebih lanjut, lihat entri 7 Rs di glosarium ini dan lihat Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar.

ML

Lihat pembelajaran mesin.

modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat Strategi untuk memodernisasi aplikasi di. AWS Cloud

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat Mengevaluasi kesiapan modernisasi untuk aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat Menguraikan monolit menjadi layanan mikro.

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya “Apakah produk ini buku, mobil, atau telepon?” atau “Kategori produk mana yang paling menarik bagi pelanggan ini?”

infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan infrastruktur yang tidak dapat diubah sebagai praktik terbaik.

# O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi lebih lanjut, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja Well-Architected AWS .

teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi Industri 4.0.

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat panduan integrasi operasi.

jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat Membuat jejak untuk organisasi dalam CloudTrail dokumentasi.

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis dan permintaan ke bucket S3. `PUT` `DELETE`

identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga OAC, yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

# P

batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat Batas izin dalam dokumentasi IAM.

Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakan berbasis sumber daya), atau menentukan izin maksimum untuk semua akun di organisasi (lihat kebijakan kontrol layanan). AWS Organizations

ketekunan poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka. Untuk informasi selengkapnya, lihat Mengaktifkan persistensi data di layanan mikro.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

Kondisi kueri yang mengembalikan `true` atau`false`, biasanya terletak di `WHERE` klausa.

predikat pushdown

Teknik optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat Kontrol pencegahan dalam Menerapkan kontrol keamanan pada. AWS

principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam istilah dan konsep Peran dalam dokumentasi IAM.

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona yang dihosting pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau lebih. VPCs Untuk informasi selengkapnya, lihat Bekerja dengan zona yang dihosting pribadi di dokumentasi Route 53.

kontrol proaktif

Kontrol keamanan yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat panduan referensi Kontrol dalam AWS Control Tower dokumentasi dan lihat Kontrol proaktif dalam Menerapkan kontrol keamanan pada AWS.

manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

rantai cepat

Menggunakan output dari satu prompt LLM sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

pseudonimisasi

Proses penggantian pengenal pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

publish/subscribe (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam MES berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

# Q

rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

# R

Matriks RACI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

LAP

Lihat Retrieval Augmented Generation.

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

refactor

Lihat 7 Rs.

Wilayah

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah "Berapa harga rumah ini akan dijual?" Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. Ketersediaan tinggi dan pemulihan bencana adalah pertimbangan umum ketika merencanakan ketahanan di. AWS Cloud Untuk informasi lebih lanjut, lihat AWS Cloud Ketahanan.

kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang

didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Tipe dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat Kontrol responsif dalam Menerapkan kontrol keamanan pada AWS.

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

Teknologi AI generatif di mana LLM merujuk sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat Apa itu RAG.

rotasi

Proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensil.

kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

# D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke AWS Management Console atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensil pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat Apa yang ada di rahasia Secrets Manager? dalam dokumentasi Secrets Manager.

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.

pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans EC2 Amazon, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCPs menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCPs daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti uptime dan kinerja layanan.

indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator tingkat layanan.

model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama.

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan

mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat Pendekatan bertahap untuk memodernisasi aplikasi di. AWS Cloud

SPOF

Lihat satu titik kegagalan.

skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam gudang data atau untuk tujuan intelijen bisnis.

pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini diperkenalkan oleh Martin Fowler sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan Amazon CloudWatch Synthetics untuk membuat tes ini.

sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke LLM untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

# T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda dapat membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai AWS sumber daya Anda.

variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. Untuk informasi selengkapnya, lihat Apa itu gateway transit dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Organizations dengan AWS layanan lain dalam AWS Organizations dokumentasi.

penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

# U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. Untuk informasi lebih lanjut, lihat panduan Mengukur ketidakpastian dalam sistem pembelajaran mendalam.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau

memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat lingkungan.

# V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPCs yang memungkinkan Anda untuk merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Apa itu peering VPC di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

# W

cache hangat

Cache buffer yang berisi data saat ini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap tidak dapat diubah.

# Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan zero-day.

kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

bisikan zero-shot

Memberikan LLM dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa bidikan yang diminta.

aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

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