Mengoperasionalkan AI agen pada AWS

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Mengoperasionalkan AI agen pada AWS

Table of Contents

Mengoperasionalkan AI agen pada AWS

Aaron Sempf, Brad Ryan, Bhargs Srivathsan, dan Akhil Bhaskar, Amazon Web Services

Agustus 2025 (sejarah dokumen)

Agentic AI bukanlah fitur — ini adalah paradigma operasional baru. Organizations yang berinvestasi dalam arsitektur disiplin, kerangka kerja kepercayaan, dan model penyebaran yang selaras dengan bisnis akan memimpin generasi berikutnya dari perusahaan yang adaptif dan cerdas.

Agentic AI mewakili konvergensi agen perangkat lunak otonom dan AI generatif. Ini memadukan pengambilan keputusan dan perilaku agen yang diarahkan pada tujuan dengan pemahaman bahasa dan kemampuan generasi model bahasa besar (). LLMs Agen ini dapat bernalar, bertindak, beradaptasi, dan berkolaborasi di seluruh lingkungan perusahaan yang dinamis. Untuk mengoperasionalkan potensi ini, perusahaan harus mengubah pola pikir mereka dari penerapan model ke infrastruktur agen.

Panduan ini memberikan strategi organisasi untuk mengubah AI agen dari eksperimen terisolasi menjadi infrastruktur skala perusahaan yang menghasilkan nilai. Ini dapat membantu Anda menanamkan agen cerdas di seluruh alur kerja dengan tata kelola, skalabilitas, dan penyelarasan bisnis.

Area fokus utama dan rekomendasi

Panduan ini berfokus pada area dasar berikut saat mengoperasionalkan AI agen. Rekomendasi organisasi dan bisnis disediakan untuk setiap area fokus:

- Area fokus 1: Klarifikasi maksud dan ruang lingkup agen— Sejajarkan agen dengan prioritas bisnis dan kemacetan kognitif. Perlakukan agen sebagai rekan tim digital, bukan hanya sebagai alat.
- Area fokus 2: Desain untuk komposisi dan kolaborasi— Merangkul sistem multi-agen dengan arsitektur modular, protokol semantik, dan delegasi dinamis melalui agen arbiter.
- Area fokus 3: Arsitek untuk multi-tenancy dan kontrol— Bangun infrastruktur yang dapat diskalakan dan sadar penyewa dengan layanan agen bersama, tata kelola terpusat, dan akses berbasis peran.
- Area fokus 4: Membangun kepercayaan melalui identitas, pagar pembatas, dan observabilitas— Menegakkan ketertelusuran, kontrol runtime, dan penjelasan untuk mendapatkan kepercayaan pemangku kepentingan.

- Area fokus 5: Kelola siklus hidup— Membangun pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD), pembuatan versi cepat, telemetri, dan pelatihan ulang berkelanjutan untuk mendukung kinerja dan efisiensi AI agen.
- Area fokus 6: Sejajarkan model agen dengan model bisnis— Monetisasi kemampuan agen melalui model berbasis penggunaan, metrik ROI internal, dan penawaran komersial.

Anda dapat menggunakan rekomendasi dalam panduan ini untuk mempersiapkan bisnis Anda untuk AI agen dalam skala besar. Ini menguraikan bagaimana organisasi harus merestrukturisasi di sekitar AI agen, termasuk membangun DevOps tim agen (AgentOps), sistem yang dapat dioperasikan, dan mengubah strategi manajemen yang meningkatkan adopsi. Ini menekankan pemikiran keputusan-pertama dan penyelarasan dengan Kerangka Well-Architected. AWS

Audiens yang dituju

Panduan ini ditujukan untuk arsitek perusahaan, pemimpin AI/ML teknik, dan ahli strategi transformasi digital yang merancang dan menskalakan sistem agen, menanamkan AI ke dalam alur kerja bisnis inti, dan LLMs mengoperasionalkan dan agen otonom di lingkungan produksi. Untuk memahami konsep dan rekomendasi dalam panduan ini, Anda harus terbiasa dengan arsitektur cloud-native modern dan sistem terdistribusi, model bahasa besar, kemampuan model dasar, dan prinsip-prinsip tata kelola AI, dan rekayasa platform DevOps.

Tujuan

Dengan menerapkan rekomendasi dalam panduan ini, organisasi Anda dapat mencapai hasil bisnis berikut:

- Pengambilan keputusan yang dipercepat dan pelaksanaan alur kerja melalui agen otonom yang berorientasi pada tujuan yang mengurangi kemacetan manusia dan beban kognitif.
- Penerapan kemampuan cerdas yang dapat diskalakan dan hemat biaya di seluruh unit bisnis, melalui platform agen multi-penyewa yang dapat digunakan kembali.
- Ketahanan, kepercayaan, dan tata kelola yang lebih besar dalam sistem AI, yang memungkinkan adopsi yang percaya diri dalam lingkungan yang diatur, kritis, atau menghadapi pelanggan.

Tentang seri konten ini

Panduan ini adalah bagian dari seri tentang AI agen di AWS. Untuk informasi lebih lanjut dan untuk melihat panduan lain dalam seri ini, lihat Agentic AI di situs web AWS Prescriptive Guidance.

Fondasi strategis untuk AI agen

Sistem agen bukanlah hal baru. Agen perangkat lunak, termasuk otomatisasi proses robot (RPA) dan mesin keputusan, telah ada selama beberapa dekade. Tetapi mereka sederhana dan deterministik, dirancang untuk mengikuti aturan yang telah ditentukan dan logika simbolik untuk menjalankan tugas berulang dan variasi rendah. Dengan munculnya AI generatif, permainan telah berubah. Model bahasa besar (LLMs) sekarang dapat menafsirkan masukan yang kompleks, menghasilkan respons secara dinamis, dan dengan cepat mensintesis pengetahuan. Anda sekarang dapat menskalakan agensi tanpa logika rapuh atau hard-code. Sekarang, agen dapat bernalar, membuat keputusan, memanggil alat, beradaptasi dengan konteks, dan berkoordinasi dengan agen lain di seluruh alur kerja. Mereka dapat beroperasi secara mandiri menuju tujuan, mempertahankan memori, dan merefleksikan hasil.

Namun, kemampuan mentah tidak cukup. Kecerdasan tanpa integrasi menghasilkan hal baru, bukan dampak. Untuk membuka nilai dari yang kuat LLMs, perusahaan harus beralih melampaui eksperimen terisolasi ke ekosistem yang direkayasa. Agen harus diperlakukan sebagai layanan tingkat produksi yang beroperasi di bawah disiplin yang sama seperti sistem perusahaan mana pun. Itu termasuk tata kelola, observabilitas, model identitas aman, dan manajemen siklus hidup. Mereka juga harus menghasilkan hasil bisnis yang nyata, bukan potensi spekulatif. Sistem ini harus dirancang dengan batas-batas yang jelas untuk pengambilan keputusan dan toleransi kesalahan. Penting untuk menggabungkan mekanisme pemulihan otomatis, pemantauan kinerja waktu nyata, dan manajemen sumber daya yang dapat diskalakan. Ini membantu Anda menangani sifat interaksi agen yang dinamis dan non-deterministik sambil mempertahankan tingkat layanan yang konsisten di seluruh alur kerja perusahaan.

Pada tingkat dasar, perusahaan harus memikirkan kembali bagaimana kecerdasan tertanam ke dalam struktur operasi. Agen harus dirancang untuk berintegrasi dengan sistem inti, mematuhi kebijakan perusahaan, dan memberikan nilai yang terukur. Mereka perlu beroperasi dalam skala besar, lintas departemen, domain, dan konteks pengguna. Mengoperasionalkan AI agen pada akhirnya adalah tentang penggunaan; itu adalah perbedaan antara menyebarkan AI yang melakukan tugas terisolasi dan menyebarkan agen yang mengembangkan model bisnis Anda.

Agentic AI mewakili filosofi operasi baru yang membutuhkan perubahan mendasar dalam cara kami mendekati sistem, proses, dan orang untuk meningkatkan kecerdasan di seluruh organisasi. Agen menjadi aset strategis yang memperkuat kemampuan manusia. Dengan mengintegrasikan AI agen ke dalam operasi mereka, organisasi dapat membuka wawasan yang mendorong nilai bisnis, meningkatkan kemampuan manusia, dan mengoptimalkan alur kerja yang kompleks.

Area fokus strategis untuk AI agen

Untuk beralih dari prototipe awal ke tingkat produksi dan sistem penghasil nilai, tim memerlukan strategi yang koheren yang memadukan arsitektur, proses, dan pemikiran produk.

Banyak organisasi masih mendekati AI dengan pola pikir yang mengutamakan alat atau model-sentris. AI generatif telah memperkuat eksperimen, tetapi seringkali tanpa keselarasan yang jelas dengan strategi bisnis atau hasil yang terukur. Tanpa peran strategis yang ditentukan, agen berisiko menjadi eksperimen baru yang menguras sumber daya daripada memberikan nilai yang dapat diskalakan. Untuk menetapkan peran strategis AI agen, organisasi harus memulai dengan prioritas bisnis. Identifikasi area kelebihan kognitif, kemacetan keputusan, atau alur kerja yang terfragmentasi di mana otonomi dapat memberikan bantuan. Gunakan pernyataan masalah khusus domain untuk membentuk tanggung jawab agen. Perlakukan agen sebagai rekan tim digital — bukan alat — yang dapat bernalar, mendelegasikan, dan beradaptasi.

Ilmu keputusan adalah disiplin menggabungkan ilmu data, analitik, dan pemodelan perilaku untuk meningkatkan pengambilan keputusan. Ini harus diintegrasikan di awal proses arsitektur agen untuk menyelaraskan desain dengan hasil bisnis. Dengan mengidentifikasi pola keputusan, mensimulasikan trade-off, dan mengukur dampak nilai, ilmu keputusan dapat membantu Anda menentukan di mana otonomi agen dapat memberikan nilai tertinggi. Ilmu keputusan dapat mempercepat keputusan, mengurangi kesalahan, dan memungkinkan adaptasi waktu nyata. Yayasan berdasarkan informasi data ini mendasarkan desain agen dalam wawasan yang terukur, dan memungkinkan integrasi yang lebih ketat dengan teknologi perusahaan yang ada, seperti mesin aturan, platform analitik, dan model prediktif.

