



Ekonomi untuk AI agen di AWS

AWS Panduan Preskriptif



AWS Panduan Preskriptif: Ekonomi untuk AI agen di AWS

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

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

Table of Contents

Pengantar	1
Audiens yang dituju	1
Tujuan	2
Tentang seri konten ini	2
Memahami ekonomi AI agen	3
Penilaian tugas	3
Penilaian dampak risiko	4
Pengembalian investasi	6
Mengukur kesuksesan dan ROI	7
Gunakan fondasi Anda	7
Tetapkan target	7
Melacak metrik	7
Gunakan AgentOps	7
Menilai biaya proses manusia	8
Biaya tenaga kerja	9
Biaya kinerja	9
Biaya teknologi	10
Biaya peluang	10
Biaya risiko dan cacat	11
Menerapkan sistem AI agen	12
Memasukkan umpan balik manusia	12
Pembelajaran perilaku	13
Pembelajaran berkelanjutan	14
Kolaborasi manusia-AI	14
Harga berbasis hasil	15
Model tradisional, dimuka	15
Model berbasis hasil	15
Menggunakan AWS Marketplace	16
Studi kasus: Operasi rekrutmen	18
Skenario A	18
Struktur biaya dasar	19
Metrik operasional	20
Analisis biaya berbasis volume	21
Analisis ROI	21

Perbandingan biaya kumulatif	21
Manfaat tambahan	22
Skenario B	22
Struktur biaya dasar	23
Metrik operasional	23
Analisis biaya berbasis volume	24
Analisis ROI	24
Perbandingan biaya kumulatif	25
Membandingkan skenario	25
Kesimpulan dan sumber daya	28
Sumber daya	28
Riwayat dokumen	30
Glosarium	31
#	31
A	32
B	35
C	37
D	40
E	44
F	46
G	48
H	49
I	50
L	53
M	54
O	58
P	61
Q	64
R	64
D	67
T	71
U	72
V	73
W	73
Z	74
.....	lxxvi

Ekonomi untuk AI agen di AWS

Hans Schabert dan Prasanta Roy, Amazon Web Services

Januari 2026 ([sejarah dokumen](#))

Organizations yang mengadopsi otomatisasi berbasis AI dan sistem AI agen perlu membuat keputusan ekonomi yang tepat antara tenaga kerja manusia dan agen cerdas. Ini menjadi penting untuk operasi cloud yang berkelanjutan. Panduan ini membantu Anda mengevaluasi, menerapkan, dan mengoptimalkan trade-off ekonomi antara tenaga kerja manusia dan sistem AI agen. AWS Anda dapat memaksimalkan laba atas investasi (ROI) Anda sambil mempertahankan keunggulan operasional.

Tidak ada sistem yang 100% benar. Prinsip dasar ini mendorong analisis ekonomi sistem AI manusia dan agen. Organizations harus bergerak melampaui perbandingan biaya sederhana untuk mengevaluasi dampak ekonomi total, profil risiko, persyaratan kualitas keputusan, dan penciptaan nilai strategis jangka panjang.

Perilaku pelanggan bergeser secara dramatis dari investasi teknologi dimuka tradisional ke pay-per-outcome model yang menyelaraskan biaya dengan hasil bisnis. Transformasi ini membutuhkan pendekatan baru untuk evaluasi, implementasi, dan optimalisasi kolaborasi manusia-agen.

Jalan menuju kesuksesan mengikuti pola yang jelas: mulai dengan pekerjaan yang sesuai, ukur semuanya, dan skala apa yang berhasil. Organizations yang mengadopsi pendekatan ini mencapai keunggulan kompetitif berkelanjutan melalui alokasi sumber daya cerdas dan otomatisasi yang berfokus pada hasil.

Audiens yang dituju

Panduan ini ditujukan untuk hal-hal berikut:

- Eksekutif (CEOs, CTOs, CFOs) yang membuat keputusan investasi strategis
- Arsitek perusahaan yang merancang strategi otomatisasi organisasi
- Praktisi operasi keuangan yang mengoptimalkan manajemen keuangan cloud
- Pemimpin teknologi yang mengevaluasi pendekatan implementasi AI
- Pemimpin unit bisnis yang ingin memahami ROI otomatisasi
- Profesional pengadaan yang menavigasi model harga AI baru

Untuk memahami konsep dalam panduan ini, kami sarankan Anda meninjau [Yayasan AI agen di AWS](#).

Tujuan

Panduan ini membantu Anda memahami hal-hal berikut:

- Bagaimana mengevaluasi pekerjaan untuk potensi otomatisasi agen
- Model ekonomi untuk membandingkan biaya tenaga kerja manusia dengan investasi sistem AI agen
- Pay-per-outcome model penetapan harga dan dampaknya terhadap ekonomi proyek AI
- Teknik pengukuran untuk mendemonstrasikan ROI dan mengelola risiko
- Strategi penskalaan yang mengubah biaya tetap menjadi hasil variabel

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.

Memahami ekonomi AI agen pada AWS

Salah satu prinsip utamanya adalah menentukan kapan harus menggunakan agen AI dan kapan harus menggunakan metode deterministik tradisional. Organizations harus secara sistematis mengevaluasi pekerjaan mana yang memerlukan otomatisasi agen dan mana yang harus menggunakan otomatisasi tradisional atau operasi manusia yang berkelanjutan. Keputusan ini membutuhkan pemahaman hubungan antara karakteristik tugas, toleransi risiko, dan pendekatan operasional.

Sebelum memutuskan untuk menerapkan AI agen, Anda harus menggunakan kerangka keputusan untuk memahami dampak ekonomi. Kerangka keputusan mencakup tiga pertanyaan kunci berikut:

1. [Penilaian tugas](#) — Apakah tugas ini tepat untuk agen AI?
2. [Penilaian dampak risiko](#) — Apa risiko yang terlibat?
3. [Pengembalian investasi](#) — Apakah akan hemat biaya?

Penilaian tugas

Tugas dengan kompleksitas tinggi, aturan keputusan standar dapat mengambil manfaat dari pendekatan AI agen. Tugas sederhana yang sangat standar lebih baik dilayani oleh otomatisasi tradisional atau otomatisasi proses robot. Sistem AI agen unggul dalam penalaran, memahami konteks, atau membuat keputusan secara adaptif. Mereka menambah nilai di luar pemrosesan berbasis aturan. Implementasi AI agen yang berhasil membutuhkan sistem yang mampu belajar dan beradaptasi.

Pertimbangkan faktor-faktor berikut saat mengevaluasi tugas:

- Kompleksitas — Tingkat penalaran dan pemahaman konteks yang diperlukan. Tugas yang membutuhkan pemahaman kontekstual, interpretasi bernuansa, atau respons adaptif terhadap perubahan kondisi mendukung pendekatan agen daripada otomatisasi tradisional, sementara tugas mekanis atau kalkulasi murni mungkin tidak memerlukan kecerdasan agen.
- Standardisasi — Adanya pola dan aturan yang jelas. Agentic AI direkomendasikan jika tugas tersebut membutuhkan pemahaman kontekstual. Jika tidak diperlukan adaptasi atau pembelajaran, pertimbangkan otomatisasi tradisional.
- Volume — Frekuensi kinerja tugas. Agentic AI direkomendasikan untuk kegiatan otonom. Otomatisasi tradisional direkomendasikan untuk tugas volume tinggi dan konsisten. Namun,

volume saja tidak menentukan pendekatan. Keputusan volume rendah dan bernilai tinggi mungkin membenarkan bantuan agen untuk meningkatkan kualitas keputusan daripada pengurangan biaya.

