Mempercepat siklus hidup pengembangan perangkat lunak dengan AI generatif AWS

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Mempercepat siklus hidup pengembangan perangkat lunak dengan AI generatif AWS

Table of Contents

Mempercepat siklus hidup pengembangan perangkat lunak dengan AI generatif AWS

Chetan Makvana, Amazon Web Services

April 2025 (riwayat dokumen)

Meningkatnya permintaan akan perangkat lunak berkualitas tinggi mendorong organisasi untuk terus mencari cara untuk mempercepat siklus hidup pengembangan perangkat lunak (SDLC) mereka. Karena organisasi berusaha untuk tetap kompetitif, sangat penting untuk mengurangi waktu ke pasar sambil mempertahankan atau meningkatkan kualitas produk. Untuk memenuhi tantangan ini, pengalaman pengembangan perangkat lunak harus berkembang dan menggunakan teknologi, metodologi, dan praktik mutakhir yang merampingkan proses dan memberdayakan tim pengembangan perangkat lunak agar lebih produktif dan kreatif. Munculnya pengalaman pengembangan generasi berikutnya menandai perubahan signifikan dalam bagaimana perangkat lunak disusun, dibangun, diuji, dan digunakan. Ini mengintegrasikan berbagai kemampuan — termasuk pengembangan cloud-native, otomatisasi berbasis AI, manajemen proyek lanjutan, alat kolaboratif, dan DevSecOps —yang secara kolektif meningkatkan efisiensi dan efektivitas SDLC.

Di garis depan transformasi ini adalah munculnya AI generatif dalam rekayasa perangkat lunak. Menurut Gartner, 40% tim rekayasa platform akan menggunakan AI untuk menambah setiap fase SDLC pada tahun 2027, dibandingkan dengan hanya 5% pada tahun 2023. Laporan ini juga menyatakan bahwa para pemimpin rekayasa perangkat lunak sekarang harus bersiap untuk mengadopsi AI generatif di berbagai bidang yang lebih luas yang penting untuk proses pengembangan. Dalam laporan lain, McKinseypenelitian menunjukkan bahwa perusahaan dengan indeks kecepatan pengembang yang lebih tinggi menumbuhkan pendapatan 4-5 kali lebih cepat, memiliki pengembalian pemegang saham 60% lebih tinggi, dan 55% lebih inovatif. Dengan merangkul AI generatif di luar sekadar pembuatan kode, organisasi dapat membuka tingkat efisiensi, produktivitas, dan inovasi baru dalam alur kerja pengembangan perangkat lunak mereka. Hal ini dapat mengurangi upaya manual, wawasan permukaan, dan menambah keahlian manusia.

Tujuan

Dokumen strategi ini menguraikan kerangka kerja, kemampuan dasar, kasus penggunaan, praktik terbaik, dan metrik keberhasilan yang dapat membantu Anda mempercepat SDLC Anda dengan AI

generatif. Ini menjelaskan bagaimana mengintegrasikan AI generatif secara efektif di semua tahap pengembangan untuk meningkatkan kualitas dan efisiensi produk.

Dokumen strategi ini dapat membantu Anda dan organisasi Anda mencapai tujuan berikut:

- Terapkan kerangka kerja, kemampuan dasar, kasus penggunaan, praktik terbaik, dan metrik keberhasilan untuk mempercepat SDLC Anda dengan AI generatif.
- Mengintegrasikan AI generatif secara efektif di semua tahap pengembangan untuk meningkatkan kualitas produk, kecepatan rilis, dan efisiensi pengembangan.
- Beradaptasi dengan pengembangan perangkat lunak generasi berikutnya dengan menggabungkan teknologi, metodologi, dan praktik AI mutakhir yang merampingkan proses dan memberdayakan tim pengembangan.

Audiens yang dituju

Dokumen strategi ini ditujukan untuk para pemimpin TI, manajer teknik, kepala petugas teknologi, dan tim pengembangan perangkat lunak yang ingin mempercepat siklus hidup pengembangan perangkat lunak mereka dengan menerapkan AI generatif pada praktik pengembangan mereka.

Memahami pengalaman pengembangan perangkat lunak

Pengalaman pengembangan perangkat lunak mencakup lingkungan, alat, dan proses yang digunakan tim pengembangan Anda di seluruh siklus hidup pengembangan perangkat lunak (SDLC). Ini mencakup lingkungan pengembangan terintegrasi (IDE), platform kolaborasi, kerangka kerja pengujian, sistem manajemen pengetahuan, pipa penyebaran, dan banyak lagi.

Pengalaman pengembangan yang dirancang dengan baik menyederhanakan alur kerja, mengurangi upaya manual, dan memberdayakan tim Anda untuk fokus pada tugas bernilai tinggi, yang pada akhirnya mempercepat SDLC Anda. Misalnya, dengan mengintegrasikan IDE, sistem kontrol versi, dan alat penyebaran secara mulus, Anda memungkinkan pengembang untuk menulis, menguji, dan menyebarkan kode dengan kecepatan dan efisiensi yang lebih besar dibandingkan dengan rantai alat terfragmentasi yang memerlukan handoff manual dan peralihan konteks. Demikian pula, mengintegrasikan kerangka kerja manajemen pengetahuan yang kuat membantu tim dengan mudah mengakses dan berbagi pengetahuan kelembagaan, praktik terbaik, dan dokumentasi. Ini meningkatkan produktivitas dan kemampuan pemecahan masalah mereka secara keseluruhan.