Untuk membantu menetapkan peran strategis agen, bagian ini memperkenalkan area fokus dasar yang membentuk tulang punggung operasionalisasi AI agen. Setiap peta ke pekerjaan inti yang harus dilakukan dari perspektif pemimpin teknis, arsitek, atau pemilik produk yang bertanggung jawab atas bagaimana agen disusun dan dirancang. Area fokus ini bukan langkah berurutan. Masing-masing layak ditinjau kembali di seluruh siklus hidup sistem untuk menumbuhkan ekosistem agen yang tangguh, terukur, dan dapat dimonetisasi.

Bagian ini berisi area fokus berikut:

- Area fokus 1: Klarifikasi maksud dan ruang lingkup agen
- Area fokus 2: Desain untuk komposisi dan kolaborasi
- Area fokus 3: Arsitek untuk multi-tenancy dan kontrol
- Area fokus 4: Membangun kepercayaan melalui identitas, pagar pembatas, dan observabilitas

Area fokus 1: Klarifikasi maksud dan ruang lingkup agen

Job to be done: “Bantu saya memastikan bahwa setiap agen memecahkan masalah nyata dengan batasan yang jelas, bukan hanya demo keren.”

Agentic AI bukan hanya tentang membangun kemampuan. Ini tentang memecahkan masalah yang tepat, dengan cara yang benar, untuk hasil yang benar. Itu dimulai dengan menjadi sangat jelas tentang maksud dari solusi AI agen.

Strategi

Terlalu sering, organisasi memulai dengan apa yang dapat dilakukan model (seperti menelepon APIs, menjawab pertanyaan, atau menghasilkan ringkasan) dan memperbaiki kasus penggunaan di sekitarnya. Hal ini menyebabkan cakupan creep, integrasi yang buruk, dan agen yang secara teknis mengesankan tetapi secara operasional tidak berguna. Sebagai gantinya, mulailah dengan mendefinisikan peran agen melalui pertanyaan spesifik seperti berikut:

- Hasil spesifik apa yang menjadi tanggung jawab agen?
- Siapa yang bertindak atas nama?
- Siapa yang diuntungkan?
- Di mana otonomi agen dimulai dan berakhir?
- Apa yang terjadi ketika gagal?

Agen yang tercakup dengan baik memiliki pekerjaan yang jelas, tanggung jawab yang jelas, dan kriteria keberhasilan yang terukur. Jangan menganggap agen sebagai asisten atau chatbot. Sebaliknya, berikan jabatan. Anggap saja sebagai agen sukses pelanggan, penangan pengembalian produk, atau monitor kepatuhan.

Saat melibatkan pemangku kepentingan atau pelanggan, tekankan skalabilitas dan kemampuan beradaptasi sistem AI agen. Agen-agen ini berkembang dengan bisnis, terus meningkat melalui pembelajaran dan umpan balik. Untuk mengurangi resistensi dan mempercepat adopsi, soroti bagaimana alat agen dirancang dengan mempertimbangkan empati pekerja. Mereka memberikan transparansi, kontrol, dan mekanisme penggantian opsional yang membangun kepercayaan. Alih-alih

mengganti orang, agen meningkatkan kemampuan manusia dan pengambilan keputusan, membantu karyawan untuk tetap berada dalam lingkaran dan fokus pada tugas-tugas bernilai tinggi.

Kunci keberhasilan implementasi adalah menyelaraskan AI agen dengan hasil bisnis yang spesifik dan berdampak tinggi. Dorong tim dan mitra untuk memulai dengan proyek percontohan terfokus yang memecahkan titik nyeri yang terlihat. Kemenangan cepat menghasilkan laba atas investasi (ROI) yang terukur, membangun pembelian internal, dan menciptakan momentum untuk adopsi yang lebih luas.

Untuk memandu adopsi dan kedewasaan, organisasi dapat membingkai desain agen di sepanjang model evolusioner. Otonomi agen, kompleksitas, dan dampak bisnis semakin meningkat. Berikut ini adalah tahapan model ini:

- Agen pengamat memunculkan wawasan dari kebisingan. Contohnya adalah agen sentimen pasar yang melacak persepsi merek di seluruh saluran digital.
- Asisten agen mendukung pengambilan keputusan manusia. Contohnya adalah agen penasihat kesepakatan yang mensintesis data pesaing dan kondisi pasar untuk tim penjualan.
- Agen otonom bertindak secara independen dalam batas-batas yang ditentukan. Contohnya adalah agen alokasi sumber daya yang secara dinamis menyesuaikan infrastruktur cloud berdasarkan permintaan.
- Agen orkestrator mengoordinasikan alur kerja multi-agen. Contohnya adalah agen pengoptimalan rantai pasokan yang mengelola interaksi antara inventaris, logistik, dan agen peramalan.
- Agen inovator menghasilkan kemungkinan strategis baru. Contohnya adalah agen inovasi model bisnis yang menganalisis tren pasar dan merekomendasikan aliran pendapatan baru.

Membingkai agen di sekitar hasil strategis dan tingkat kematangan ini meningkatkan fokus, mempercepat adopsi, dan membangun kepercayaan pemangku kepentingan.

Untuk mendukung penyelarasan di area fokus ini Layanan AWS, seperti Amazon Quick, dapat memvisualisasikan indikator kinerja utama (KPIs) yang terkait dengan hasil yang digerakkan oleh agen. Anda dapat menggunakan Amazon CloudWatch untuk memantau perilaku agen, metrik kinerja, dan kesehatan sistem dalam waktu dekat. Gunakan umpan balik operasional untuk menyetel interaksi agen dan penggunaan sumber daya. AWS CloudTraildapat memberikan visibilitas ke dalam aktivitas agen dan pola integrasi selama fase eksperimen dan penyempurnaan awal.

Nilai bisnis dari mendefinisikan maksud dan ruang lingkup

Adopsi AI agen merupakan perubahan penting dalam cara organisasi mendekati transformasi digital dan keunggulan operasional. Ini bukan hanya tentang otomatisasi. Ini tentang memungkinkan otonomi cerdas yang mempercepat pengambilan keputusan dan realisasi nilai.

Penggerak bisnis utama meliputi yang berikut:

- Keunggulan kompetitif — Pengadopsi awal memperoleh keunggulan strategis melalui wawasan yang lebih cepat, layanan yang lebih baik, dan operasi adaptif.
- Peningkatan pengalaman pelanggan — Agen menawarkan dukungan real-time, personal, dan selalu aktif yang meningkatkan kepuasan dan loyalitas.
- Efisiensi operasional — Agentic AI secara signifikan mengurangi beban kognitif manusia dengan mengotomatiskan tugas keputusan yang kompleks dan berulang. Hal ini memungkinkan staf untuk fokus pada kegiatan bernilai lebih tinggi dan dapat mengurangi biaya.

Kasus penggunaan dunia nyata di seluruh industri meliputi:

- Layanan keuangan — Agen AI dapat memberikan saran keuangan yang dipersonalisasi dan mendeteksi penipuan.
- Perawatan Kesehatan — Triase dan agen rencana perawatan dapat meningkatkan throughput klinis.
- Retail — Agen dapat bertindak sebagai asisten belanja cerdas atau mengoptimalkan inventaris secara real time.
- Manufaktur — Agen dapat melakukan pemeliharaan prediktif atau mengoordinasikan rantai pasokan.

Area fokus 2: Desain untuk komposisi dan kolaborasi

Job to be done: “Biarkan saya membangun agen seperti saya membangun layanan - modular dan dapat diuji, sehingga mereka dapat disusun dan diatur sesuai kebutuhan.”

Banyak upaya AI dimulai sebagai pilot monolitik, model-sentris. Mereka berguna, tetapi sulit untuk diskalakan di seluruh domain atau beradaptasi dengan masalah yang kompleks. Nilai senyawa ketika agen ini dirancang untuk saling beroperasi. Dalam teknologi, composability adalah tindakan menggabungkan komponen modular untuk menciptakan solusi yang fleksibel dan terukur yang dapat

beradaptasi dengan perubahan. Tanpa komposabilitas, kecerdasan menjadi terkunci dalam alur kerja tertentu. Selain itu, kolaborasi agen memperkenalkan kompleksitas orkestrasi, manajemen negara, dan negosiasi protokol yang mungkin tidak dapat ditangani oleh tim otomatisasi tradisional.

Strategi

Merangkul paradigma multi-agen. Agen model seperti departemen organisasi: modular, khusus, dan interoperable. Tentukan antarmuka yang jelas, format konteks bersama, dan protokol komunikasi standar, seperti Model Context Protocol (MCP) atau Agent2Agent (A2A). Mengadopsi pola orkestrasi multi-agen, seperti kawanan, grafik, atau koordinasi hierarkis. Pola-pola ini membantu agen menemukan kemampuan dan meminta layanan dari satu sama lain secara dinamis, baik dalam alur kerja paralel, sekuensial, atau berdasarkan konsensus, tergantung pada struktur tugas dan tingkat kepercayaan.

Untuk mempromosikan kolaborasi yang terukur dan diatur, gunakan agen arbiter. Agen semacam ini adalah otoritas netral yang memfasilitasi pendelegasian tugas berdasarkan kemampuan yang diketahui dan strategi mundur. Meskipun bukan pengontrol terpusat, agen arbiter memainkan peran penting dalam kepercayaan dan kepatuhan. Ini memastikan bahwa tugas sensitif atau diatur hanya diarahkan ke agen yang memenuhi persyaratan identitas dan kebijakan. Ini bertindak sebagai penjaga gerbang untuk alur kerja terikat kebijakan. Ini memberlakukan isolasi dan memungkinkan delegasi yang dapat dijelaskan. Yang terpenting, agen arbiter bukanlah hambatan; ia hidup berdampingan dengan agen koordinasi diri yang beroperasi secara horizontal. peer-to-peer Agen ini mendelegasikan sub-tugas, berbagi konteks, dan menyelesaikan dependensi secara langsung.