- Nilai — Dampak bisnis per penyelesaian tugas. Pertimbangkan AI agen untuk hasil bernilai tinggi yang membutuhkan kemampuan otonom seperti manusia. Pertimbangkan otomatisasi tradisional untuk tugas berulang dan konsisten, yang dapat dilakukan dengan cara deterministik.

Penilaian dampak risiko

Saat ini ada empat pendekatan penyebaran AI agen: sepenuhnya otonom, human in the loop, co-pilot, atau dipimpin manusia dengan dukungan agen. Masing-masing memiliki profil risiko dan toleransi kesalahan mereka sendiri, dan mereka semua melibatkan manusia dalam kapasitas tertentu. Tabel berikut menjelaskan rincian risiko dari pendekatan ini.

Tingkat otonomi	Profil risiko	Toleransi kesalahan	Contoh kasus penggunaan	Keterlibatan manusia
Sepenuhnya otonom	Risiko Rendah	1-2% dapat diterima	<ul style="list-style-type: none"> • Kategorisasi data dasar • Perutean dokumen • Pembuatan laporan standar 	<ul style="list-style-type: none"> • Pengawasan minimal • Audit berkala
Manusia dalam lingkaran	Risiko Sedang	Di bawah 0,5%	<ul style="list-style-type: none"> • Tanggapan draf • Moderasi konten • Pemrosesan klaim awal 	<ul style="list-style-type: none"> • Ulasan reguler • Penanganan pengecualian • Jaminan kualitas

Kopilot	Resiko tinggi	Mendekat-nol	<ul style="list-style-type: none"> Masukan perencanaan strategis Penilaian risiko Keputusan investasi 	<ul style="list-style-type: none"> Manusia membuat keputusan akhir Agen memberikan rekomendasi
Dipimpin manusia dengan dukungan agen	Risiko Kritis	Toleransi nol	<ul style="list-style-type: none"> Keputusan hukum Diagnosis medis Kepatuhan terhadap peraturan 	<ul style="list-style-type: none"> Proses drive manusia Agen menyediakan penelitian atau analisis dan informasi pendukung saja

Tabel berikut menjelaskan pertimbangan utama ketika memilih antara pendekatan ini.

Pertimbangan	Sepenuhnya otonom	Manusia dalam lingkaran	Kopilot	Dipimpin manusia
Efisiensi biaya	Tertinggi	Tinggi	Sedang	Rendah
Skalabilitas	Tidak terbatas.	Tinggi	Sedang	Terbatas
Kecepatan pemrosesan	Tercepat	Cepat	Sedang	Lambat
Manajemen risiko	Basic	Ditingkatkan	Kuat	Terkuat

Penanganan kompleksitas	Tugas sederhana	Tugas yang cukup kompleks	Tugas kompleks	Tugas-tugas penting
-------------------------	-----------------	---------------------------	----------------	---------------------

Kerangka pertimbangan ini membantu organisasi mencocokkan tingkat otonomi dengan profil risiko, skala operasi dengan tepat, menyeimbangkan efisiensi dengan kontrol, menerapkan tata kelola yang tepat, dan mengoptimalkan alokasi sumber daya.

Pengembalian investasi

Menghitung laba atas investasi untuk sistem AI agen dimulai dengan analisis biaya yang komprehensif. Organizations harus terlebih dahulu menghitung biaya manusia mereka saat ini, termasuk gaji, tunjangan, dan biaya ruang kerja, bersama dengan biaya khusus proses dan biaya tersembunyi seperti pelatihan, cakupan, dan downtime.

Untuk analisis impas, organisasi harus mempertimbangkan biaya implementasi, biaya operasional yang sedang berlangsung, dan volume yang diperlukan untuk membenarkan investasi. Penting juga untuk memperhitungkan variasi musiman dan manfaat kurva pembelajaran yang muncul saat sistem matang dan meningkat seiring waktu.

Ketika mengevaluasi agen AI, organisasi harus ingat bahwa sistem ini biasanya memiliki biaya di muka yang lebih tinggi tetapi biaya per transaksi yang lebih rendah dibandingkan dengan operasi manusia. Selain itu, agen AI menunjukkan peningkatan kinerja dari waktu ke waktu dan menawarkan skalabilitas yang lebih baik daripada tim manusia. Hal ini membuat mereka semakin hemat biaya karena skala penyebaran dan pengalaman operasional terakumulasi.

Mengukur keberhasilan dan ROI sistem AI agen

Mengukur keberhasilan dalam implementasi sistem AI agen membutuhkan pendekatan yang sistematis. Bagian ini memberikan metodologi yang jelas untuk evaluasi dan optimasi berkelanjutan yang menggunakan analisis Anda yang ada daripada memulai dari awal.

Langkah 1: Gunakan fondasi Anda yang ada

Mulailah dengan penilaian biaya yang komprehensif sesuai dengan rekomendasi di bagian [Menilai biaya proses Anda saat ini](#). Ini memberikan dasar operasional untuk perhitungan ROI Anda. Seperti yang dijelaskan di bagian [Penilaian dampak risiko](#), pilih di antara empat tingkat otonomi (sepenuhnya otonom, manusia dalam lingkaran, pendekatan co-pilot, dipimpin manusia dengan dukungan agen) untuk menentukan kriteria pengukuran yang tepat dan ambang toleransi kesalahan untuk setiap proses.

Langkah 2: Tetapkan target sukses yang jelas

Tetapkan arsitektur dan target keberhasilan yang menekankan sistem berkemampuan belajar, seperti yang dijelaskan dalam bagian [Pola sukses untuk menerapkan sistem AI agen](#). Fokus pada peningkatan berkelanjutan daripada kinerja statis. Tetapkan jadwal ROI dengan menggunakan metodologi analisis impas yang ditunjukkan dalam [Studi kasus: Membandingkan biaya AI manusia dan agen](#) untuk operasi rekrutmen. Sertakan poin keputusan yang jelas untuk menghentikan agen yang tidak berkinerja.

Langkah 3: Lacak metrik kunci

Pantau kinerja keuangan berdasarkan garis dasar yang telah Anda tetapkan, dan lacak penghematan biaya dan peningkatan nilai strategis. Ukur metrik operasional, termasuk tingkat kesalahan dalam ambang batas yang dapat diterima untuk tingkat otonomi yang Anda pilih, peningkatan kecepatan pemrosesan, dan peningkatan konsistensi. Fokus pada indikator strategis yang menunjukkan kemampuan belajar dan adaptasi dari waktu ke waktu.

Langkah 4: Gunakan AgentOps

Terapkan kerangka kerja pembelajaran berkelanjutan dari bagian [Memasukkan umpan balik manusia ke dalam sistem AI agen untuk](#) mengoptimalkan pengambilan keputusan melalui integrasi umpan

balik manusia yang sistematis. Buat sistem pembelajaran real-time yang menggabungkan wawasan manusia untuk peningkatan kinerja. Pantau transformasi menuju model bisnis berbasis hasil seperti yang dijelaskan dalam [Transformasi ekonomi ke harga berbasis hasil untuk sistem AI](#) agen. AWS

Menilai biaya proses manusia Anda saat ini

Memahami biaya proses Anda yang sebenarnya sangat penting untuk membuat keputusan berdasarkan informasi tentang investasi sistem AI agen. Pertama, Anda harus menetapkan dasar yang akurat tentang biaya proses Anda saat ini, termasuk semua biaya tersembunyi, tingkat kegagalan, dan biaya peluang. Ini membantu Anda mengembangkan perhitungan ROI yang tepat dan membuat keputusan strategis. Penilaian biaya komprehensif ini berfungsi sebagai landasan penting untuk mengevaluasi apakah sistem AI agen dapat memberikan nilai asli sebagai pendamping produktif.