Pengalaman pengembangan perangkat lunak memiliki dampak langsung pada kinerja keseluruhan dan keberhasilan tim pengembangan perangkat lunak. Pengalaman yang kurang optimal dapat mengarah pada hal-hal berikut:

- Mengurangi produktivitas — Alat yang tidak efisien, alur kerja yang kompleks, dan kurangnya otomatisasi menghambat produktivitas tim, yang memperlambat pengiriman fitur dan pembaruan.
- Peningkatan utang teknis — Alat dan proses ad-hoc yang terintegrasi dengan buruk dapat mengakibatkan utang teknis, yang membuatnya lebih menantang untuk memelihara dan meningkatkan skala sistem perangkat lunak Anda dari waktu ke waktu.
- Inovasi yang berkurang — Ketika terhambat oleh tugas manual dan berulang, kemampuan tim Anda untuk mengeksplorasi teknologi baru dan mendorong inovasi terkendala.
- Kualitas yang dikompromikan — Proses pengujian dan penyebaran yang terfragmentasi meningkatkan risiko cacat dan kerentanan perangkat lunak. Ini dapat berdampak negatif pada kualitas keseluruhan perangkat lunak yang dikirimkan.

Dengan berinvestasi dalam pengalaman pengembangan perangkat lunak yang dirancang dengan baik, Anda dapat membuka manfaat yang signifikan, seperti waktu yang lebih cepat ke pasar, peningkatan kualitas perangkat lunak, peningkatan kepuasan tim pengembangan perangkat lunak, dan kelincahan bisnis yang lebih besar.

Memberdayakan pengalaman pengembangan perangkat lunak dengan AI generatif

Integrasi AI generatif ke dalam siklus hidup pengembangan perangkat lunak (SDLC) mewakili perubahan paradigma dalam bagaimana seluruh tim pengembangan perangkat lunak memahami, merancang, mengimplementasikan, dan memelihara solusi perangkat lunak. Generative AI memiliki potensi untuk merevolusi setiap fase SDLC, termasuk manajemen proyek, pengumpulan persyaratan, desain, pengkodean, pengujian, penyebaran, dan pemeliharaan.

Pada intinya, pengalaman pengembangan bertenaga AI generatif bertindak sebagai kolaborator cerdas untuk seluruh tim pengembangan perangkat lunak Anda, termasuk manajer produk, desainer, arsitek solusi, pengembang, penguji, dan personel operasi. Ini memberikan bantuan sadar konteks, menghasilkan artefak (seperti cerita pengguna, mock-up desain, cuplikan kode, dan kasus uji), menawarkan saran yang hampir real-time, dan bahkan memprediksi potensi masalah sebelum muncul. Pendekatan AI-augmented ini secara signifikan mengurangi beban kognitif pada anggota tim. Hal ini memungkinkan mereka untuk fokus pada keputusan strategis tingkat tinggi dan pemecahan masalah yang kompleks sementara AI generatif menangani tugas-tugas yang lebih biasa dan berulang.

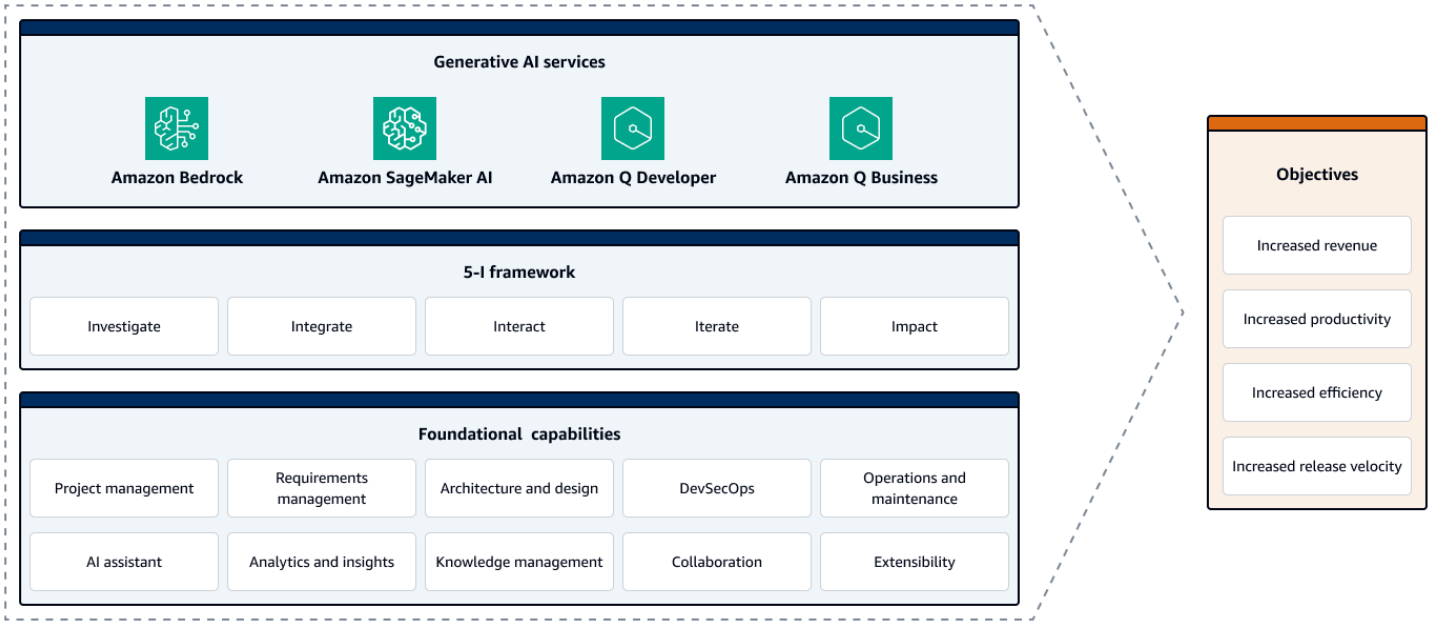
Generative AI juga berfungsi sebagai penguat pengetahuan. Ini membantu anggota tim dengan cepat mengakses informasi yang relevan, praktik terbaik, dan pola dari repositori data yang luas. Ini dapat secara efektif mendemokratisasikan keahlian di seluruh organisasi. Dengan mengintegrasikan kemampuan AI generatif secara mulus di seluruh rantai alat pengembangan, Anda dapat menciptakan lingkungan yang lebih intuitif, efisien, dan produktif untuk seluruh tim pengembangan perangkat lunak Anda. Pengalaman pengembangan yang ditingkatkan ini mempercepat SDLC dan meningkatkan kualitas secara keseluruhan. Ini juga mengurangi kesalahan dan mendorong inovasi karena anggota tim dapat mengeksplorasi ide dan pendekatan baru dengan lebih cepat.