Model hibrida ini mendukung penugasan deterministik (melalui agen arbiter) dan kolaborasi yang muncul. Ini memadukan struktur dengan fleksibilitas. Dalam arsitektur ini, agen dapat diklasifikasikan ke dalam peran khusus berikut:

- Agen keputusan, seperti penegak kebijakan, pengalokasi sumber daya, dan evaluator risiko
- Agen pengetahuan, seperti agregator konteks, pengenal pola, dan detektor anomali
- Agen eksekusi, seperti pelaksana tugas, pengontrol kualitas, dan manajer integrasi

Untuk berkoordinasi secara efektif, sistem multi-agen harus mendukung protokol interaksi yang kuat untuk manajemen negara, pemulihan kegagalan, dan resolusi konflik. Hal ini mendorong stabilitas dan akuntabilitas bahkan ketika agen beroperasi secara independen.

Tetapkan aturan yang jelas untuk penskalaan, seperti instantiasi agen berbasis beban, alokasi sumber daya yang sadar konteks, serta penemuan dan pendaftaran kemampuan otomatis. Langkah-

langkah ini membantu sistem untuk tumbuh secara dinamis dalam menanggapi permintaan atau kompleksitas.

Agen desain untuk menjadi ready-to-use modul dalam substrat pesan terdistribusi. Misalnya, Anda dapat menggunakan Amazon EventBridge dengan A2A atau MCP daripada layanan siloed. Mengadopsi pembuatan versi, CI/CD saluran pipa, dan templat agen untuk mendukung stabilitas sistem sekaligus mempercepat adopsi internal dan evolusi siklus hidup. Dorong penggunaan kembali kode dan standardisasi untuk mengurangi gesekan integrasi dan mempromosikan ekosistem yang tangguh.

Kolaborasi adalah pengganda kekuatan. Ini membuka skala, spesialisasi, dan ketahanan di seluruh lingkungan multi-agen. Untuk mendukung kolaborasi dinamis ini, organisasi harus merancang bidang kontrol ringan untuk koordinasi agen. Bidang kontrol ini mencakup yang berikut:

- Pendaftaran kemampuan yang menentukan apa yang dapat dilakukan setiap agen dan mendukung metadata berversi untuk penemuan rekan
- Logika arbitrase tugas yang menggunakan agen arbiter atau supervisor untuk mengarahkan tugas berdasarkan konteks, ketersediaan, dan kebijakan
- Siklus hidup dan pelacakan status yang memungkinkan konteks keputusan real-time dan handoff yang aman

Pesawat kontrol memastikan bahwa sistem multi-agen tetap dapat diperluas, selaras dengan kebijakan, dan toleran terhadap kesalahan, tanpa memusatkan otoritas atau memperlambat operasi.

Namun, lingkungan multi-agen juga membawa tantangan operasional. Mempertahankan konteks di seluruh interaksi agen, mengelola status bersama, dan mengoordinasikan tindakan dapat mendorong kompleksitas dan biaya. Biaya dapat meningkat jika Anda menggunakan token LLMs tersebut selama komunikasi antar agen. Biaya-biaya ini harus ditimbang terhadap manfaat bisnis gabungan dari otonomi cerdas dalam skala besar.

Untuk mengatasi tantangan ini, pertimbangkan platform agen yang mengabstraksikan masalah utama, seperti berikut ini:

- Protokol komunikasi standar dan format semantik
- Logika orkestrasi bawaan dan perutean dinamis
- Konteks bersama dan manajemen memori antar agen
- Penanganan mundur dan degradasi yang anggun selama kegagalan

Untuk tim yang mengadopsi strategi multi-agen, pendekatan terbaik adalah memulai dari yang kecil dan merancang untuk skala. Mulailah dengan solusi agen tunggal yang ditargetkan yang memecahkan masalah nyata. Kemudian, secara progresif menyusun agen-agen ini ke dalam sistem kooperatif di mana masing-masing dapat menemukan, mengoordinasikan, dan mendelegasikan berdasarkan tujuan bersama dan konteks seluruh sistem.

Yang penting, penanganan kesalahan yang kuat dan degradasi yang anggun harus menjadi prinsip desain utama. Sistem multi-agen harus mampu melanjutkan alur kerja sebagian atau memulai logika cadangan ketika agen tidak tersedia atau gagal. Ini meningkatkan keandalan tanpa kopling kaku.

Layanan AWS menawarkan fitur yang kuat untuk mendukung arsitektur ini dalam skala besar. Amazon EventBridge dan EventBridge Pipes menyediakan backbone terstruktur yang digerakkan oleh peristiwa untuk pengiriman pesan multi-agen. Untuk mengelola perilaku modular, AWS AppConfigaktifkan pengalihan konfigurasi dinamis yang aman di seluruh instance agen. Untuk mendukung konteks bersama dan manajemen memori, gunakan Amazon DynamoDB untuk persistensi status yang ringan dan sadar penyewa serta pengambilan konteks cepat di seluruh agen. Anda dapat menggunakan Amazon Simple Storage Service (Amazon S3) untuk menyimpan histori prompt terstruktur, artefak bersama, atau output yang dihasilkan agen. Untuk alur kerja yang lebih kompleks yang memerlukan koordinasi stateful, AWS Step Functionsdapat mengatur proses yang berjalan lama dengan pos pemeriksaan dan logika pemulihan kesalahan. Bersama-sama, layanan ini membantu Anda membuat sistem multi-agen yang dapat dikomposisi, tangguh, dan terhubung secara semantik yang disesuaikan dengan permintaan perusahaan.

Nilai bisnis sistem multi-agen

Sementara banyak organisasi memulai perjalanan AI mereka dengan solusi agen tunggal, potensi penuh AI agen dibuka melalui sistem multi-agen yang dapat diskalakan. Sistem ini adalah kunci untuk memecahkan masalah yang kompleks dan terdistribusi dan menciptakan ekosistem AI yang kuat dan fleksibel yang berkembang dengan kebutuhan bisnis.

Manfaat bisnis inti dari sistem multi-agen meliputi:

- Skalabilitas — Tugas dan beban kerja dapat didistribusikan ke seluruh agen khusus untuk meningkatkan kapasitas dan kinerja.
- Fleksibilitas — Agen dapat ditambahkan, diganti, atau dimodifikasi dengan gangguan minimal, memungkinkan kelincahan dalam lingkungan yang dinamis.
- Ketahanan — Stabilitas sistem dipertahankan bahkan ketika agen individu gagal, berkat peran yang berlebihan dan kegagalan cerdas.

- Spesialisasi — Agen yang dibuat khusus melakukan tugas dengan efisiensi dan presisi yang lebih besar.
- Efisiensi biaya — Komponen agen yang dapat digunakan kembali mempercepat pengembangan dan mengurangi biaya penyebaran kemampuan baru.

Sementara sistem multi-agen membutuhkan perencanaan yang lebih awal, mereka memberikan kelincahan jangka panjang, kecepatan, dan kapasitas inovasi. Perusahaan yang berinvestasi dalam arsitektur kolaborasi agen fleksibel diposisikan untuk menerapkan kemampuan AI baru dengan cepat, beradaptasi dengan permintaan yang berubah, dan memimpin dalam lanskap kompetitif yang semakin didorong oleh agen.

Area fokus 3: Arsitek untuk multi-tenancy dan kontrol

Job to be done: “Bantu saya meningkatkan penggunaan agen di beberapa pelanggan tanpa kehilangan kendali, akuntabilitas, atau visibilitas.”

Prototipe awal baik-baik saja untuk membuktikan nilai secara terpisah, tetapi sebagian besar bisnis perlu secara bersamaan mendukung banyak pelanggan, departemen, atau alur kerja. Itu berarti setiap agen harus beroperasi dalam batas-batas kebijakan, data, dan identitas yang jelas. Tanpa multi-tenancy, operasi menjadi rapuh dan mahal, dan tata kelola menjadi tambal sulam.

Strategi

Ikuti prinsip-prinsip dari arsitektur perangkat lunak sebagai layanan (SaaS). Misalnya, desain untuk isolasi penyewa, penegakan kebijakan, dan kontrol sumber daya. Agen arsitek dan platform orkestrasi dengan memori, konfigurasi, dan identitas yang sadar penyewa. Untuk menegakkan batasan, gunakan penandaan, kontrol akses berbasis peran (RBAC), dan pelingkupan manajemen identitas dan akses.

Mengadopsi lapisan observabilitas terpadu di mana telemetri agen dikumpulkan berdasarkan konteks penyewa. Menerapkan mesin kebijakan terpusat dan pengalihan kemampuan berbasis konfigurasi untuk menegakkan aturan perilaku dinamis.

Membangun penyebaran agen sebagai layanan. Memungkinkan tim internal atau pelanggan untuk menggunakan kemampuan agen sebagai skalabel, diatur APIs. AWS memberikan dasar yang kuat untuk pola-pola ini. Anda dapat menggunakan Amazon Cognito untuk mengelola identitas pengguna dan penyewa, AWS Organizationsserta kebijakan kontrol layanan (SCPs) untuk tata kelola lintas akun, dan AWS Resource Access Manager (AWS RAM) untuk berbagi kemampuan dengan

aman. Selain itu, AWS AppConfigdapat secara dinamis mengelola perilaku agen oleh penyewa atau lingkungan. Layanan ini membantu menegakkan batasan dan kebijakan sambil mendukung infrastruktur bersama.

Transisi dari penyebaran statis ke penyediaan dinamis ini mengubah AI agen menjadi platform di seluruh perusahaan.

Nilai bisnis platform agen multi-penyewa

Multi-tenancy lebih dari sekadar kenyamanan arsitektur—ini adalah akselerator bisnis. Ketika agen cerdas berkembang biak di seluruh departemen dan tim, organisasi harus mendukung pertumbuhan tanpa menduplikasi infrastruktur atau memecah tata kelola.

Manfaat bisnis utama dari sistem multi-tenant meliputi:

- Skalabilitas — Platform agen multi-penyewa memungkinkan tim internal, unit bisnis, atau klien untuk menggunakan kemampuan AI lebih cepat tanpa memerlukan lingkungan yang dipesan lebih dahulu.
- Efisiensi biaya — Infrastruktur bersama meminimalkan penyebaran yang berlebihan, mengkonsolidasikan biaya operasional, dan menyederhanakan pemeliharaan di seluruh lingkungan.
- Tata kelola dan pengurangan risiko - Kontrol kebijakan terpusat, model identitas, dan observabilitas membantu agen beroperasi lebih aman dan sesuai, di semua penyewa.
- Penggunaan kembali layanan — Untuk mempromosikan penggunaan kembali dan mengurangi duplikasi, agen yang sadar penyewa dapat ditawarkan sebagai layanan internal, seperti untuk pengayaan, kepatuhan, atau ringkasan.