Penilaian biaya dasar sangat penting untuk alasan kunci berikut:

- Akurasi ROI — Garis dasar biaya yang akurat mendukung proyeksi ROI realistis yang memperhitungkan spektrum penuh dari biaya operasional saat ini.
- Strategi implementasi agen — Pemahaman biaya yang komprehensif membantu organisasi mengidentifikasi proses yang paling menjanjikan untuk penerapan sistem AI agen awal.
- Pengukuran kinerja - Garis dasar yang ditetapkan menyediakan kerangka pengukuran untuk melacak manfaat aktual dan yang diproyeksikan dari implementasi AI agen.

Organizations harus secara sistematis mengidentifikasi dan mengevaluasi semua faktor biaya yang mempengaruhi ekonomi proses sebelum membandingkan alternatif manusia dan agen. Penilaian ini memastikan perhitungan dasar yang akurat dengan memperhitungkan penggerak biaya yang jelas dan tersembunyi. Ini menempatkan penekanan khusus pada biaya kegagalan, tingkat kegagalan historis, dan peluang bisnis yang terlewatkan yang mewakili total biaya sebenarnya dari proses saat ini.

Bagian ini menjelaskan cara mengumpulkan data di setiap kategori biaya untuk menetapkan pengukuran dasar yang akurat untuk proses Anda saat ini. Ini membahas sumber informasi dan memberikan contoh untuk kategori biaya berikut:

- [Biaya tenaga kerja](#)
- [Kinerja manusia dan biaya konsistensi](#)
- [Biaya teknologi dan infrastruktur](#)

- [Kehilangan biaya peluang bisnis](#)
- [Biaya risiko dan cacat](#)

Biaya tenaga kerja

Ekstrak 24 bulan data penggajian yang mencakup gaji pokok, lembur, tunjangan, dan biaya pelatihan. Gunakan sistem informasi sumber daya manusia (HRIS) Anda untuk melacak biaya rekrutmen dan tingkat turnover. Sistem pelacakan waktu mengungkapkan produktivitas aktual versus jam yang dijadwalkan. Platform manajemen kinerja menunjukkan korelasi antara tingkat keterampilan dan biaya kompensasi. Hitung tarif per jam yang terisi penuh yang dialokasikan untuk overhead manajemen.

Berikut ini adalah contoh daftar driver biaya untuk tenaga kerja.

Pengemudi biaya	Dampak bisnis
Kompensasi dasar	\$25—150 per jam terisi penuh
Manfaat dan pajak penggajian	25— 40% dari gaji pokok
Pelatihan dan pengembangan	5-15% dari biaya tenaga kerja tahunan
Overhead manajemen	15-25% dari biaya tenaga kerja langsung

Kinerja manusia dan biaya konsistensi

Menggabungkan data dari sistem manajemen proyek yang menunjukkan variasi penyelesaian tugas dengan sistem kehadiran. Ini dapat mengungkapkan pola absensi dan perubahan musiman. Platform layanan pelanggan menunjukkan rentang kinerja individu melalui metrik resolusi, dan data manajemen hubungan pelanggan penjualan (CRM) dapat menunjukkan variasi efisiensi dalam penutupan kesepakatan. Sistem manajemen mutu memberikan tingkat cacat dan memproses data kepatuhan di seluruh tim dan lokasi. Sistem alur kerja menangkap waktu penyelesaian, penundaan persetujuan, dan frekuensi penanganan pengecualian. Analisis komunikasi mengungkapkan overhead koordinasi melalui frekuensi pertemuan dan pola kolaborasi.

Berikut ini adalah contoh daftar driver biaya untuk kinerja dan konsistensi manusia.

Pengemudi biaya	Dampak bisnis
Fluktuasi produktivitas	20— 50% rentang kinerja
Ketidakhadiran dan cakupan	15— 25% kapasitas tambahan yang dibutuhkan
Siklus kelelahan dan motivasi	Varians produktivitas 10-30%
Ketidakkonsistenan prosedur	Kehilangan efisiensi 10-40%
Variasi kontrol kualitas	10-30% dari total biaya
Overhead koordinasi	15-25% dari biaya operasional

Biaya teknologi dan infrastruktur

Platform manajemen lisensi menunjukkan biaya perangkat lunak dan tingkat pemanfaatan. Pemantauan infrastruktur menyediakan data uptime, metrik kinerja, dan biaya pemeliharaan. Sistem help desk melacak dukungan overhead dan masalah teknis yang berulang. Sistem manajemen vendor menangkap total biaya hubungan teknologi, termasuk biaya integrasi dan kinerja tingkat layanan.

Berikut ini adalah contoh daftar driver biaya untuk teknologi dan infrastruktur.

Pengemudi biaya	Dampak bisnis
Sistem teknologi	\$50—500 per pengguna per bulan
Ruang kerja dan peralatan	\$200—1.000 per karyawan per bulan

Kehilangan biaya peluang bisnis

Platform CRM berisi waktu respons prospek, tingkat konversi, dan dokumentasi peluang yang hilang. Otomatisasi pemasaran menunjukkan dampak penundaan tindak lanjut pada konversi prospek. Sistem dukungan pelanggan mengungkapkan bagaimana masalah operasional mempengaruhi kepuasan dan retensi. Analisis kompetitif memberikan persyaratan respons pasar dan data menang atau rugi yang menghubungkan kinerja operasional dengan hasil pendapatan.

Berikut ini adalah contoh daftar driver biaya untuk peluang bisnis yang hilang.

Pengemudi biaya	Dampak bisnis
Penundaan respons pasar	Pendapatan per hari keterlambatan
Kendala kapasitas	Kehilangan peluang bisnis
Alokasi sumber daya inovasi	Biaya peluang pekerjaan rutin
Penundaan akuisisi pelanggan	Kehilangan timbal 50-90% dari respons lambat

Biaya risiko dan cacat

Dokumentasi polis asuransi menunjukkan biaya untuk kewajiban umum, tanggung jawab profesional, kompensasi pekerja, dan pertanggungjawaban kewajiban dunia maya. Laporan penilaian risiko internal mengidentifikasi kerentanan operasional dan biaya mitigasi terkait. Sistem pelacakan cacat mendokumentasikan kegagalan produk atau layanan, termasuk biaya deteksi, biaya penggantian, dan klaim garansi. Jadwal penggantian aset menunjukkan tingkat kegagalan peralatan dan biaya penggantian. Laporan insiden keselamatan melacak kecelakaan di tempat kerja dan klaim kompensasi pekerja terkait. Rencana kelangsungan bisnis merinci biaya sistem cadangan dan investasi pemulihan bencana.

Berikut ini adalah contoh daftar driver biaya untuk risiko dan cacat.