Untuk mengadopsi pengalaman pengembangan yang didukung AI generatif di organisasi Anda, pertimbangkan elemen-elemen kunci berikut:

- Kerangka kerja 5-I— Terdiri dari lima dimensi, kerangka kerja 5-I memberikan pendekatan komprehensif untuk menavigasi proses pengembangan perangkat lunak modern. Ini menawarkan metodologi terstruktur yang membantu Anda menerapkan AI generatif secara sistematis di semua tahap SDLC.

- Kemampuan dasar— Untuk sepenuhnya menggunakan kekuatan AI generatif di seluruh dimensi pengembangan perangkat lunak modern, Anda perlu membangun seperangkat kemampuan dasar yang kuat. Kemampuan ini membentuk tulang punggung pengalaman pengembangan yang didukung AI. Kemampuan ini membantu Anda mengintegrasikan dan menggunakan AI generatif di seluruh SDLC.

Bersama-sama, kerangka kerja 5-I dan kemampuan dasar membentuk strategi untuk menata ulang pengalaman pengembangan perangkat lunak. Lima dimensi menyediakan kerangka kerja strategis untuk menerapkan AI generatif, dan kemampuan dasar mempersiapkan organisasi Anda untuk mendukung pendekatan berbasis AI ini. Layanan AWS, seperti Amazon Bedrock, Amazon SageMaker AI, Amazon Q Developer, dan Amazon Q Business, menyediakan kemampuan dan fitur AI generatif yang dapat Anda integrasikan ke dalam pengalaman pengembangan perangkat lunak Anda.



Kerangka kerja 5-I untuk pengalaman pengembangan perangkat lunak bertenaga AI

Kerangka kerja 5-I menyediakan pendekatan terstruktur bagi tim pengembangan perangkat lunak untuk secara efektif mengintegrasikan AI generatif ke dalam praktik pengembangan mereka. Ini membantu Anda membangun fondasi yang kuat untuk menggunakan AI generatif di seluruh SDLC.

Ini juga membantu Anda mengatur praktik pengembangan, alur kerja, dan pola pikir yang tepat untuk sepenuhnya memanfaatkan potensi AI generatif.

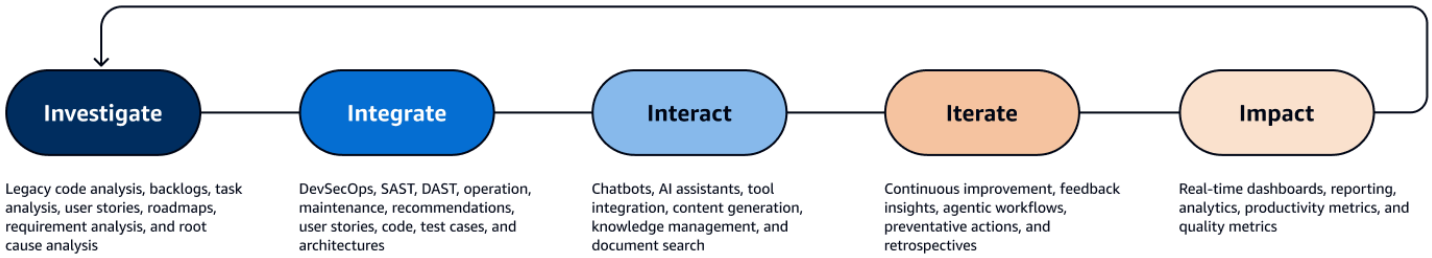
Bagian ini berisi topik berikut:

- Ikhtisar kerangka kerja
- Mengintegrasikan dengan siklus hidup pengembangan perangkat lunak

Ikhtisar kerangka kerja

Kerangka kerja 5-I dibangun di sekitar lima dimensi utama: Investigate, Integrate, Interact, Iterate, dan Impact. Setiap dimensi mewakili area kritis di mana AI generatif secara signifikan meningkatkan proses pengembangan perangkat lunak. Dengan mengintegrasikan AI generatif secara strategis di seluruh dimensi ini, kerangka kerja ini membahas kebutuhan pengembangan perangkat lunak modern yang terus berkembang. Ini dapat mengurangi beban kognitif dan memperkuat potensi kreatif. Ia mengakui bahwa pengalaman pengembangan yang ideal bukan hanya tentang alat — ini tentang menciptakan lingkungan di mana AI secara mulus meningkatkan kemampuan manusia di setiap tahap.