Contoh kasus penggunaan untuk sistem multi-penyewa meliputi yang berikut:

- Agen kepatuhan yang ditempatkan di seluruh anak perusahaan menyesuaikan logikanya dengan peraturan lokal melalui konfigurasi khusus penyewa. Ini menghilangkan kebutuhan untuk membangun agen terpisah untuk setiap wilayah.
- Agen otomatisasi alur kerja internal melayani beberapa departemen dengan batas data dan izin yang berbeda. Ini mempertahankan isolasi sambil mempercepat pemenuhan tugas.

Dengan merancang agen sebagai multi-tenant-aware layanan, organisasi menghindari overhead inisiatif AI tersilo. Sebaliknya, mereka mengembangkan platform intelijen terpadu. Arsitektur ini

memungkinkan peluncuran yang dapat diskalakan, konsistensi operasional, dan ROI yang lebih baik. Ini juga membuatnya lebih mudah untuk memperluas adopsi AI di seluruh perusahaan.

Area fokus 4: Membangun kepercayaan melalui identitas, pagar pembatas, dan observabilitas

Job to be done: “Beri saya keyakinan bahwa agen akan bertindak aman dan dapat diprediksi, terutama ketika tidak ada yang menonton.”

Agen otonom menantang model kontrol tradisional. Kemampuan mereka untuk bernalar dan bertindak secara independen menimbulkan risiko jika mereka tidak dikelola dengan baik. Tanpa kepemilikan, auditabilitas, atau kendala kebijakan yang jelas, mereka dapat menyimpang dari perilaku yang dimaksudkan. Membangun kepercayaan organisasi membutuhkan lebih dari sekedar keandalan teknis. Ini menuntut penjelasan, akuntabilitas, dan konsistensi.

Strategi

Membangun sistem kontrol identitas pertama sebagai tulang punggung otonomi tepercaya. Setiap agen harus beroperasi dengan identitas yang dapat diverifikasi, izin tercakup, dan riwayat eksekusi yang dapat dilacak. Agen harus disematkan dalam kerangka kerja tanpa kepercayaan yang mencakup pengikatan penyewa, pewarisan akses kontekstual, dan penegakan runtime melalui pagar pembatas dan mesin kebijakan. Ini memungkinkan Anda untuk mengaudit, membalikkan, atau membatasi tindakan agen berdasarkan aturan organisasi dan postur risiko.

Sematkan penegakan kepercayaan saat runtime melalui pagar pembatas cerdas. Ini termasuk kontrol tarif dan pembatasan berdasarkan pola perilaku atau kondisi beban kerja, batasan sumber daya yang diberlakukan bersama auto-scaling, dan penilaian keputusan untuk mengevaluasi risiko. Pemicu build untuk melibatkan human-in-the-loop alur kerja saat ambang batas terlampaui.

Setiap agen juga harus transparan dan dapat dijelaskan. Sematkan telemetri terstruktur melalui pencatatan, jejak, dan ringkasan penalaran untuk mengekspos logika keputusan. Mendukung jalur keputusan dan penelusuran dampak. Ini membantu Anda menghubungkan tindakan agen kembali ke metrik atau hasil utama. Menerapkan mekanisme deteksi drift yang memantau penyimpangan dari perilaku atau kebijakan yang diharapkan.

Memperkenalkan agen reflektif yang terus mengamati perilaku agen dan pola sistem. Mereka harus menandai anomali atau inkonsistensi secara real time. Agen-agen ini berkontribusi pada loop umpan balik tata kelola yang dapat memulai validasi ulang, adaptasi, atau penonaktifan kemampuan.

Menetapkan dewan tata kelola yang meninjau kebijakan agen, menyetujui perubahan kemampuan, dan mengawasi protokol respons insiden. Kepercayaan harus diperoleh, diukur, dan terus diperkuat.

AWS memberikan dasar yang kuat untuk menerapkan kerangka kepercayaan ini:

- AWS Identity and Access Management (IAM) memberlakukan eksekusi berbasis peran dan batas izin
- Amazon CloudWatch dan AWS X-Raymendukung visibilitas penuh dan keterlacakan.
- Amazon GuardDuty dan AWS Configmendeteksi anomali keamanan atau penyimpangan kebijakan.

Bersama-sama, layanan ini memungkinkan penegakan identitas, keamanan runtime, dan tata kelola berbasis kepercayaan dalam skala besar. Mereka dapat membantu membuat sistem otonom menjadi kuat dan dapat diandalkan.

Nilai bisnis otonomi tepercaya

Ketika agen menjadi lebih otonom, kepercayaan menjadi pendorong penting untuk adopsi perusahaan, tata kelola, dan kinerja operasional. Membangun fondasi identitas, observabilitas, dan pagar pembatas membantu organisasi untuk menskalakan AI agen ke dalam domain sensitif, tanpa mengorbankan tata kelola atau kontrol.

Penggerak bisnis utama meliputi yang berikut:

- Jaminan tata kelola — Model identitas yang kuat, jalur audit, dan batas izin mengurangi risiko kepatuhan dan mendukung penyelarasan peraturan.
- Kontinuitas operasional — Runtime guardrails dan deteksi anomali membantu mencegah perilaku yang tidak diinginkan dan mendukung pemulihan diri dari kegagalan edge-case.
- Kepercayaan pemangku kepentingan — Penjelasan keputusan dan telemetri membangun kepercayaan dengan pemangku kepentingan internal, manajer risiko, dan auditor eksternal.
- Ketahanan insiden - Observabilitas tertanam mempercepat analisis akar penyebab dan waktu respons ketika masalah muncul.

Contoh kasus penggunaan meliputi:

- Dalam layanan keuangan, agen pendeteksi penipuan harus mengekspos alasan mereka, mencatat setiap tindakan dengan identitas yang dapat dilacak, dan beroperasi di bawah peran IAM yang tercakup ketat.
- Dalam perawatan kesehatan, agen triase otonom harus menegakkan pemeriksaan keamanan runtime, meningkat ke tinjauan manusia ketika ambang batas terpenuhi, dan menyediakan log lengkap untuk pengawasan klinis.

Dengan menanamkan mekanisme kepercayaan ke dalam siklus hidup agen, organisasi dapat mengizinkan sistem mereka beroperasi secara mandiri dengan akuntabilitas. Yayasan ini mengurangi risiko dan memberdayakan agen untuk bertindak atas nama bisnis dengan transparansi dan integritas.

Pada akhirnya, otonomi tepercaya mempercepat adopsi dengan memberikan kepercayaan kepada pengguna dan kepemimpinan untuk meningkatkan skala agen cerdas di seluruh operasi inti.

Area fokus 5: Kelola siklus hidup

Job to be done: “Pastikan tim saya dapat meningkatkan agen dari waktu ke waktu, tanpa kekacauan atau kepahlawanan.”

Tidak seperti aplikasi tradisional yang hanya dibentuk oleh kode, perilaku agen juga dibentuk oleh prompt, memori, alat, dan konteks pelatihan. Faktor-faktor ini melayang seiring waktu. Drift mengikis keandalan, meningkatkan biaya, dan membuat debugging hampir tidak mungkin. Tanpa kontrol siklus hidup, agen berhenti memberikan nilai dan mulai mengumpulkan risiko.

Strategi

Menetapkan DevOps untuk agen (AgentOps) sebagai praktik. Integrasikan CI/CD jaringan pipa yang disesuaikan untuk agen. Gunakan pipeline ini untuk menguji output yang cepat, memvalidasi integrasi alat, dan perilaku kinerja biaya profil. Pertahankan riwayat versi petunjuk, kebijakan, dan interaksi model.

Gunakan loop umpan balik dari data observabilitas untuk memulai pelatihan ulang, penyetelan cepat, atau pensiun agen. Menggabungkan mekanisme refleksi seluruh sistem, seperti daftar perbaikan, untuk melembagakan pembelajaran.

Bangun dasbor telemetri kinerja yang menunjukkan akurasi keputusan, latensi, biaya, dan keandalan. Untuk merampingkan dan mempercepat manajemen siklus hidup menggunakan AWS infrastruktur, tim dapat menggunakan toolkit agen. Salah satu contohnya adalah Strands Agents SDK, yang

menyediakan perkakas terstruktur untuk pembuatan versi yang cepat, pendaftaran alat, dan integrasi CI/CD dengan Layanan AWS, seperti, dan. AWS CodePipelineAWS Cloud Development Kit (AWS CDK)AWS Lambda Selain itu, gunakan Amazon S3 dan Amazon Elastic File System (Amazon EFS) untuk menyimpan artefak agen dan data pelatihan. Gunakan AWS Step Functionsuntuk mengotomatiskan alur kerja pelatihan ulang atau validasi yang kompleks. Anda dapat menggunakan Amazon SageMaker AI saat agen memerlukan penyetelan model khusus atau alur kerja fine-tuning di luar orkestrasi LLM. Disiplin siklus hidup mengubah agen dari eksperimen menjadi aset yang tahan lama dan berkembang.

Seiring waktu, sistem siklus hidup ini membentuk tulang punggung inovasi. Ini membantu Anda untuk menyusun ulang, melatih kembali, dan menerapkan kembali kemampuan dengan kelincahan. Ini mengubah lapisan agen menjadi sistem kehidupan, yang mampu berkembang sebagai respons terhadap umpan balik dan peluang.

Nilai bisnis manajemen siklus hidup

Manajemen siklus hidup yang efektif adalah pendorong utama kinerja agen dan efisiensi biaya. Ini memastikan bahwa agen cerdas terus memberikan hasil yang akurat, andal, dan selaras dengan nilai saat mereka berkembang. Agen tidak tetap berharga secara default. Mereka harus berkembang selaras dengan perubahan persyaratan bisnis, alur kerja, dan lingkungan data. AgentOps Tim yang disiplin membantu agen tetap akurat, efisien, dan selaras dengan tujuan perusahaan dari waktu ke waktu.