Pengemudi biaya	Dampak bisnis
Biaya asuransi	1-5% dari anggaran operasional
Biaya kesalahan	\$50—5.000 per insiden kesalahan
Dampak kesalahan manusia	2-15% dari total biaya operasional
Tingkat kesalahan dan pengerjaan ulang	1,5—4 kali biaya asli untuk koreksi

Pola yang berhasil untuk mengimplementasikan sistem AI agen AWS

[State of Enterprise AI Adoption](#) (laporan ISG 2025) mengungkapkan bahwa hambatan utama keberhasilan implementasi AI bukanlah kemampuan teknis tetapi kesenjangan pembelajaran. Istilah ini mengacu pada sistem yang tidak dapat beradaptasi, mengingat konteks, atau meningkatkan dari waktu ke waktu. Organizations yang menerapkan alat AI statis melihat tingkat kegagalan yang tinggi. Berikut ini adalah karakteristik umum dari sistem AI agen yang mencapai kesuksesan:

- Memori kontekstual — Sistem yang menyimpan riwayat percakapan dan preferensi pengguna
- Integrasi umpan balik — Kemampuan untuk belajar dari koreksi dan meningkatkan kinerja
- Adaptasi alur kerja - Penyesuaian otomatis untuk mengubah persyaratan bisnis
- Perbaikan berkelanjutan — Peningkatan terukur melalui pengalaman operasional

Organizations yang mencapai implementasi AI yang sukses sering memprioritaskan hal-hal berikut:

- Menggunakan ekosistem mitra yang komprehensif daripada membangun dan mengeksplorasi kemampuan AI secara mandiri
- Sistem berkemampuan belajar melalui alat statis
- Fokus hasil bisnis atas perbandingan fitur teknis
- Integrasi alur kerja bukan alat mandiri
- Adaptasi berkelanjutan daripada implementasi satu kali

[Pola-pola ini selaras dengan banyak Layanan AWS kemampuan, terutama akses model fondasi di Amazon Bedrock, arsitektur berbasis peristiwa di AWS Lambda, dan pemantauan komprehensif yang ditawarkan melalui Amazon CloudWatch](#) Untuk informasi lebih lanjut tentang mengintegrasikan umpan balik manusia dan sistem yang mampu belajar, lihat bagian [Memasukkan umpan balik manusia ke dalam sistem AI agen dalam panduan ini](#).

Memasukkan umpan balik manusia ke dalam sistem AI agen

Tidak ada sistem yang 100% berhasil, dan kegagalan pasti akan terjadi. Dengan setiap kegagalan, ada biaya perubahan yang terkait. Human in the loop adalah pendekatan AI di mana AI melakukan tugas, tetapi intervensi atau persetujuan manusia diperlukan. Pendekatan ini harus digunakan ketika biaya kegagalan lebih tinggi daripada biaya memiliki human-in-the-loop solusi.

Keberhasilan sistem AI agen sangat bergantung pada kemampuan agen untuk belajar dan meningkatkan melalui umpan balik manusia. Biaya usaha manusia harus dipertimbangkan, tergantung pada tingkat upaya yang diperlukan. Tidak seperti alat otomatisasi statis yang menjalankan aturan yang telah ditentukan, human-in-the-loop solusi memiliki sistem agen yang mampu belajar yang menciptakan kemitraan dinamis antara agen otonom dan manusia. Keahlian manusia terus meningkatkan kinerja agen sementara agen menangani pemrosesan rutin dalam skala besar. Pendekatan kolaboratif ini mengubah implementasi AI dari penerapan satu kali menjadi proses pengoptimalan yang berkelanjutan. Sistem ini beradaptasi dengan pola organisasi, menginternalisasi standar kualitas, dan menyempurnakan kemampuan pengambilan keputusan berdasarkan pengalaman operasional dunia nyata. Dengan secara sistematis menangkap koreksi, persetujuan, dan wawasan manusia, organisasi dapat membangun agen AI yang memahami konteks, mengenali pola, dan semakin selaras dengan tujuan bisnis dari waktu ke waktu.

Untuk solusi yang tidak memerlukan intervensi atau dukungan manusia, tidak perlu memasukkan biaya spesifik manusia ke dalam ekonomi agen.

Pembelajaran perilaku dari operator manusia

Operator manusia memberikan umpan balik penting yang dapat digunakan sistem AI agen untuk belajar, beradaptasi, dan meningkatkan respons mereka dari waktu ke waktu. Loop umpan balik ini menciptakan lingkungan kolaboratif di mana keahlian manusia meningkatkan kemampuan agen sementara agen menangani pemrosesan rutin.

Melalui pengenalan pola perilaku manusia, agen belajar dari pola interaksi manusia untuk mencerminkan pendekatan komunikasi yang sukses. Ini membantu mereka beradaptasi dengan pola keputusan organisasi dan tingkat toleransi risiko. Sistem menginternalisasi harapan kualitas melalui koreksi dan persetujuan manusia. Mereka juga dapat mempelajari tanggapan yang tepat untuk segmen pelanggan dan konteks bisnis yang berbeda.

Mekanisme pengumpulan umpan balik yang efektif secara sistematis menangkap pengeditan dan modifikasi manusia terhadap respons agen. Mereka menganalisis apa yang disetujui, ditolak, atau dimodifikasi oleh pengulas manusia dalam rekomendasi agen. Dengan memahami mengapa kasus-kasus tertentu memerlukan intervensi manusia dan menggabungkan evaluasi manusia terhadap kinerja agen di berbagai skenario dan tingkat kompleksitas, sistem ini terus menyempurnakan kemampuan mereka untuk menyelaraskan lebih dekat dengan standar dan harapan organisasi.

Operasi pembelajaran berkelanjutan

Integrasi pembelajaran real-time memungkinkan sistem AI agen untuk menggabungkan umpan balik manusia dan meningkatkan respons agen segera melalui pembaruan model dinamis. Sistem ini menggunakan wawasan manusia untuk mengidentifikasi pola baru dan kasus tepi. Ini meningkatkan kemampuan pengenalan pola mereka sambil membangun memori organisasi melalui pengalaman belajar yang dipandu manusia. Penyempurnaan berkelanjutan berdasarkan umpan balik manusia-operator dan hasil bisnis mendorong optimalisasi kinerja yang berkelanjutan.

Pelatihan yang dipandu manusia menangkap pengetahuan ahli untuk meningkatkan kemampuan pengambilan keputusan agen. Ini mentransfer keahlian penting dari operator berpengalaman ke sistem AI. Melalui pembelajaran berbasis skenario, sistem menggunakan contoh buatan manusia untuk meningkatkan penanganan situasi yang kompleks. Mereka juga menyelaraskan standar kinerja agen dengan harapan kualitas manusia melalui kalibrasi kualitas. Pendekatan ini menggabungkan wawasan manusia tentang budaya organisasi dan harapan pelanggan. Adaptasi budaya ini membantu agen merespons dengan tepat di berbagai konteks.

Keunggulan operasional dengan kolaborasi manusia-AI

Optimalisasi sadar risiko otomatis memungkinkan evaluasi berkelanjutan terhadap kondisi operasi dan probabilitas kesalahan dengan pengawasan manusia untuk skenario berisiko tinggi. Ini membantu sistem belajar dari penilaian risiko manusia dan meningkatkan pengambilan keputusan di masa depan. [Amazon Bedrock](#) menyediakan akses ke beberapa model pondasi dengan kemampuan dan profil biaya yang berbeda. Hal ini memungkinkan perutean cerdas yang mempertimbangkan profil biaya dan risiko sambil menggabungkan umpan balik manusia untuk mengoptimalkan pemilihan model. Penyetelan kinerja menyeimbangkan efisiensi dengan minimalisasi tingkat kesalahan dengan menggabungkan umpan balik manusia pada standar kualitas dan pertukaran kinerja yang dapat diterima. Keputusan otomatis mempertimbangkan total biaya kepemilikan yang disesuaikan dengan risiko. Operator memberikan panduan tentang toleransi risiko organisasi dan pembobotan prioritas bisnis. Ini membantu Anda mengoptimalkan biaya sambil menyelaraskan dengan tujuan organisasi.