Diagram berikut menunjukkan lima dimensi pengembangan perangkat lunak bertenaga AI. Untuk setiap dimensi, ini menunjukkan di mana Anda dapat mengintegrasikan AI generatif untuk mendorong efisiensi dan inovasi.



Berikut ini adalah lima dimensi dalam kerangka kerja:

- Selidiki — Tingkatkan setiap tugas analitis dalam proses pengembangan perangkat lunak Anda dengan AI generatif. Gunakan AI generatif untuk memahami persyaratan, memproses data dalam jumlah besar, mengenali pola, dan menghasilkan wawasan yang mungkin berada di luar kapasitas manusia atau akan membutuhkan waktu lebih lama untuk diproduksi. Wawasan ini membantu Anda membuat keputusan yang lebih tepat, mengidentifikasi peluang peningkatan dengan cepat, dan menghadirkan perangkat lunak berkualitas tinggi dengan lebih efisien. Generative AI

dapat menjadi mitra cerdas untuk proses analitis di seluruh SDLC. Dengan memanfaatkan AI generatif, Anda menerapkan analisis mendalam ke area kritis, seperti pengumpulan persyaratan, pemeriksaan basis kode lama, dan pengoptimalan backlog produk. Misalnya, pemilik produk dapat menggunakan AI generatif untuk menganalisis perjalanan atau persyaratan pengguna sebelum membuat cerita pengguna. Tim pengembangan dapat mengungkap inefisiensi dan mengidentifikasi peluang pengoptimalan dalam basis kode yang ada. DevOps insinyur dapat menerapkan analisis akar penyebab untuk mendiagnosis masalah kinerja atau kerentanan keamanan dengan cepat, yang dapat meningkatkan keandalan.

- Integrasikan — Integrasikan AI generatif untuk mengotomatiskan berbagai tugas dan proses di seluruh SDLC. Ini termasuk secara otomatis membuat cuplikan kode, kasus uji, desain arsitektur, cerita pengguna, dan pipeline penerapan. Dengan mengotomatiskan tugas-tugas manual ini, tim dapat fokus pada pekerjaan yang lebih strategis dan inovatif, yang mendorong waktu yang lebih cepat untuk memasarkan dan aplikasi berkualitas tinggi. Dimensi Integrate mewakili perubahan paradigma dalam pengembangan perangkat lunak, di mana AI menjadi bagian integral dari proses pengembangan. Ia bekerja bersama tim pengembangan perangkat lunak Anda untuk meningkatkan produktivitas, meningkatkan kualitas, dan mendorong inovasi. Ini menghasilkan waktu yang lebih cepat untuk memasarkan. Ini menantang tim pengembangan perangkat lunak Anda untuk secara teratur menilai proses dan alur kerja mereka dengan menanyakan pada setiap langkah: “Bisakah ini otomatis?”
- Berinteraksi — Gunakan asisten generatif yang didukung AI untuk memberi tim Anda dukungan kontekstual instan di berbagai tugas dan pertanyaan. Asisten cerdas ini bertindak sebagai kolaborator berpengetahuan luas yang mengambil dari gudang informasi yang luas. Mereka dapat menjawab pertanyaan pengkodean, menawarkan saran desain, menjelaskan prosedur operasi standar, dan membantu memecahkan masalah yang kompleks. Mengintegrasikan asisten AI ini ke dalam alur kerja pengembangan meningkatkan produktivitas dan menumbuhkan lingkungan pemecahan masalah yang lebih kolaboratif.
- Iterasi — Gunakan AI generatif untuk mengaktifkan penyesuaian cepat dan berbasis data di seluruh SDLC. Anda dapat terus menganalisis data dari sumber seperti umpan balik pelanggan, pola penggunaan, tren pasar, dan metrik kinerja tim untuk membuat keputusan yang tepat dengan cepat. Kemampuan beradaptasi ini menyempurnakan pengembangan perangkat lunak Anda dari proses statis yang telah ditentukan sebelumnya menjadi pendekatan yang lancar dan responsif. Ini bermanifestasi dalam berbagai cara, termasuk prioritas dinamis backlog, alokasi sumber daya yang fleksibel, strategi pengujian adaptif, dokumentasi yang berkembang, dan proses penerapan responsif. Misalnya, manajer produk dapat menggunakan wawasan yang dihasilkan AI untuk menyusun ulang backlog mereka, mengintegrasikan persyaratan pelanggan baru dan

tren pasar dalam waktu dekat. DevOps Insinyur dapat mengadaptasi rencana penyebaran dan konfigurasi infrastruktur berdasarkan analisis kinerja, memastikan bahwa aplikasi tetap tangguh dan dioptimalkan. Tim pengembangan dapat menerjemahkan umpan balik dari retrospektif sprint ke dalam peningkatan yang dapat ditindaklanjuti untuk iterasi berikutnya, mendorong budaya peningkatan proses berkelanjutan.