Penggerak bisnis utama meliputi yang berikut:

- Konsistensi kinerja - Pengujian berkelanjutan, validasi cepat, dan pelatihan ulang membantu agen mempertahankan kualitas keputusan di seluruh kondisi dan kumpulan data yang berubah.
- Optimalisasi biaya — Profil berbasis Telemetri mengidentifikasi alat yang tidak efisien, permintaan token tinggi, atau eksekusi yang tidak perlu. Kemudian, Anda dapat menyetel untuk mengurangi biaya operasional.
- Iterasi lebih cepat — Otomatisasi siklus hidup dengan CI/CD mempercepat siklus pengembangan, membantu tim untuk bereksperimen, menerapkan, dan meningkatkan agen dengan percaya diri.
- Pengurangan risiko — Pembuatan versi yang cepat, dukungan rollback, dan mekanisme evaluasi terstruktur membantu mencegah regresi dan mendukung manajemen perubahan yang aman dan andal.

Contoh kasus penggunaan meliputi yang berikut:

- Agen dukungan pelanggan dipantau untuk latensi, biaya model, dan umpan balik pengguna. Observabilitas mengungkapkan lonjakan biaya, yang mendorong penyetelan ulang prompt yang disematkan dan logika model mundur.
- Agen ringkasan kontrak diperbarui berdasarkan umpan balik dari tim hukum. Permintaan berversi diuji di lingkungan kotak pasir sebelum rilis produksi, mendukung keamanan dan kualitas.

Dengan manajemen siklus hidup terstruktur, organisasi bergerak melampaui pemeliharaan reaktif ke perbaikan proaktif dan berkelanjutan. Agen menjadi aset digital adaptif yang diukur, disempurnakan, dan divalidasi ulang terhadap tujuan bisnis. Praktik ini mengubah ekosistem agen menjadi sistem berkinerja tinggi, sadar biaya, dan tangguh yang memberikan nilai tahan lama sambil mengikuti perubahan.

Area fokus 6: Sejajarkan model agen dengan model bisnis

Job to be done: “Tunjukkan dampaknya, sehingga saya bisa membenarkan investasi berkelanjutan.”

Bahkan agen yang mampu secara teknis menjadi kewajiban jika mereka tidak terikat dengan hasil bisnis. Agen harus melayani efisiensi, monetisasi, atau diferensiasi strategis. Namun sebagian besar bisnis berjuang untuk menentukan bagaimana agen masuk ke dalam model harga, kemasan, atau penggunaan. Tanpa keselarasan yang jelas dengan nilai bisnis, sulit untuk membenarkan penskalaan atau bahkan mempertahankan investasi.

Strategi

Mengadopsi praktik manajemen produk. Perlakukan agen sebagai layanan yang dapat dimonetisasi dengan ROI yang terukur. Tentukan strategi penetapan harga berdasarkan keputusan, sesi, atau hasil. Kemudian, paket kemampuan agen menjadi penawaran berjenjang yang selaras dengan segmen pelanggan atau unit bisnis internal.

Untuk mempromosikan keberlanjutan, organisasi harus menangkap nilai langsung dan pengganda pertumbuhan melalui penyebaran agen. Pertimbangkan untuk menggunakan metrik ROI berikut untuk mengukur nilai langsung:

- Biaya per keputusan — Biaya pemrosesan agen benchmark terhadap setara manusia.
- Kompresi waktu — Mengukur nilai siklus yang dipercepat, seperti penjualan atau persetujuan yang lebih cepat.
- Pengurangan kesalahan - Ukur penghematan dari peningkatan akurasi, konsistensi, dan kepatuhan.

Di luar keuntungan langsung ini, agen dapat membuka peluang pertumbuhan jangka panjang berikut:

- Capability stacking — Gabungkan layanan agen untuk membuat solusi vertikal khusus domain.
- Efek jaringan — Meningkatkan nilai melalui ekosistem multi-agen di mana senyawa koordinasi digunakan.
- Perluasan pasar — Menghasilkan aliran pendapatan baru melalui layanan yang dapat dikonsumsi secara eksternal dan berkemampuan agen.

Buat loop umpan balik dari metrik bisnis (seperti penghematan biaya, peningkatan konversi, atau time-to-resolution) untuk mendorong evolusi agen yang berkelanjutan. Analisis telemetri penggunaan dan skor kepuasan pengguna untuk menyempurnakan penyelarasan nilai dan prioritas peta jalan Anda. Dengan menghubungkan kapabilitas agen secara langsung ke model bisnis, organisasi memposisikan diri untuk menangkap nilai yang berkelanjutan dan majemuk — bukan hanya hasil teknis.

Berikut ini Layanan AWS mendukung penyelarasan ini dengan menyediakan kerangka kerja pelacakan dan monetisasi yang kuat:

- AWS Cost Explorerdan Amazon CloudWatch memberikan wawasan tentang biaya per agen dan efisiensi operasional.
- Amazon API Gateway memungkinkan akses terukur, pembatasan tarif, dan harga berjenjang untuk titik akhir agen.
- AWS Marketplacemenyediakan saluran untuk agen penerbitan dan solusi agen sebagai produk komersial.

Layanan ini membantu Anda mengubah fungsionalitas agen menjadi penawaran digital yang dapat diskalakan dan didorong oleh nilai yang selaras dengan strategi pertumbuhan dan monetisasi perusahaan.

Pengiriman perangkat lunak yang berkembang untuk AI agen

Pengiriman perangkat lunak modern telah dibentuk oleh asumsi sederhana — bahwa Anda mengontrol sistem yang Anda kirimkan. Anda menentukan persyaratan, menulis logika, menguji hasil yang diharapkan, dan menerapkan layanan yang dapat diprediksi. Bahkan Agile dan DevOps pendekatan masih mengandalkan prinsip bahwa setiap sprint memberikan sesuatu yang deterministik, dapat diverifikasi, dan sebagian besar dalam pengawasan manusia.

Agentic AI menjungkirbalikkan fondasi itu. Sistem agen menafsirkan, bernalar, dan beradaptasi daripada mengikuti skrip. Perilaku mereka bergantung pada kode yang Anda tulis, konteks tempat mereka beroperasi, input yang diberikan, alat yang dapat mereka akses, dan tujuan yang ditetapkan. Singkatnya, mereka tidak mengikuti perintah; mereka mengejar hasil.

Ini membuat pengiriman lebih sedikit tentang kontrol dan lebih banyak tentang penyelarasan. Daripada memberikan instruksi, Anda harus membentuk bagaimana perilakunya. Ini berarti bahwa siklus hidup pengembangan perangkat lunak tradisional (SDLC) tidak lagi cocok karena dirancang untuk sistem berbasis logika yang dikendalikan manusia.

Bagian ini berisi topik berikut:

- Zona niat untuk AI agen
- Mengembangkan siklus hidup pengiriman untuk AI agen
- Mempersiapkan tim untuk AI agen

Zona niat untuk AI agen

Alih-alih tahapan kaku, seperti mendefinisikan, membangun, menguji, dan melepaskan, kita membutuhkan model yang mencakup otonomi, ketidakpastian, dan kemunculan. Sebagai gantinya, Anda menggunakan zona niat. A z one of intent mendefinisikan ruang terbatas di mana agen dapat beroperasi dengan otonomi, dalam batasan. Tujuannya adalah untuk beralih dari micromanaging setiap tugas ke merancang lingkungan di mana agen dapat bertindak, belajar, dan berkolaborasi dengan aman. Anda menentukan apa (hasil yang diinginkan), mengapa (maksud), dan pagar pembatas (batasan, kebijakan, dan batas kepercayaan). Mengingat batas-batas dan informasi ini, agen mencari tahu caranya.

Alih-alih jalur perakitan, pikirkan lingkungan sebagai wilayah udara. Anda mengontrol siapa yang bisa masuk, apa yang bisa mereka lakukan, dan ke mana mereka bisa pergi. Tapi begitu masuk, mereka bebas menavigasi sesuai kebutuhan. Begitulah skala sistem agen tanpa kekacauan.

Ini bukan hanya pergeseran filosofis; itu praktis. Output non-deterministik dari sistem berbasis agen tidak dapat sepenuhnya diuji melalui pengujian unit. Itu tidak dapat diversikan seperti binari statis. Agen berubah seiring waktu, beradaptasi dengan data baru, dan berinteraksi dengan sistem lain dengan cara yang tidak terduga. Mencoba mengirimkannya menggunakan model tradisional mengarah ke arsitektur yang rapuh dan tidak dapat diskalakan. Paling buruk, itu mengarah pada kepercayaan palsu pada sistem yang sebenarnya tidak dapat Anda atur.

Ketika tim merangkul pengiriman berbasis niat, mereka mendapatkan dua keuntungan:

- Kontrol di mana yang paling penting — Mereka mendefinisikan batas, bukan output.
- Skalabilitas melalui delegasi — Mereka memungkinkan agen untuk menangani kompleksitas manusia tidak dapat melakukan hardcode.

Ini adalah bagaimana Anda beralih dari prototipe terisolasi ke sistem agen tingkat produksi nyata yang dapat berulang kali dan andal memberikan nilai.

Mengembangkan siklus hidup pengiriman untuk AI agen

Untuk mendukung perilaku adaptif yang cerdas, SDLC harus dibingkai ulang dari kontrol deterministik ke niat adaptif. Berikut ini adalah perubahan yang diperlukan untuk mengembangkan SDLC tradisional untuk AI agen:

- Perencanaan menjadi desain niat. Tim menentukan tujuan, kendala, dan perilaku agen yang diharapkan. Kebijakan dan kriteria keberhasilan dibingkai dalam hal keselarasan, bukan logika.
- Arsitektur menjadi perancah. Tim fokus pada mendefinisikan peran, antarmuka, pagar pembatas, mekanisme mundur, dan pengamatan daripada membuat skrip setiap jalur keputusan.
- Pengujian menjadi evaluasi perilaku. Alih-alih menegaskan output tertentu, tim memvalidasi apakah agen tetap dalam batas yang dapat diterima dan memenuhi niat di bawah masukan yang bervariasi.
- Deployment menjadi orkestrasi berkelanjutan. Sistem agen digunakan dengan kontrol runtime, pemantauan langsung, dan saluran umpan balik yang memungkinkan penyetelan waktu nyata.
- Iterasi menjadi umpan balik dan adaptasi. Alih-alih siklus patch perubahan kode tradisional, tim mengamati bagaimana agen berevolusi, di mana mereka berhasil, atau ketika mereka melayang.