Sistem pembelajaran yang ditingkatkan manusia memprioritaskan masukan manusia dengan dampak kesalahan dan konsekuensi bisnis. Ini menciptakan sistem pembelajaran yang memahami akurasi teknis dan konteks bisnis melalui umpan balik berbobot risiko. Analisis kinerja reguler menggabungkan metrik risiko dan analisis biaya kesalahan, dengan wawasan manusia memberikan konteks yang tidak dapat ditangkap oleh sistem otomatis. Pengembangan praktik terbaik menekankan manajemen risiko dan pencegahan kesalahan dengan menggabungkan pengenalan pola otomatis dengan keahlian dan penilaian manusia. Pembangunan kemampuan organisasi melalui

program pelatihan mengembangkan keterampilan manusia untuk mengelola sistem AI agen dan kemampuan agen untuk mendukung pengambilan keputusan manusia. Ini memastikan pendekatan komprehensif untuk kolaborasi manusia-AI yang memperkuat kedua komponen kemitraan.

Transformasi ekonomi ke penetapan harga berbasis hasil untuk sistem AI agen di AWS

Pergeseran dari model biaya tetap tradisional ke harga berbasis hasil merupakan transformasi mendasar dalam bagaimana organisasi menyusun operasi ekonomi mereka dan mengelola risiko. Transformasi ini berfungsi sebagai jalur untuk modernisasi konstan proses yang ada sambil membiayai transformasi AI agen. Ini memungkinkan organisasi untuk berevolusi dari operasi statis, intensif sumber daya menjadi model bisnis yang dinamis dan didorong oleh hasil.

Model tradisional, dimuka

Departemen sering beroperasi sebagai pusat biaya dengan biaya tenaga kerja langsung yang dibiayai alokasi biaya. Organizations biasanya ingin mengurangi alokasi biaya ini. Jika prosesnya tidak dimodernisasi, departemen harus memberikan hasil yang sama dengan tenaga kerja yang lebih kecil. Ini biasanya menurunkan kualitas. Model bisnis tradisional menciptakan tantangan yang signifikan, termasuk:

- Penskalaan biaya linier dengan peningkatan volume — Ini mengharuskan organisasi untuk mempekerjakan staf tambahan untuk menangani peningkatan volume.
- Komitmen biaya tetap — Ini bertahan terlepas dari kinerja bisnis dan efisiensi proses.
- Perencanaan lanjutan — Fleksibilitas terbatas selama kemerosotan ekonomi dan kendala kapasitas memerlukan perencanaan awal.
- Siklus degradasi kualitas - Pengurangan anggaran menyebabkan penurunan kualitas layanan ketika biaya dipotong tanpa perbaikan proses.

Model berbasis hasil

Model berbasis hasil modern mengikat pembayaran langsung dengan hasil bisnis yang terukur, seperti perekrutan yang berhasil diselesaikan, metrik kualitas yang dicapai, peningkatan efisiensi proses, atau peningkatan produktivitas yang direalisasikan. Ini secara fundamental menggeser risiko keuangan dari unit bisnis ke penyedia layanan sambil menciptakan keselarasan insentif alami. Berikut ini adalah manfaat utama dari model berbasis hasil:

- Skala biaya langsung dengan nilai bisnis yang dihasilkan
- Keselarasan alami antara biaya operasional dan pendapatan
- Fleksibilitas untuk menyesuaikan kapasitas berdasarkan kondisi pasar
- Pay-per-success model mengurangi risiko keuangan dengan mengalihkan eksposur keuangan dari investasi di muka ke kinerja operasional yang berkelanjutan
- Fokus pada sistem kemampuan belajar yang meningkat dari waktu ke waktu, bukan alternatif statis

Transformasi ini jauh melampaui pusat biaya internal untuk secara fundamental membentuk kembali bagaimana organisasi terlibat dengan mitra eksternal dan penyedia layanan. Dengan menerapkan harga berbasis hasil untuk kolaborasi mitra, organisasi dapat mendorong peningkatan kualitas jangka panjang dan mengurangi biaya sambil secara tidak langsung menekankan modernisasi AI agen.

Organizations dapat bereksperimen dengan cepat, mengukur kinerja dengan jelas, dan skala berdasarkan nilai bisnis aktual yang dihasilkan daripada komitmen sumber daya tetap tradisional. Pendekatan ini memungkinkan yang berikut:

- Evolusi hubungan vendor — Mitra menjadi berinvestasi dalam kesuksesan pelanggan, bukan hanya pengiriman layanan.
- Metrik hasil standar — Menyederhanakan proses pengadaan di beberapa penyedia.
- Responsif pasar — Cepat beradaptasi dengan perubahan kondisi pasar dan kebutuhan pelanggan.
- Keunggulan kompetitif - Pemanfaatan sumber daya yang unggul dan kemampuan operasional yang ditingkatkan.
- Kemitraan berbasis kualitas - Kolaborasi jangka panjang berfokus pada peningkatan berkelanjutan dan hasil yang terukur.

Menggunakan AWS Marketplace sebagai pay-per-outcome enabler

Enabler utama untuk transformasi ini adalah [AWS Marketplace](#), yang berfungsi sebagai kendaraan transaksi untuk pekerjaan agen dan penetapan harga berbasis hasil. Ini menyediakan akses ke ratusan agen AI pra-bangun dan solusi agen dengan model harga berbasis penggunaan yang transparan. Ini dapat membantu menghilangkan biaya lisensi di muka, mengurangi kompleksitas implementasi, dan memungkinkan organisasi untuk fokus pada sistem berkemampuan belajar yang beradaptasi dan meningkat dari waktu ke waktu daripada alternatif statis.

Menggunakan AWS Marketplace dapat memberikan manfaat sebagai berikut:

- Eksperimen cepat - Uji beberapa solusi tanpa investasi modal yang signifikan
- Harga transparan — Biaya berbasis penggunaan dengan atribusi yang jelas terhadap hasil bisnis
- Solusi yang terbukti - Akses ke agen yang telah teruji pertempuran dari penyedia berpengalaman
- Integrasi bawaan - Konektivitas mulus dengan yang ada Layanan AWS
- Mitigasi risiko - Kemampuan untuk beralih penyedia berdasarkan kinerja
- Akses kemampuan belajar — Ketersediaan sistem adaptif tanpa biaya pengembangan internal

Pendekatan ini memungkinkan organisasi untuk membandingkan beberapa opsi berdasarkan penyampaian hasil dan kemampuan belajar daripada daftar fitur. Ini juga dapat membantu Anda menetapkan kriteria keberhasilan dan metodologi pengukuran yang jelas dan menegosiasikan harga berbasis hasil yang terkait dengan hasil bisnis dan peningkatan sistem. Dengan membiayai transformasi AI agen melalui model berbasis hasil, organisasi dapat memodernisasi proses mereka secara terus menerus sambil hanya membayar untuk perbaikan yang terukur dan hasil yang sukses.