- Dampak — Terapkan AI generatif untuk menilai efektivitas dan kinerja proses pengembangan perangkat lunak Anda. Dengan menggunakan analitik dan metrik yang didukung AI, Anda mendapatkan wawasan yang lebih dalam tentang efisiensi pengembangan, kualitas kode, keterlibatan pengguna, dan kinerja aplikasi secara keseluruhan. Pendekatan berbasis data ini membantu Anda membuat keputusan berdasarkan informasi, mengoptimalkan alur kerja pengembangan, dan terus meningkatkan kualitas dan pengalaman pengguna aplikasi Anda. Saat menilai produktivitas tim perangkat lunak, AI generatif menganalisis berbagai titik data, seperti frekuensi komit kode, waktu penyelesaian masalah, kecepatan rilis, tingkat pengiriman fitur, dan banyak lagi. Ini juga dapat mengevaluasi kualitas tinjauan kode, efektivitas alat kolaborasi, dan dampak praktik pengembangan yang berbeda pada output tim secara keseluruhan. Dengan menghubungkan metrik ini dengan hasil proyek, AI mengidentifikasi pola dan tren yang mungkin terlewatkan oleh analis manusia, dan mereka dapat memberikan wawasan yang dapat ditindaklanjuti yang meningkatkan produktivitas tim. Selain itu, AI generatif dapat membantu Anda membandingkan kinerja tim terhadap standar industri atau data historis, menawarkan rekomendasi yang dipersonalisasi untuk perbaikan. Ini juga dapat memprediksi potensi kemacetan atau risiko dalam proses pengembangan sehingga Anda dapat mengambil tindakan proaktif.

Mengintegrasikan dengan siklus hidup pengembangan perangkat lunak

SDLC terdiri dari beberapa fase, yang dapat berbeda dari organisasi ke organisasi. Umumnya, fase-fase ini meliputi: persyaratan dan perencanaan, desain dan arsitektur, implementasi, pengujian, penyebaran, dan operasi dan pemeliharaan.

Tabel berikut memetakan dimensi kerangka kerja 5-I ke fase SDLC dan memberikan tingkat integrasi untuk setiap dimensi.

Dimensi kerangka kerja	Persyaratan dan perencanaan	Desain dan arsitektur	Implementasi	Pengujian	Deployment	Operasi dan pemeliharaan
Selidiki	Tinggi	Rendah	Rendah	Rendah	Rendah	Sedang
Integrasikan	Sedang	Sedang	Tinggi	Sedang	Tinggi	Tinggi
Interaksi	Tinggi	Tinggi	Tinggi	Sedang	Sedang	Tinggi
Iterasi	Sedang	Rendah	Rendah	Rendah	Rendah	Sedang
Dampak	Tinggi	Sedang	Tinggi	Rendah	Tinggi	Tinggi

Tingkat integrasi bervariasi dari tinggi ke rendah. Pemetaan mengungkapkan area fokus utama untuk setiap dimensi. Misalnya, Investigasi menunjukkan intensitas tinggi dalam tahap persyaratan dan perencanaan. Integrasi menunjukkan intensitas tinggi dalam fase implementasi, penyebaran, dan operasi dan pemeliharaan.

Dengan menggunakan pemetaan ini, Anda dapat memprioritaskan upaya Anda secara efektif. Kami menyarankan Anda fokus pada tinggi, lalu sedang, dan kemudian rendah. Pastikan Anda mengadopsi pendekatan yang seimbang dan berdampak yang meningkatkan pengalaman pengembangan perangkat lunak dengan AI generatif.

Kemampuan dasar untuk pengalaman pengembangan perangkat lunak bertenaga AI

Agar berhasil menerapkan pengalaman pengembangan perangkat lunak bertenaga AI generatif, Anda perlu menetapkan serangkaian kemampuan dasar yang menjangkau banyak persona di organisasi Anda. Kemampuan ini mewakili kemampuan Anda untuk secara efektif menyebarkan sumber daya, mengimplementasikan proses, dan mencapai hasil yang diinginkan dalam konteks pengembangan perangkat lunak bertenaga AI. Dengan mengembangkan kemampuan ini, Anda menciptakan fondasi yang kuat yang membantu Anda mengintegrasikan AI generatif dengan mulus di semua tahap SDLC.

AWS menyediakan layanan utama untuk membantu Anda menerapkan kemampuan ini. Misalnya, Amazon Q Developer membantu mempercepat pengembangan perangkat lunak dengan bertindak sebagai asisten yang didukung AI. Amazon Q Business membantu Anda mendapatkan jawaban yang cepat dan relevan untuk pertanyaan mendesak, memecahkan masalah, dan menghasilkan konten. Itu juga dapat bertindak atas nama Anda dengan mengintegrasikan alat yang terkait dengan pengembangan perangkat lunak. Amazon Bedrock menyediakan akses ke model pondasi dan serangkaian kemampuan yang luas untuk menyesuaikan alur kerja dan persyaratan pengembangan tertentu.

Dengan mengembangkan kemampuan ini Layanan AWS, Anda menciptakan fondasi yang kuat yang membantu Anda mengintegrasikan AI generatif dengan mulus di semua tahap SDLC.

Berikut ini adalah kemampuan dasar yang harus Anda fokuskan:

- Manajemen proyek
- Manajemen persyaratan
- Arsitektur dan desain
- Kolaborasi
- DevSecOps
- Operasi dan pemeliharaan
- Asisten AI
- Analitik dan wawasan
- Manajemen pengetahuan
- Ekstensibilitas

Setiap kemampuan dasar terintegrasi dengan dimensi kerangka kerja dan tahapan SDLC yang berbeda. Integrasi ini membantu Anda menggunakan kemampuan AI secara efektif selama proses pengembangan perangkat lunak Anda. Ini meningkatkan efisiensi, kualitas, dan inovasi di setiap langkah. Sinergi antara kemampuan dasar ini, kerangka kerja, dan tahapan SDLC menciptakan ekosistem yang komprehensif untuk pengembangan perangkat lunak bertenaga AI. Ini membantu Anda memanfaatkan potensi penuh AI generatif, mendorong peningkatan berkelanjutan, mempercepat siklus pengembangan, dan menghadirkan produk perangkat lunak berkualitas.