Jika perlu, tim melakukan intervensi melalui kendala yang diperbarui, pelatihan ulang, dan penambahan atau modifikasi mekanisme kontrol.

Praktik yang ada yang berfokus pada iterasi, eksperimen, dan umpan balik cepat ada di tengah jalan. Pergeseran ke sistem agen bukanlah penolakan terhadap prinsip-prinsip Agile. Faktanya, ini adalah evolusi alami dari mereka. Pemikiran tangkas menekankan kemampuan beradaptasi, umpan balik, dan solusi kerja di atas rencana yang kaku. Itu selaras sempurna dengan sifat sistem agen, yang belajar, beradaptasi, dan merespons konteks secara real time. Jika Anda sudah menjalankan siklus pendek, memvalidasi asumsi dengan cepat, dan mengelola ketidakpastian melalui pengiriman berkelanjutan, Anda diperlengkapi dengan baik untuk memimpin transisi ini.

Tetapi ada perbedaan utama. Pendekatan Agile tradisional mengasumsikan bahwa hal yang disampaikan adalah deterministik. Ini mengasumsikan bahwa, setelah dibangun, benda itu akan berperilaku konsisten dan dapat diprediksi, dengan hasil berulang untuk input yang sama. Pengulangan ini membantu Anda men-debug, menguji, dan mengulangi dengan percaya diri. Sistem agen merusak model itu. Mereka probabilistik, peka konteks, dan mampu berkembang secara independen. Itu berarti beberapa praktik Agile menjadi kurang berguna, seperti pelacakan kecepatan berdasarkan penyelesaian cerita, kriteria penerimaan yang ketat, atau perencanaan sprint deterministik.

Aspek SDLC tradisional berikut berlaku untuk AI agen:

- Pengembangan dan pengiriman berulang
- Umpan balik pelanggan sebagai sinyal utama
- Kolaborasi lintas fungsi
- Integrasi dan penyebaran berkelanjutan

Aspek SDLC tradisional berikut harus berkembang untuk AI agen:

- Mendefinisikan ulang dilakukan sebagai selaras dengan maksud. Fokus pada apakah perilaku agen memenuhi tujuan yang dimaksudkan dalam batasan yang ditentukan.
- Beralih dari kriteria penerimaan ke pagar perilaku.
- Perluas definisi done untuk memasukkan kesiapan runtime, yang mencakup observabilitas, penjelasan, dan mekanisme umpan balik yang mendukung pembelajaran dan kepercayaan berkelanjutan.
- Prioritaskan loop umpan balik waktu nyata dan pelacakan perilaku di atas perencanaan di muka

Kabar baiknya adalah Anda tidak perlu membuang buku pedoman SDLC. Anda hanya perlu mengembangkannya dari mengelola kode hingga membentuk perilaku. Dalam sistem agen, kesuksesan bukan hanya tentang apakah perangkat lunak berjalan, tetapi bagaimana perilakunya.

Mempersiapkan tim untuk AI agen

Rekayasa perangkat lunak tidak akan hilang. Ini berkembang. Pekerjaan bergeser dari fungsi menulis ke membentuk kerangka kerja dan mekanisme kontrol untuk perilaku cerdas. Dalam dunia AI agen, membangun bukan lagi bagian yang sulit — mengelola kemunculan adalah. Bagi sebagian besar tim teknik, evolusi terasa seperti pergeseran pola pikir daripada lompatan teknis. Alih-alih bertanya “Apa yang akan dilakukan sistem?” pertanyaannya menjadi “Apa yang telah kita berdayakan untuk dikejar, dan bagaimana kita akan tahu jika itu tetap di jalur?”

Untuk tim teknik, evolusi menuju agen AI memerlukan perubahan berikut:

- Pergeseran budaya — Tim harus merasa nyaman dengan ketidakpastian dan otonomi dalam sistem yang tidak mereka kendalikan sepenuhnya.
- Peran baru - Desainer niat, penguji perilaku, dan insinyur observabilitas menjadi inti pengiriman.
- Bahasa bersama — Tim membutuhkan pemahaman yang jelas dan bersama tentang tujuan, pagar pembatas, dan sinyal keberhasilan, seperti bagaimana mereka membutuhkan spesifikasi dan kasus uji.

Saat AI generatif matang, kita akan melihat lebih banyak sistem agen berinteraksi dengan pelanggan, produk, dan operasi. Organisasi yang sukses tidak akan menjadi orang-orang dengan model terbaik. Ini akan menjadi orang-orang yang dapat mengintegrasikan agen ke dalam alur kerja dunia nyata dengan kepercayaan diri, kontrol, dan kecepatan. Itu berarti bahwa model pengiriman dan tim teknik harus berkembang bersama. Zona niat memberi Anda abstraksi untuk melakukan itu. Mereka membantu Anda mengoperasionalkan otonomi tanpa menyerahkan akuntabilitas. Mereka juga menawarkan kerangka kerja bersama di seluruh tim untuk membantu mengatur sistem yang tidak dapat dikodekan dengan keras.

Untuk informasi lebih lanjut tentang mempersiapkan tim untuk AI agen, lihat bagian Mempersiapkan bisnis untuk AI agen pada skala besar dari panduan ini.

Mempersiapkan bisnis untuk AI agen dalam skala besar

Saat area fokus yang dijelaskan dalam panduan ini bertemu, AI agen bergeser dari fungsi yang terisolasi menjadi lapisan intelijen terpadu yang dapat dipahami sebagai platform kemampuan. Platform ini tidak hanya menjalankan tugas. Ini berevolusi, beradaptasi, dan berkoordinasi di seluruh domain. Agen menjadi layanan modular, dapat digunakan kembali, dan dapat ditemukan yang mempercepat inovasi, mengurangi beban kognitif, dan mendorong hasil yang terukur di seluruh perusahaan. Tampilan platform ini menetapkan panggung untuk kecerdasan terukur yang tertanam di seluruh model operasi.

Mengoperasionalkan AI agen membutuhkan lebih dari sekadar menyebarkan agen cerdas. Ini menuntut transformasi mendasar dalam bagaimana bisnis mengatur tim, merancang proses, dan mengatur teknologi. Sama seperti pergeseran ke cloud atau model operasi yang DevOps didefinisikan ulang, AI agen memperkenalkan era baru otomatisasi keputusan, pembelajaran berkelanjutan, dan koordinasi otonom. Keberhasilan tergantung pada penyelarasan sistem, orang-orang, dan proses di sekitar filosofi operasi baru ini.

Bagian ini berisi topik berikut:

- Menyelaraskan tim dan model kepemilikan
- Mengelola perubahan dan kesiapan organisasi
- Arsitektur untuk interoperabilitas dan kolaborasi
- Membangun tata kelola menjadi kain agen
- Mengadopsi pola pikir operasi keputusan-pertama
- Penskalaan dengan tujuan dan niat

Menyelaraskan tim dan model kepemilikan

Langkah pertama menuju kedewasaan adalah penyelarasan lintas fungsi. Bisnis harus membentuk AgentOps tim yang mencakup AI/ML praktisi dan spesialis domain, seperti arsitek sistem terdistribusi, insinyur perangkat lunak, pemilik produk, prospek kepatuhan, dan arsitek platform. Tim-tim ini bersama-sama memiliki seluruh siklus hidup agen, mulai dari desain dan penerapan hingga pelatihan ulang dan pemantauan.

Penyediaan dan rilis agen harus mengikuti praktik cloud-native, seperti menggunakan AWS Cloud Development Kit (AWS CDK)dan AWS CodePipelineuntuk infrastruktur sebagai kode dan penerapan

otomatis. Struktur ini mendorong akuntabilitas bersama dan mempercepat iterasi. Sama seperti DevOps menyatukan pembangunan dan operasi, AgentOps menghubungkan intelijen dengan pemerintahan dan eksekusi.

Agar efektif, tim-tim ini juga membutuhkan bahasa bersama. Pemangku kepentingan bisnis harus memahami apa itu agen, bagaimana mereka beroperasi, dan hasil apa yang mereka dorong. Pelatihan dan pemberdayaan internal sangat penting. Dengan mengungkap agen dan menanamkan model mental ini ke dalam percakapan sehari-hari, organisasi membuka partisipasi yang lebih luas dan inovasi yang lebih selaras.

Untuk mempercepat pengembangan dan integrasi penggunaan agen Layanan AWS, tim dapat mengadopsi kerangka kerja seperti Strands Agents SDK, yang menawarkan perkakas berbasis CLI untuk perancah, konfigurasi, dan agen pengemasan. Strands Agents dirancang untuk bekerja secara mulus dengan AWS infrastruktur, seperti Amazon Bedrock, AWS LambdaAmazon, the EventBridge AWS CDK, dan. AWS CodePipeline Ini memungkinkan pembuatan prototipe dan penyebaran yang cepat sambil mempertahankan standar tingkat produksi.

Tetapi struktur dan perkakas saja tidak cukup. Penskalaan AI agen membutuhkan kesiapan budaya, pendidikan, dan kepemimpinan yang disengaja untuk memastikan bahwa adopsi berakar di seluruh organisasi.

Mengelola perubahan dan kesiapan organisasi

Berhasil menskalakan AI agen membutuhkan lebih dari sekadar menyebarkan infrastruktur atau agen cerdas. Ini menuntut pendekatan terstruktur untuk perubahan organisasi. Ini termasuk kesiapan budaya, pengembangan keterampilan, loop umpan balik berbasis metrik, dan penyelarasan eksekutif untuk memastikan bahwa adopsi disengaja dan berkelanjutan.

Mendorong evolusi budaya

- Posisikan agen sebagai rekan satu tim, bukan pengganti, untuk mengurangi resistensi dan membangun kepercayaan.
- Berkomunikasi secara transparan tentang kemampuan dan keterbatasan agen untuk menetapkan harapan yang realistis.
- Menetapkan protokol handoff yang jelas ketika agen harus meningkatkan keputusan ke otoritas yang lebih tinggi atau mendelegasikan bagian dari proses kepada kolaborator manusia.

Membangun kerangka pengembangan keterampilan

- Memberikan pelatihan berbasis peran yang disesuaikan dengan insinyur, manajer produk, prospek domain, dan petugas kepatuhan.
- Buat pusat keunggulan untuk berbagi praktik terbaik, pola perkakas, dan aset yang dapat digunakan kembali.
- Pasangkan spesialis AI dengan pakar domain melalui program bimbingan untuk menjembatani kesenjangan pengetahuan.