Studi kasus: Membandingkan biaya AI manusia dan agen untuk operasi rekrutmen

Operasi rekrutmen memberikan studi kasus yang menarik untuk mengevaluasi pertukaran ekonomi antara sistem AI manusia dan agen, tetapi perhitungan ROI sangat bergantung pada baseline operasional Anda saat ini. Organizations yang mengevaluasi investasi AI agen sering mengajukan pertanyaan mendasar: “Bagaimana jika kita hanya mengoptimalkan proses manusia yang ada?” Untuk mengatasi hal ini secara langsung, analisis ini menyajikan dua skenario berbeda yang mencakup jangkauan efisiensi operasional manusia.

[Skenario A](#) memodelkan riwayat hidup 45 menit (CV) atau melanjutkan waktu penyaringan. [Skenario B](#) menunjukkan operasi manusia yang dioptimalkan pada 15 menit per aplikasi, yang merupakan peningkatan efisiensi 66%. Misalnya, peningkatan ini dapat dicapai melalui proses yang efisien, perekrut berpengalaman, atau alat khusus.

Dengan membandingkan kemampuan sistem agen yang identik dengan garis dasar kinerja manusia yang berbeda ini, kami mengungkapkan bagaimana efisiensi proses yang ada berdampak pada perhitungan ROI, jadwal impas, dan keputusan implementasi strategis. Pendekatan skenario ganda ini melayani berbagai tujuan. Ini mencegah organisasi menolak AI agen dengan mengasumsikan optimasi proses saja sudah cukup. Ini juga membantu organisasi dengan proses yang sudah efisien memahami ekonomi spesifik mereka. Selain itu, skenario ini menyoroti ketika keuntungan non-finansial, seperti ketersediaan dan skalabilitas 24/7, menjadi faktor keputusan utama. Memahami dinamika ekonomi ini di seluruh baseline efisiensi yang berbeda memungkinkan organisasi untuk membuat keputusan berdasarkan informasi tentang di mana dan kapan harus menerapkan sistem AI agen untuk dampak bisnis yang maksimal.

Skenario A: Waktu skrining 45 menit

Skenario A mewakili operasi rekrutmen di mana perekrut manusia menghabiskan 45 menit untuk menyaring setiap resume. Skenario ini memodelkan perekrut tingkat menengah dengan biaya penuh tahunan sebesar \$112.250. Perekrut ini memproses aplikasi selama jam kerja standar dengan karakteristik kinerja manusia yang khas. Sebaliknya, sistem AI agen memerlukan investasi awal sebesar \$23.000 untuk pengembangan, penyesuaian, dan integrasi ATS, dan memiliki biaya operasi bulanan minimal \$500 untuk infrastruktur cloud. Agen memproses aplikasi hanya dalam 5 menit dengan ketersediaan 24/7, mencapai tingkat kesalahan 2% dan kapasitas bulanan melebihi 8.600 aplikasi. Ini adalah kesenjangan efisiensi yang dramatis, di mana agen beroperasi 9 kali lebih cepat

per aplikasi dan kapasitas bulanan 39 kali lebih besar. Bagian ini membahas analisis struktur biaya, metrik operasional, perbandingan berbasis volume, dan perhitungan ROI kumulatif selama enam bulan pertama operasi.

Struktur biaya dasar

Tabel berikut menunjukkan biaya persiapan awal untuk skenario A.

Komponen	Operasi manusia	Sistem AI agen
Pengembangan dan kustomisasi agen	N/A	\$15.000
Integrasi sistem pelacakan pemohon (ATS)	N/A	\$5.000
Pelatihan dan optimasi	N/A	\$3.000
Total biaya pengaturan awal	\$0	\$23.000

Tabel berikut menunjukkan biaya tetap tahunan untuk skenario A.

Komponen	Operasi manusia	Sistem AI agen
Gaji pokok	\$65.000	N/A
Manfaat (30%)	\$19.500	N/A
Ruang kerja dan peralatan	\$12.000	N/A
Pengawasan manajemen (15%)	\$9.750	N/A
Pelatihan dan pengembangan	\$6.000	N/A
Total biaya tetap tahunan	\$112.250	N/A

Tabel berikut menunjukkan biaya operasi bulanan untuk skenario A.

Komponen	Operasi manusia	Sistem AI agen
Komputasi awan	N/A	\$200
Penyimpanan	N/A	\$100
Operasi basis data	N/A	\$100
Memantau	N/A	\$100
Total biaya tetap bulanan	\$9.354	\$500

Metrik operasional

Tabel berikut menunjukkan metrik operasional untuk skenario A.

Metrik	Operasi manusia	Sistem AI agen
Waktu pemrosesan per aplikasi	45 menit	5 menit
Kapasitas per jam	1.33 aplikasi	12 aplikasi
Kapasitas harian (24 jam)	10-11 aplikasi	288 aplikasi
Kapasitas bulanan	220 aplikasi	8.640 aplikasi
Biaya per aplikasi	\$45	\$2,50
Biaya per sewa yang berhasil	\$2.200	\$125
Tingkat kesalahan	5%	2%
Biaya koreksi kesalahan	\$90 per kesalahan	\$45 per eskalasi

Analisis biaya berbasis volume

Tabel berikut menunjukkan analisis biaya berbasis volume untuk skenario A. Dalam contoh ini, biaya sistem AI agen mencakup biaya tetap dan biaya pengaturan diamortisasi sebesar \$1.917 per bulan selama 12 bulan.

Volume bulanan	Biaya manusia	Biaya sistem AI agen	Tabungan bulanan
100 aplikasi	\$4.500	\$750	\$3.750
500 aplikasi	\$22.500	\$2.667	\$19.833
1.000 aplikasi	\$45.000	\$4.917	\$40.083

Analisis ROI

Tabel berikut menunjukkan analisis ROI untuk skenario A yang didasarkan pada pemrosesan 500 aplikasi per bulan.

Metrik	Nilai
Biaya manusia bulanan	\$22.500
Biaya agen bulanan	\$2.667
Tabungan bulanan	\$19.833
Tabungan tahunan	\$237.996
Periode impas	1,16 bulan

Perbandingan biaya kumulatif

Tabel berikut menunjukkan perbandingan biaya kumulatif untuk skenario A selama enam bulan pertama, dengan asumsi 500 aplikasi per bulan.

Bulan	Biaya manusia	Biaya sistem AI agen	Tabungan kumulatif
-------	---------------	----------------------	--------------------

1	\$22.500	\$25.667	-\$3,167
2	\$45.000	\$28.334	\$16.666
3	\$67.500	\$31,001	\$36.499
4	\$90.000	\$33.668	\$56.332
5	\$112.500	\$36.335	\$76.165
6	\$135.000	\$39.002	\$95.998

Manfaat tambahan dari sistem AI agen

Berikut ini adalah manfaat tambahan yang diberikan oleh sistem AI agen di Skenario A:

- Skalabilitas - Dapat menangani lonjakan volume tanpa biaya tambahan
- Ketersediaan - Operasi 24/7 dengan respons langsung
- Konsistensi — Aplikasi kriteria penyaringan seragam
- Efisiensi waktu - Berkurang secara signifikan time-to-hire
- Pengalaman pengguna - Umpan balik instan untuk kandidat