Tabel berikut menunjukkan bagaimana kemampuan dasar dan subkapabilitas memetakan ke dimensi kerangka kerja dan fase SDLC.

Kemampuan : subkapabilitas	Selidiki	Integrasikan	Interaksi	Iterasi	Dampak
Manajemen proyek: Manajemen masalah	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Tidak ada	Tidak ada
Manajemen proyek: Sprint dan manajemen tugas	Persyaratan dan perencanaan	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Tidak ada
Manajemen proyek: Manajemen backlog produk	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Persyaratan dan perencanaan	Tidak ada
Manajemen proyek: Pemetaan cerita pengguna	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Tidak ada	Tidak ada
Manajemen proyek: Pelaporan dan analitik	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Tidak ada	Persyaratan dan perencanaan
Manajemen proyek: Manajemen peta jalan produk	Persyaratan dan perencanaan	Tidak ada	Persyaratan dan perencanaan	Tidak ada	Tidak ada

Kemampuan : subkapabilitas	Selidiki	Integrasikan	Interaksi	Iterasi	Dampak
Manajemen proyek: Loop umpan balik	Tidak ada	Tidak ada	Tidak ada	Persyaratan dan perencanaan	Tidak ada
Manajemen proyek: Retrospektif	Tidak ada	Tidak ada	Tidak ada	Persyaratan dan perencanaan	Tidak ada
Manajemen persyaratan	Persyaratan dan perencanaan	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Tidak ada
Arsitektur dan desain: Desain solusi	Desain dan arsitektur	Desain dan arsitektur	Tidak ada	Tidak ada	Tidak ada
Kolaborasi: Manajemen dokumentasi	Semua fase SDLC	Tidak ada	Semua fase SDLC	Tidak ada	Tidak ada
Kolaborasi: Berbagi pengetahuan	Semua fase SDLC	Tidak ada	Semua fase SDLC	Tidak ada	Tidak ada
Kolaborasi: Manajemen aset proyek	Tidak ada	Semua fase SDLC	Semua fase SDLC	Tidak ada	Tidak ada
DevSecOps: CI/CD	Pengujian, Penerapan	Implementasi, Pengujian, Penerapan	Deployment	Tidak ada	Tidak ada

Kemampuan : subkapabilitas	Selidiki	Integrasikan	Interaksi	Iterasi	Dampak
DevSecOps : DevOps keamanan	Implementasi	Implementasi, Pengujian, Operasi dan pemeliharaan	Tidak ada	Implementasi, Pengujian, Operasi dan pemeliharaan	Tidak ada
DevSecOps : Pemantauan kinerja aplikasi	Tidak ada	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada
DevSecOps: Agregasi log dan analitik	Operasi dan pemeliharaan	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada
DevSecOps: AIOps	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Operasi dan pemeliharaan	Tidak ada
DevSecOps : Perbaikan berkelanjutan	Tidak ada	Tidak ada	Tidak ada	Operasi dan pemeliharaan	Tidak ada
DevSecOps: Pemantauan dasbor	Tidak ada	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada
DevSecOps : Wawasan kinerja	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Operasi dan pemeliharaan	Tidak ada

Kemampuan : subkapabilitas	Selidiki	Integrasikan	Interaksi	Iterasi	Dampak
Operasi dan pemeliharaan: Manajemen insiden	Tidak ada	Tidak ada	Tidak ada	Operasi dan pemeliharaan	Tidak ada
Operasi dan pemeliharaan: Peningkatan kode	Tidak ada	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada
Operasi dan pemeliharaan: Optimalisasi kode	Operasi dan pemeliharaan	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada
Operasi dan pemeliharaan: Manajemen utang teknis	Tidak ada	Operasi dan pemeliharaan	Operasi dan pemeliharaan	Tidak ada	Tidak ada
Operasi dan pemeliharaan: Manajemen perubahan	Tidak ada	Implementasi, Penerapan	Tidak ada	Tidak ada	Tidak ada

Kemampuan : subkapabilitas	Selidiki	Integrasikan	Interaksi	Iterasi	Dampak
Operasi dan pemeliharaan: Rekayasa terbalik	Operasi dan pemeliharaan	Tidak ada	Tidak ada	Tidak ada	Tidak ada
Operasi dan pemeliharaan: Modernisasi kode	Tidak ada	Implementasi	Tidak ada	Tidak ada	Tidak ada
Operasi dan pemeliharaan: Optimalisasi kinerja	Tidak ada	Operasi dan pemeliharaan	Tidak ada	Operasi dan pemeliharaan	Tidak ada
Analitik dan wawasan	Tidak ada	Persyaratan dan perencanaan	Tidak ada	Tidak ada	Semua fase SDLC
Asisten AI	Tidak ada	Tidak ada	Semua fase SDLC	Tidak ada	Tidak ada
Manajemen pengetahuan	Tidak ada	Tidak ada	Semua fase SDLC	Tidak ada	Tidak ada
Ekstensibilitas	Tidak ada	Deployment	Tidak ada	Tidak ada	Tidak ada