Tentukan metrik dan loop umpan balik

- Jangkar teknis dan bisnis KPIs ke nilai strategis untuk menilai dampak. Contoh nilai termasuk latensi keputusan, akurasi resolusi, dan penghematan biaya.
- Secara sistematis dan terus menerus menangkap umpan balik pengguna ke titik gesekan permukaan dan tantangan adopsi.
- Lakukan retrospektif reguler untuk mengevaluasi kinerja agen, tren penggunaan, dan peluang peningkatan.

Menyelaraskan kepemimpinan dari atas

- Dapatkan sponsor eksekutif dengan menghubungkan inisiatif agen dengan hasil strategis dan ROI.
- Membentuk komite tata kelola lintas fungsi yang mencakup kepemimpinan teknis dan bisnis.
- Menyesuaikan strategi komunikasi untuk kejelasan dan keterlibatan di semua tingkat organisasi.

Pendekatan sistematis untuk manajemen perubahan ini memastikan bahwa implementasi teknologi dicocokkan dengan kematangan organisasi. Ini menciptakan fondasi untuk kepercayaan, adopsi, dan nilai bisnis jangka panjang.

Arsitektur untuk interoperabilitas dan kolaborasi

Penyebaran agen yang terisolasi memberikan kemenangan lokal. Tetapi nilai bisnis muncul ketika agen dapat menemukan, memanggil, dan berkolaborasi satu sama lain secara dinamis. Ini berarti mendefinisikan standar untuk pendaftaran agen, otentikasi, dan pertukaran kemampuan. Secara arsitektur, ini mencerminkan pergeseran dari monolit ke layanan mikro, yang merupakan unit yang

dapat dikomposisi, dapat digunakan kembali, dan digabungkan secara longgar yang memecahkan masalah kompleks bersama-sama.

Protokol yang muncul, seperti A2A dan MCP, adalah dasar. Memungkinkan interoperabilitas semantik di seluruh agen, alat, dan sistem memori. A2A mendukung interaksi tingkat rekan, yang memungkinkan agen untuk menegosiasikan kepemilikan tugas, berbagi konteks, dan mengoordinasikan alur kerja. MCP melengkapi ini dengan menawarkan skema bersama untuk bertukar data kontekstual antara agen dan lingkungan mereka. Ini menstandarisasi bagaimana fungsi dipanggil, APIs diakses, dan status dipertahankan. Bersama-sama, protokol ini mempromosikan ekstensibilitas, konsistensi, dan pemeliharaan jangka panjang di seluruh ekosistem agen.

Tata kelola tetap kritis. Lapisan kontrol, seperti agen arbiter, memungkinkan delegasi sadar kebijakan tanpa menimbulkan kemacetan terpusat. Agen ini bertindak sebagai broker kepercayaan. Mereka menegakkan batasan sambil membiarkan agen lain mengatur diri sendiri. Kolaborasi agen membantu organisasi untuk meningkatkan ekosistem AI agen mereka dengan kelincahan dan kepercayaan.

Membangun tata kelola menjadi kain agen

Dengan otonomi yang lebih besar datang risiko yang lebih besar. Tata kelola harus disematkan ke dalam arsitektur agen sejak hari pertama. Ini termasuk mendefinisikan batas-batas kebijakan yang mencakup apa yang diizinkan untuk dilakukan agen, menegakkan model identitas yang menentukan siapa mereka bertindak atas nama, dan menerapkan penjelasan dan keterlacakan. Sistem observabilitas harus menangkap telemetri pada perilaku agen dengan menggunakan layanan seperti Amazon CloudWatch dan AWS X-Ray, yang menyediakan pencatatan terpusat dan penelusuran terdistribusi di seluruh alur kerja agen. Agen reflektif dapat terus mengaudit dan menilai kinerja berdasarkan umpan telemetri ini.

Tata kelola juga harus berkembang saat ekosistem agen matang. Ketika agen menjadi lebih mampu dan lebih otonom, mekanisme pengawasan harus menjadi lebih adaptif. Pembaruan kebijakan, gerbang kemampuan, dan kendala perilaku runtime harus dinamis dan dapat ditegakkan dalam skala besar. Kepercayaan bukanlah fitur bolt-on. Hal ini terus diperkuat melalui arsitektur, perilaku, dan proses. AWS Identity and Access Management (IAM) dan AWS AppConfigmemainkan peran penting dalam menegakkan identitas aman, batas izin runtime, dan perubahan perilaku khusus lingkungan di seluruh agen.

Mengadopsi pola pikir operasi keputusan-pertama

Otomatisasi tradisional berfokus pada efisiensi proses, yang menjalankan skrip atau alur kerja yang telah ditentukan sebelumnya lebih cepat dan lebih andal. Agentic AI, sebaliknya, memperkenalkan otomatisasi keputusan pertama. Agen menilai konteks, menimbang opsi, dan menyesuaikan perilaku secara real-time. Pergeseran dari pola pikir eksekusi-pertama ke keputusan-pertama ini membutuhkan pemikiran baru tentang metrik dan hasil kesuksesan. Alih-alih mengukur keberhasilan secara eksklusif dengan penyelesaian tugas, keberhasilan untuk AI agen diukur dengan seberapa baik keputusan selaras dengan maksud, kebijakan, dan kondisi yang berkembang.

Daripada hanya mengukur penyelesaian tugas atau waktu siklus, organisasi harus mengevaluasi kualitas keputusan time-to-action, dan responsif terhadap perubahan. KPIs harus mencakup metrik seperti:

- Kualitas keputusan — Seberapa baik agen mempersonalisasi responsnya terhadap pengguna atau skenario tertentu? Apakah itu membuat keputusan bernuansa yang selaras dengan tujuan bisnis dan konteks pengguna?
- T ime-to-action — Seberapa cepat dan cerdas agen menilai situasi dan merespons? Apakah latensi cukup rendah untuk terasa adaptif dan seperti manusia?
- Pembongkaran kognitif — Berapa banyak analisis manual, triase, atau pengambilan keputusan rutin yang dapat ditangani agen atas nama manusia? Apakah itu mengurangi usaha atau hanya menggesernya?

Bisnis yang merangkul pola pikir keputusan-pertama dapat menjadi lebih tangguh, adaptif, dan mampu beroperasi pada tingkat kompleksitas yang baru.

Penskalaan dengan tujuan dan niat

Berhasil menskalakan AI agen bukan tentang bereksperimen dengan lebih banyak alat. Ini tentang membangun lapisan intelijen perusahaan yang tahan lama. Ini membutuhkan investasi dalam infrastruktur platform, budaya operasional, kerangka kerja tata kelola, dan penyelarasan strategis. Bisnis harus mengadopsi pendekatan yang disengaja. Mereka harus memperlakukan agen bukan sebagai eksperimen tetapi sebagai komponen inti dari model operasi digital mereka.

Menyelaraskan dengan AWS Well-Architected Framework membantu sistem Anda memenuhi standar perusahaan untuk keandalan, keamanan, efisiensi kinerja, dan pengoptimalan biaya. Alat seperti Strands Agents SDK dapat mempercepat perjalanan ini dengan memberikan petunjuk

terstruktur, pendaftaran alat, dan kesiapan CI/CD. Ini membantu tim beralih dari eksperimen ke pengiriman yang dapat diskalakan dengan menggunakan alur kerja yang sudah dikenal AWS .

Agentic AI bukanlah alat; ini adalah pergeseran dalam bagaimana intelijen tertanam ke dalam operasi. Organizations yang mempersiapkannya dapat mengotomatisasi lebih banyak, beroperasi lebih cerdas, beradaptasi lebih cepat, dan menciptakan keuntungan yang langgeng di dunia yang semakin kompleks.

Kesimpulan untuk mengoperasionalkan AI agen

Agentic AI mewakili lebih dari sekadar pergeseran teknologi. Ini menandai munculnya sistem operasi baru untuk perusahaan. Organisasi-organisasi yang merangkul transformasi ini bergerak melampaui kasus penggunaan otomatisasi yang sempit dan membangun intelijen menjadi fondasi operasi mereka. Pergeseran ini adalah tentang mendesain ulang bagaimana keputusan dibuat, bagaimana sistem beradaptasi, dan bagaimana hasil direalisasikan dalam skala besar.

Di era yang ditentukan oleh meningkatnya kompleksitas, permintaan waktu nyata, dan kelebihan informasi, model otomatisasi skrip tradisional telah mencapai batasnya. Sukses sekarang bergantung pada kemampuan untuk menanamkan kecerdasan langsung ke dalam alur kerja untuk membuat sistem yang memahami, bernalar, bertindak, dan berkembang. Agentic AI dapat menyelaraskan otonomi dengan tujuan, pengambilan keputusan dengan tata kelola, dan kemampuan beradaptasi dengan akuntabilitas.

Transisi ini membutuhkan perpindahan dari eksekusi pertama ke pemikiran keputusan-pertama. Sistem agen tidak hanya mengikuti instruksi. Mereka menafsirkan tujuan, menimbang trade-off, dan mengejar hasil dalam batasan yang ditentukan. Dalam konteks ini, kesuksesan diukur tidak hanya dengan penyelesaian tugas. Hal ini juga diukur dengan kualitas, kelincahan, dan penjelasan keputusan yang dibuat secara real time. Organizations harus memikirkan kembali metrik, insentif, dan desain sistem untuk mendukung agen yang beroperasi secara cerdas di bawah ketidakpastian.

Mengoperasionalkan AI agen bukanlah peningkatan. plug-and-play Ini adalah transformasi arsitektur dan budaya. Ini membutuhkan praktik disiplin di seluruh manajemen siklus hidup, penegakan kepercayaan, interoperabilitas, dan penyelarasan dengan model bisnis. Ini juga menyerukan evolusi model pengiriman, seperti membentuk zona niat, menyematkan pagar pembatas runtime, dan terus menyelaraskan perilaku agen dengan hasil strategis. Tim harus mengadopsi bahasa bersama, kepemilikan bersama, dan akuntabilitas bersama untuk kinerja dan keselamatan agen.