Skenario B: Waktu penyaringan 15 menit

Model Skenario B mengoptimalkan operasi rekrutmen di mana perekrut manusia telah merampingkan proses penyaringan mereka menjadi 15 menit per aplikasi. Ini mewakili peningkatan efisiensi 66% dibandingkan Skenario A. Skenario ini mempertahankan biaya tahunan yang sama penuh sebesar \$112.250 untuk perekrut tingkat menengah. Namun, ini menunjukkan peningkatan produktivitas manusia secara signifikan, dengan kapasitas harian meningkat menjadi 32 aplikasi selama shift 8 jam dan throughput bulanan mencapai 660 aplikasi. Peningkatan efisiensi manusia mengurangi biaya per aplikasi dari \$45 menjadi \$15, mempersempit kesenjangan ekonomi dengan sistem AI agen. Namun, agen mempertahankan keunggulan strukturalnya: waktu pemrosesan 5 menit, ketersediaan 24/7 memungkinkan 288 aplikasi harian, tingkat kesalahan 2% yang lebih rendah dibandingkan dengan manusia 5%, dan kapasitas bulanan melebihi 8.600 aplikasi. Sementara peningkatan efisiensi ini memperpanjang periode impas dari 1,16 bulan menjadi 4,76 bulan dan mengurangi penghematan bulanan dari \$19.833 menjadi \$4.833, analisis mengungkapkan bahwa sistem agen tetap layak

secara ekonomi bahkan ketika bersaing dengan operasi manusia yang sangat dioptimalkan— wawasan kritis untuk organisasi yang mengevaluasi apakah tingkat efisiensi proses mereka saat ini membenarkan investasi AI agen.

Struktur biaya dasar

Tabel berikut menunjukkan biaya tetap tahunan untuk skenario B.

Komponen	Operasi manusia	Sistem AI agen
Gaji pokok	\$65.000	N/A
Manfaat (30%)	\$19.500	N/A
Ruang Kerja & Peralatan	\$12.000	N/A
Pengawasan Manajemen (15%)	\$9.750	N/A
Pelatihan & Pengembangan	\$6.000	N/A
Total Biaya Tetap Tahunan	\$112.250	N/A

Tabel berikut menunjukkan biaya implementasi untuk skenario B.

Komponen	Operasi manusia	Sistem AI agen
Pengaturan awal	N/A	\$23.000
Biaya tetap bulanan	\$9.354	\$500

Metrik operasional

Tabel berikut menunjukkan metrik operasional untuk skenario B.

Metrik	Operasi manusia	Sistem AI agen
--------	-----------------	----------------

Waktu pemrosesan per aplikasi	15 menit	5 menit
Kapasitas per jam	4 aplikasi	12 aplikasi
Kapasitas harian (shift 8 jam)	32 aplikasi	288 aplikasi
Kapasitas bulanan	660 aplikasi	8.640 aplikasi
Biaya per aplikasi	\$15	\$2,50
Biaya per sewa yang berhasil	\$2.200	\$125
Tingkat kesalahan	5%	2%
Biaya koreksi kesalahan	\$30 per kesalahan	\$45 per eskalasi

Analisis biaya berbasis volume

Tabel berikut menunjukkan analisis biaya berbasis volume untuk skenario B. Dalam contoh ini, biaya sistem AI agen mencakup biaya tetap dan biaya pengaturan diamortisasi sebesar \$1.917 per bulan selama 12 bulan.

Volume bulanan	Biaya manusia	Biaya sistem AI agen	Tabungan bulanan
100 aplikasi	\$1.500	\$750	\$750
500 aplikasi	\$7.500	\$2.667	\$4.833
1.000 aplikasi	\$15.000	\$4.917	\$10.083

Analisis ROI

Tabel berikut menunjukkan analisis ROI untuk skenario B yang didasarkan pada pemrosesan 500 aplikasi per bulan.

Metrik	Nilai
--------	-------

Biaya manusia bulanan	\$7.500
Biaya sistem AI agen bulanan	\$2.667
Tabungan bulanan	\$4.833
Tabungan tahunan	\$57.996
Periode impas	4.76 bulan

Perbandingan biaya kumulatif

Tabel berikut menunjukkan perbandingan biaya kumulatif untuk skenario B selama enam bulan pertama, dengan asumsi 500 aplikasi per bulan.

Bulan	Biaya manusia	Biaya sistem AI agen	Tabungan kumulatif
1	\$7.500	\$25.667	-\$18.167
2	\$15.000	\$28.334	-\$13.334
3	\$22.500	\$31,001	-\$8.501
4	\$30.000	\$33.668	-\$3,668
5	\$37.500	\$36.335	\$1.165
6	\$45.000	\$39.002	\$5.998

Membandingkan biaya dan manfaat untuk setiap skenario

Metrik	Skenario A	Skenario B	Dampak
Waktu penyaringan	45 menit	15 menit	Peningkatan 66%
Kapasitas harian	10—11 aplikasi	32 aplikasi	Kenaikan 200%
Biaya per aplikasi	\$45	\$15	Pengurangan 66%

Tabungan bulanan (500 aplikasi)	\$19.833	\$4.833	Penurunan 76%
Periode impas	1,16 bulan	4.76 bulan	310% lebih lama

Skenario B menunjukkan peningkatan efisiensi yang signifikan dalam operasi manusia, dengan peningkatan waktu pemrosesan yang meningkatkan kapasitas tanpa jumlah karyawan tambahan dan mengurangi biaya per aplikasi secara substansional. Namun, dampak keuangan mengungkapkan gambaran yang lebih bernuansa: sementara ROI tetap positif, organisasi menghadapi periode impas yang diperpanjang dan pengurangan tabungan bulanan dibandingkan dengan Skenario A. Hasil ini menyoroti faktor keputusan penting untuk implementasi—sistem agen tetap layak secara finansial bahkan terhadap operasi manusia yang dioptimalkan, tetapi organisasi harus mengadopsi perspektif investasi jangka panjang dan dengan hati-hati mempertimbangkan fluktuasi volume dan kebutuhan skalabilitas saat mengevaluasi jadwal penerapan dan pengembalian yang diharapkan.

Namun, sistem AI agen masih mempertahankan keunggulan operasional kritis yang melampaui penghematan biaya murni. Ini menyediakan ketersediaan 24/7 untuk keterlibatan kandidat langsung terlepas dari zona waktu atau jam kerja. Ini memberikan kualitas penyaringan yang konsisten dengan menerapkan kriteria seragam untuk setiap aplikasi, timbangan untuk menangani lonjakan volume tanpa menimbulkan biaya tambahan. Ini menawarkan respons kandidat langsung yang meningkatkan pengalaman merek dan kandidat pemberi kerja, dan beroperasi dengan faktor kelelahan nol yang memastikan kinerja berkualitas tinggi yang sama pada aplikasi pertama seperti yang seperseribu.

Kesalahan manusia biasanya dihasilkan dari kelelahan, gangguan, atau kesenjangan pengetahuan dan sering melibatkan miskomunikasi atau informasi yang salah. Kesalahan sistem AI agen biasanya berasal dari kasus tepi, input ambigu, atau keterbatasan data pelatihan. Kesalahan ini cenderung lebih konsisten.

Metrik kualitas dan pengalaman mengungkapkan pertukaran yang jelas antara kemampuan manusia dan agen:

- Kepuasan pelanggan — Manusia unggul dalam empati dan pemecahan masalah yang kompleks, dan agen memberikan informasi yang konsisten dan akurat untuk pertanyaan rutin.
- Waktu respons - Waktu respons menguntungkan agen dengan ketersediaan 24/7 langsung. Manusia memberikan dukungan jam kerja dengan potensi penundaan antrian.
- Konsistensi — Agen memberikan tanggapan yang identik untuk pertanyaan serupa. Manusia dapat bervariasi dalam pendekatan dan aplikasi pengetahuan.