Kasus penggunaan AI generatif untuk manajemen proyek

Manajemen proyek yang efektif adalah jantung dari pengembangan perangkat lunak yang sukses. Dalam konteks AI generatif, manajemen proyek mengambil dimensi baru. Ini bisa menjadi lebih prediktif, adaptif, dan berbasis data. Alat manajemen proyek bertenaga AI menganalisis data proyek historis untuk menghasilkan perkiraan waktu dan sumber daya yang lebih akurat. Mereka dapat secara otomatis memprioritaskan tugas berdasarkan tujuan bisnis dan kapasitas tim, dan mereka bahkan dapat memprediksi hambatan potensial sebelum terjadi. Misalnya, manajer proyek mungkin menggunakan AI generatif untuk membuat rencana proyek awal berdasarkan persyaratan proyek dan data historis dari proyek serupa. AI kemudian dapat menyarankan komposisi tim yang optimal yang memperhitungkan keterampilan, beban kerja, dan kebutuhan proyek. Sepanjang proyek, dasbor berbasis AI memberikan wawasan waktu nyata tentang status proyek dengan secara otomatis menghasilkan laporan dan menyoroti area yang memerlukan perhatian.

Pendekatan AI-augmented untuk manajemen proyek ini dapat meningkatkan efisiensi. Ini membantu manajer proyek fokus pada pengambilan keputusan strategis dan kepemimpinan tim, daripada terjebak dalam tugas-tugas administrasi rutin.

Tabel berikut menunjukkan kasus penggunaan manajemen proyek yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Subkapabilitas: Kasus penggunaan	Persona
Manajemen masalah: Membuat dan menetapkan masalah	Manajer proyek
Manajemen masalah: Mendeteksi masalah selama pengujian dan mencatatnya	Insinyur uji
Manajemen masalah: Prioritaskan masalah berdasarkan tingkat keparahan dan tetapkan ke pengembang	Manajer proyek
Manajemen masalah: Identifikasi dan gabungkan masalah duplikat	Manajer proyek

Subkapabilitas: Kasus penggunaan	Persona
Manajemen masalah: Lacak dan buat laporan tentang masalah utama, metrik, dan kesehatan proyek secara keseluruhan	Manajer proyek
Sprint dan manajemen tugas: Perkirakan upaya untuk tugas dan tetapkan poin cerita berdasarkan kapasitas tim	Master Scrum
Sprint dan manajemen tugas: Mendistribusikan tugas di antara anggota tim untuk beban kerja yang merata di seluruh sprint	Master Scrum
Sprint dan manajemen tugas: Memfasilitasi sesi perencanaan sprint yang menyelaraskan upaya tim dengan tujuan sprint	Master Scrum
Manajemen backlog produk: Menyusun ulang item backlog berdasarkan nilai bisnis, urgensi, dan umpan balik pengguna	Pemilik produk
Manajemen backlog produk: Integrasikan umpan balik pelanggan baru dan wawasan pasar ke dalam backlog produk untuk prioritas mendekati waktu nyata	Pemilik produk
Manajemen backlog produk: Identifikasi dan kelola dependensi antara item backlog untuk merampingkan pengembangan	Manajer produk
Pemetaan cerita pengguna: Buat peta perjalanan pengguna untuk mengidentifikasi semua fitur yang diperlukan dan cerita pengguna yang sesuai	Pemilik produk
Pemetaan cerita pengguna: Identifikasi celah atau langkah yang hilang dalam alur pengguna	Analis bisnis

Subkapabilitas: Kasus penggunaan	Persona
Pemetaan cerita pengguna: Prioritaskan cerita pengguna berdasarkan dampaknya terhadap nilai bisnis	Manajer produk
Pelaporan dan analitik: Buat dasbor mendekati waktu nyata yang memvisualisasikan metrik proyek utama, seperti kecepatan sprint dan tingkat resolusi masalah	Manajer proyek
Pelaporan dan analitik: Menganalisis data historis dan memprediksi hasil proyek masa depan, seperti potensi penundaan atau kemacetan	Manajer proyek
Pelaporan dan analitik: Membuat laporan khusus, seperti kinerja tim atau laporan status proyek, yang disesuaikan dengan pemangku kepentingan yang berbeda	Manajer proyek
Manajemen peta jalan produk: Membuat dan memelihara peta jalan produk yang menguraik an tonggak utama dan tanggal rilis	Manajer proyek
Manajemen peta jalan produk: Perbarui peta jalan berdasarkan perubahan prioritas atau jadwal proyek	Manajer produk
Manajemen peta jalan produk: Bagikan peta jalan dengan pemangku kepentingan untuk memberikan visibilitas ke arah produk	Manajer produk
Loop umpan balik: Kumpulkan umpan balik dari tim setelah setiap sprint dan identifikasi area untuk perbaikan	Master Scrum

Subkapabilitas: Kasus penggunaan	Persona
Retrospektif: Terjemahkan umpan balik ke item yang dapat ditindaklanjuti untuk sprint berikutnya, mendorong peningkatan berkelanjutan	Master Scrum
Retrospektif: Lacak dampak perubahan yang diterapkan dari retrospektif sebelumnya untuk mengukur efektivitasnya	Master Scrum