Kesiapan perusahaan dapat menentukan siapa yang tumbuh subur di lingkungan baru ini. Organizations harus berinvestasi dalam pemberdayaan internal, AgentOps kapabilitas, dan kerangka kerja tata kelola yang menskalakan dan menciptakan nilai jangka panjang. Mereka yang berhasil dapat membangun sistem yang lebih cerdas, dan mereka juga dapat membangun bisnis yang lebih adaptif, tangguh, dan didorong oleh wawasan.

Panduan ini meletakkan dasar. Ini menghubungkan strategi untuk eksekusi dan mempersiapkan organisasi untuk membangun platform scalable dari agen cerdas. Seri konten yang lebih luas tentang AI agen AWS memberikan panduan pelengkap. Untuk melihat panduan lain dalam seri ini, lihat

Agentic AI di situs web AWS Prescriptive Guidance. Seri konten ini menawarkan peta jalan untuk mengoperasionalkan otonomi dengan disiplin dan niat.

Untuk memulai, identifikasi ruang keputusan berdampak tinggi di mana agen dapat memberikan peningkatan terukur dalam kecepatan, akurasi, atau daya tanggap. Kemudian gunakan agen pilot terfokus yang memiliki instrumentasi, tata kelola, dan loop umpan balik. Gunakan ini untuk memvalidasi hipotesis nilai, menghasilkan momentum internal, dan membangun kepercayaan dalam pendekatan. Senyawa momentum melalui pembelajaran.

Agentic AI bukanlah tujuan; itu adalah lapisan kemampuan yang berkembang bersama bisnis Anda. Ini mewakili pergeseran jangka panjang menuju intelijen sebagai infrastruktur. Organizations yang memimpin di ruang ini dapat mengotomatisasi lebih banyak, merespons lebih cepat, beradaptasi lebih baik, dan membangun model operasional yang mampu menavigasi kompleksitas pada skala perusahaan.

Sumber daya untuk mengoperasionalkan AI agen

Layanan AWS

Berikut ini Layanan AWS dan fitur dapat membantu Anda membangun dan mengoperasionalkan sistem AI agen di: AWS Cloud

- Amazon API Gateway dapat mengekspos kemampuan agen sebagai skalabel dan menawarkan harga berbasis penggunaan.
- AWS AppConfigmenawarkan manajemen konfigurasi runtime dan pengalihan fitur untuk agen di seluruh penyewa atau lingkungan.
- Amazon Bedrock adalah layanan model dasar yang dapat digunakan agen untuk penalaran, pembuatan, dan eksekusi yang cepat.
- AWS Cloud Development Kit (AWS CDK)adalah infrastruktur sebagai layanan kode yang dapat Anda gunakan untuk menyebarkan dan mengelola tumpukan agen.
- AWS CloudTrailmerekam riwayat peristiwa sehingga Anda dapat melacak aktivitas agen, jejak audit, dan perilaku integrasi.
- Amazon CloudWatch dapat mengelola log, metrik, dan alarm untuk memantau kinerja agen dan perilaku kolaborasi multi-agen.
- AWS CodePipelinemenyediakan CI/CD otomatisasi yang dapat Anda gunakan untuk menguji, memvalidasi, dan menyebarkan kode agen.
- Amazon Cognito adalah layanan identitas yang dapat Anda gunakan untuk mengelola otentikasi pengguna dan penyewa dalam sistem multi-agen.
- AWS Configmenawarkan kepatuhan dan deteksi penyimpangan untuk kebijakan agen dan konfigurasi lingkungan.
- AWS Cost Explorerdapat melacak penggunaan tingkat agen dan membantu menyelaraskan biaya untuk memaksimalkan ROI Anda.
- Amazon DynamoDB adalah layanan penyimpanan yang dapat Anda gunakan untuk memori agen, log peningkatan, dan status kontekstual.
- Amazon Elastic File System (Amazon EFS) adalah sistem file bersama yang dapat Anda gunakan untuk kolaborasi agen atau pemrosesan perantara di seluruh alur kerja.
- Amazon EventBridge adalah bus acara inti yang dapat Anda gunakan untuk merutekan tugas dan mengatur komunikasi di kain agen.

- Amazon EventBridge Pipes dapat merampingkan konsumsi acara dan perutean untuk menghubungkan agen dan layanan.
- Amazon GuardDuty menawarkan deteksi ancaman dan pemantauan anomali yang dapat mendukung eksekusi agen yang aman.
- AWS Identity and Access Management (IAM) membantu Anda menentukan izin berbutir halus untuk eksekusi agen dan akses data.
- AWS Lambdaadalah layanan komputasi stateless yang dapat menjalankan logika agen dan swarm drone.
- AWS Marketplaceadalah platform distribusi eksternal yang dapat Anda gunakan untuk menawarkan kemampuan agen sebagai produk komersial.
- AWS Organizationsadalah layanan tata kelola lintas akun dan penegakan kebijakan yang dapat membantu Anda mengelola infrastruktur agen multi-penyewa.
- AWS Organizations Kebijakan kontrol layanan bertindak sebagai pagar pembatas untuk mengendalikan izin di tingkat akun atau unit organisasi.
- Amazon Quick adalah platform intelijen bisnis (BI) generatif yang didukung AI yang membantu Anda menganalisis data, membuat visualisasi, mengotomatiskan alur kerja, dan berkolaborasi dengan orang lain di seluruh organisasi Anda.
- AWS Resource Access Manager (AWS RAM) dapat membantu Anda berbagi kemampuan antara akun dan layanan agen.
- Amazon SageMaker AI adalah layanan yang dapat Anda gunakan untuk pelatihan model, fine-tuning, dan inferensi di luar model dasar.
- Amazon Simple Storage Service (Amazon S3) menawarkan penyimpanan objek untuk pustaka prompt, artefak model, dan data yang dihasilkan agen.
- AWS Step Functionsadalah mesin alur kerja yang dapat membantu Anda mengoordinasikan aliran multi-agen dan melatih ulang saluran pipa.
- AWS X-Raymenawarkan penelusuran terdistribusi yang dapat Anda gunakan untuk melacak alur keputusan agen dan dependensi layanan.

AWS Sumber daya lainnya

- Yayasan AI agen pada AWS
- Pola dan alur kerja AI agen di AWS
- Kerangka kerja, protokol, dan alat AI Agentic di AWS

- Membangun arsitektur tanpa server untuk AI agen di AWS
- Membangun arsitektur multi-tenant untuk AI agen di 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	—	Agustus 12, 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 — Pindahkan aplikasi dan modifikasi 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 Edition. PostgreSQL-Compatible
- 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 instans 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

A2A () Agent-to-Agent

Protokol stateful untuk kolaborasi agen-ke-agen yang mendukung delegasi tugas dan transfer negara.

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 basis data 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.

Agen

Sistem AI yang dapat secara mandiri bernalar, merencanakan, dan mengambil tindakan menggunakan alat untuk mencapai tujuan.

Agen Ops

Praktik operasional untuk membangun, menguji, menyebarkan, dan menjalankan agen AI dalam produksi dalam skala besar.

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 menyebabkan kerugian bagi 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.

blue/green penyebaran

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. AWS Well-Architected

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

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.

Pengembang Warga

Pengguna bisnis yang membuat aplikasi AI menggunakan platform tanpa code/low kode tanpa keterampilan teknis khusus.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Cloud Center of Excellence (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 CCoE 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 CCoE, 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 AWS Cloud Perusahaan. 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. Satu CI/CD pipa 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 biasanya digambarkan sebagai pipa. CI/CD dapat 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.

pertahanan-mendalam

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, pendekatan defense-in-depth 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 database.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani 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. Big-endian sistem menyimpan byte paling signifikan terlebih dahulu. Little-endian sistem 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 dapat diakses oleh pengguna akhir. Dalam sebuah CI/CD pipeline, 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. Few-shot prompt bisa efektif untuk tugas-tugas yang membutuhkan 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. FM 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.

Gerbang FM

Perantara terpusat yang mengontrol dan menormalkan akses ke model pondasi. Juga dikenal sebagai gateway LLM.

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 (OU). 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 CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

pagar pembatas (AI)

Mekanisme keamanan yang menyaring, memvalidasi, dan membatasi input dan output agen untuk membantu memastikan perilaku AI yang bertanggung jawab dan aman.

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.

manusia-dalam-lingkaran (HiTL)

Pola alur kerja di mana eksekusi agen berhenti untuk peninjauan dan persetujuan manusia pada titik keputusan kritis.

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 in the Framework. AWS Well-Architected

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 VPC masuk, keluar, dan inspeksi 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 selengkapnya, 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 VPC (dalam hal yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi 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 bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat Apa itu LLM.

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.

MCP

Lihat Protokol Konteks Model.

Protokol Konteks Model (MCP)

Protokol stateless untuk komunikasi agen -to- alat.

Server MCP

Layanan yang mengekspos satu atau lebih alat melalui Protokol Konteks Model.

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 selengkapnya, lihat Membangun mekanisme dalam AWS Well-Architected Kerangka Kerja.

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 mesin-ke-mesin (M2M) yang ringan, berdasarkan pola publish/subscribe, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik 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 API ringan. 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 terbaik dan pelajaran yang dipetik 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

Cross-functional tim 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 AWS dengan Layanan Migrasi 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. 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 Mengurai 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 mesin-ke-mesin (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 selengkapnya, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja. AWS Well-Architected

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 `PUT` dan `DELETE` permintaan ke bucket S3.

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 VPC masuk, keluar, dan inspeksi 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 kebijakanberbasis sumber daya), atau menentukan izin maksimum untuk semua akun dalam organisasi di (lihat kebijakan kontrol layanan). AWS Organizations

persistensi 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.

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 pengoptimalan 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 host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau beberapa VPC. 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.

Region

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). Jenis 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 mereferensikan 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 Konsol Manajemen AWS 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 Amazon EC2, 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 SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai 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 oleh 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.

Bayangan AI

Aplikasi AI yang tidak sah dibuat atau digunakan di luar saluran yang diatur dalam suatu organisasi.

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.

model split-and-seed

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 ASP.NET Microsoft (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

Key-value pasangan yang bertindak sebagai metadata untuk mengatur sumber daya Anda AWS . Tag membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai sumber daya AWS.

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.

alat

Fungsi atau API yang dapat dipanggil agen untuk melakukan operasi di sistem eksternal.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan VPC dan jaringan lokal Anda. 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.

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 VPC yang memungkinkan Anda 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 terkini 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.