- Penanganan eskalasi — Masalah kompleks yang membutuhkan penilaian, kreativitas, atau kecerdasan emosional tetap menjadi kekuatan manusia.

Kesimpulan dan sumber daya

Ekonomi sistem manusia dibandingkan dengan sistem AI agen mewakili lebih dari sekedar keputusan teknologi. Ini mencerminkan transformasi mendasar dalam bagaimana organisasi menciptakan nilai, mengelola risiko, dan mencapai keunggulan kompetitif. Keberhasilan membutuhkan evaluasi sistematis karakteristik pekerjaan, pengukuran hasil yang komprehensif (termasuk faktor risiko), dan penskalaan strategis berdasarkan hasil yang terbukti.

[State of Enterprise AI Adoption](#) (laporan ISG 2025) mengungkapkan bahwa sebagian besar implementasi AI gagal karena celah belajar—sistem yang tidak dapat beradaptasi, mengingat konteks, atau meningkat seiring waktu. Organizations yang mencapai kesuksesan fokus pada sistem kemampuan belajar yang terintegrasi secara mendalam ke dalam alur kerja dan menunjukkan peningkatan berkelanjutan melalui umpan balik manusia dan pengalaman operasional.

Organisasi-organisasi yang memahami prinsip-prinsip ini—dimulai dengan pekerjaan yang tepat, menguraikan pekerjaan menjadi tugas, mengukur segala sesuatu termasuk dampak risiko, dan penskalaan apa yang berhasil—akan mencapai keunggulan kompetitif berkelanjutan melalui pemanfaatan sumber daya yang optimal dan otomatisasi yang berfokus pada hasil yang tumbuh seiring dengan kesuksesan bisnis.

Masa depan adalah milik organisasi yang secara cerdas dapat menggabungkan keahlian manusia dengan kemampuan AI agen. Ini menciptakan model hibrida yang memberikan hasil unggul sambil mempertahankan fleksibilitas, kapasitas belajar, dan manfaat kolaboratif yang diperlukan untuk kondisi pasar yang dinamis.

Sumber daya

Sumber daya berikut dapat membantu Anda merencanakan, merancang, dan mengimplementasikan sistem AI agen pada AWS:

- [Membangun arsitektur tanpa server untuk AI agen pada AWS\(Panduan Preskriptif\)](#)AWS
- [Operasionalisasi AI agen pada AWS\(Panduan Preskriptif\)](#)AWS
- [Pola dan alur kerja AI agen pada AWS](#) (Panduan AWS Preskriptif)
- [Agentic AI \(Panduan](#)AWS Preskriptif)
- [Hub Optimisasi Biaya AWS](#) (Layanan AWS)
- [Dokumentasi Amazon Bedrock](#) ()Layanan AWS

- [Pilar optimasi biaya](#) (AWS Well-Architected Framework)
- [Agen dan solusi AI](#) (AWS Marketplace)

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	—	Januari 28, 2026

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada 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

ABAC

Lihat [kontrol akses berbasis atribut](#).

layanan abstrak

Lihat [layanan terkelola](#).

ASAM

Lihat [atomisitas, konsistensi, isolasi, daya tahan](#).

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat [kecerdasan buatan](#).

AIOps

Lihat [operasi kecerdasan buatan](#).

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan pemrosesan atau modifikasi 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.

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

penyebaran kenari

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

CCoE

Lihat [Cloud Center of Excellence](#).

CDC

Lihat [mengubah pengambilan data](#).

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat [integrasi berkelanjutan dan pengiriman berkelanjutan](#).

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat [database manajemen konfigurasi](#).

repositori kode

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

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

cache dingin

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

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat penyimpanan atau kelas 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 Wilayah, 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 kekritisannya dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat [Klasifikasi data](#).

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat [bahasa definisi database](#).

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat [lingkungan](#).

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat [bahasa manipulasi basis data](#).

desain berbasis domain

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

DR

Lihat [pemulihan bencana](#).

deteksi drift

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

DVSM

Lihat [pemetaan aliran nilai pengembangan](#).

E

EDA

Lihat [analisis data eksplorasi](#).

EDI

Lihat [pertukaran data elektronik](#).

komputasi tepi

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

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat [Apa itu Pertukaran Data Elektronik](#).

enkripsi

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

kunci enkripsi

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

endianness

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

titik akhir

Lihat [titik akhir layanan](#).

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

- **Development Environment** — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- **lingkungan yang lebih rendah** — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- **lingkungan produksi** — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam 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. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga [zero-shot](#) prompting.

FGAC

Lihat kontrol [akses berbutir halus](#).

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui [pengambilan data perubahan](#) untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat [model pondasi](#).

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat [pembatasan geografis](#).

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur, gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

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

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (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 .

H

HA

Lihat [ketersediaan tinggi](#).

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan

adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

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

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

|

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.

IloT

Lihat [Internet of Things industri](#).

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAC)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretasi

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

IoT

Lihat [Internet of Things](#).

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat [perpustakaan informasi TI](#).

ITSM

Lihat [manajemen layanan TI](#).

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat [kontrol akses berbasis label](#).

hak istimewa paling sedikit

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

angkat dan geser

Lihat [7 Rs](#).

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga [endianness](#).

LLM

Lihat [model bahasa besar](#).

lingkungan yang lebih rendah

Lihat [lingkungan](#).

M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat [Program Percepatan Migrasi](#).

mekanisme

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

akun anggota

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

MES

Lihat [sistem eksekusi manufaktur](#).

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

informasi selengkapnya, lihat [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server](#).

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 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 file. AWS Cloud Untuk informasi lebih lanjut, lihat entri [7 Rs](#) di glosarium ini dan lihat [Memobilisasi organisasi Anda untuk mempercepat](#) migrasi skala besar.

ML

Lihat [pembelajaran mesin](#).

modernisasi

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

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta

jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat [Mengevaluasi kesiapan modernisasi untuk](#) aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat [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 machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat [panduan integrasi operasi](#).

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat [tinjauan kesiapan operasional](#).

OT

Lihat [teknologi operasional](#).

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

Lihat informasi yang [dapat diidentifikasi secara pribadi](#).

buku pedoman

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

PLC

Lihat [pengontrol logika yang dapat diprogram](#).

PLM

Lihat [manajemen siklus hidup produk](#).

kebijakan

[Objek yang dapat menentukan izin \(lihat kebijakan berbasis identitas\), menentukan kondisi akses \(lihat kebijakan berbasis sumber daya\), atau menentukan izin maksimum untuk semua akun 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 optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona host pribadi

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

kontrol proaktif

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

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat [lingkungan](#).

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

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

pseudonimisasi

Proses penggantian pengidentifikasi 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 kredensial.

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.

PENIPUAN

Lihat [kontrol pengawasan dan akuisisi data](#).

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadatanya. 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 kredensial.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator [tingkat layanan](#).

model tanggung jawab bersama

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

SIEM

Lihat [informasi keamanan dan sistem manajemen acara](#).

titik kegagalan tunggal (SPOF)

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

SLA

Lihat [perjanjian tingkat layanan](#).

SLI

Lihat [indikator tingkat layanan](#).

SLO

Lihat [tujuan tingkat layanan](#).

split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat [Pendekatan bertahap untuk memodernisasi aplikasi](#) di AWS Cloud

SPOF

Lihat [satu titik kegagalan](#).

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda 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.

gerbang transit

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian:

ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat [lingkungan](#).

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

Cache buffer yang berisi data 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.