Kasus penggunaan AI generatif untuk manajemen persyaratan

Manajemen persyaratan adalah proses kritis yang terkait erat dengan manajemen proyek. Bayangkan seorang pemilik produk menggunakan alat AI untuk menganalisis umpan balik pelanggan, tren pasar, dan input pemangku kepentingan. Alat AI dapat menghasilkan serangkaian cerita dan persyaratan pengguna yang komprehensif, secara otomatis mengkategorikannya, mendeteksi potensi konflik atau kesenjangan, dan bahkan menyarankan prioritas berdasarkan nilai bisnis dan kompleksitas implementasi. Seiring kemajuan proyek dan persyaratan berkembang, AI dapat terus memperbarui dan menyempurnakan persyaratan untuk memastikan bahwa mereka tetap selaras dengan perubahan kebutuhan bisnis dan kendala teknis. Pendekatan dinamis berbasis AI terhadap manajemen persyaratan ini membantu memastikan bahwa upaya pengembangan tetap selaras dengan kebutuhan pengguna dan tujuan bisnis di seluruh siklus hidup proyek.

Tabel berikut menunjukkan kasus penggunaan manajemen persyaratan yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Kasus penggunaan	Persona
Buat persyaratan bisnis	Analis bisnis
Buat epos dari fitur	Pemilik produk
Lacak kemajuan epik dengan memantau penyelesaian cerita pengguna terkait	Manajer produk
Buat cerita pengguna	Pemilik produk

Kasus penggunaan	Persona
Perkirakan upaya yang diperlukan untuk setiap cerita penggunaan dan tetapkan poin cerita	Master Scrum
Tentukan kriteria penerimaan untuk setiap cerita pengguna	Pemilik produk

Kasus penggunaan AI generatif untuk arsitektur dan desain

Dengan dasar yang kuat dari manajemen proyek dan persyaratan yang terdefinisi dengan baik, kemampuan penting berikutnya adalah arsitektur dan desain. Di sini, AI generatif membuka kemungkinan baru untuk menciptakan arsitektur perangkat lunak yang kuat, terukur, dan efisien. Alat desain bertenaga AI dapat menganalisis persyaratan dan kendala untuk menyarankan pola arsitektur dan pendekatan desain yang optimal. Mereka menghasilkan beberapa alternatif desain, dan masing-masing dioptimalkan untuk prioritas yang berbeda, seperti kinerja, skalabilitas, atau pemeliharaan. Misalnya, arsitek solusi mungkin menggunakan asisten AI untuk dengan cepat menghasilkan beberapa desain arsitektur tingkat tinggi berdasarkan persyaratan proyek. Pendekatan AI-augmented ini mempercepat proses desain dan membantu arsitek membuat keputusan yang lebih tepat. Ini mengarah pada desain perangkat lunak yang lebih kuat dan tahan masa depan.

Tabel berikut menunjukkan kasus penggunaan arsitektur dan desain yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Kasus penggunaan	Persona
Buat dokumen arsitektur	Arsitek solusi
Buat dokumen desain terperinci	Pimpinan teknis
Memahami standar arsitektur dan desain yang ada	Arsitek solusi
Kembangkan mock-up dan prototipe rinci dari antarmuka pengguna	Desainer UX/UI

Kasus penggunaan AI generatif untuk kolaborasi

Pengembangan perangkat lunak secara inheren merupakan upaya kolaboratif. Anda dapat menggunakan AI generatif untuk meningkatkan kolaborasi pada tim pengembangan perangkat lunak Anda. Alat kolaborasi yang didukung AI melampaui pesan sederhana dan berbagi file. Mereka memfasilitasi komunikasi yang lebih efektif dengan meringkas utas diskusi panjang, menyoroti keputusan kunci, dan bahkan menyarankan waktu optimal untuk pertemuan berdasarkan jadwal anggota tim dan pola produktivitas. AI dapat membantu dalam tinjauan kode dengan secara otomatis mengidentifikasi potensi masalah, menyarankan perbaikan, dan bahkan menjelaskan perubahan kompleks kepada pengulas. Selama sesi brainstorming, AI dapat bertindak sebagai fasilitator, menghasilkan ide, membantu mengatur pemikiran, dan bahkan menengahi diskusi untuk memastikan bahwa semua suara didengar. Untuk tim terdistribusi, AI dapat membantu menjembatani hambatan budaya dan bahasa. Ini dapat memberikan terjemahan bahasa hampir real-time dalam obrolan dan panggilan video dan menawarkan konteks budaya untuk membantu mencegah kesalahpahaman. Dengan menambah kolaborasi manusia dengan AI, kemampuan ini membantu tim bekerja lebih efisien dan efektif, yang mendorong inovasi dan meningkatkan hasil proyek secara keseluruhan.

Tabel berikut menunjukkan bagaimana Anda dapat menggunakan AI generatif untuk meningkatkan kasus penggunaan kolaborasi.

Subkapabilitas: Kasus penggunaan	Persona
Manajemen dokumen: Membuat dan memelihara repositori dokumentasi terpusat	Penulis teknis
Manajemen dokumen: Izinkan beberapa anggota tim untuk berkolaborasi dalam dokumentasi secara real time	Tim pengembangan
Berbagi pengetahuan: Gunakan forum diskusi sebagai platform bagi pengembang untuk mengajukan pertanyaan, berbagi pengetahuan, dan memecahkan masalah secara kolaboratif	Tim pengembangan
Berbagi pengetahuan: Gunakan forum diskusi untuk mendokumentasikan dan melacak keputusan yang dibuat selama diskusi proyek, memastikan bahwa alasan di balik keputusan	Manajer produk

Subkapabilitas: Kasus penggunaan	Persona
utama ditangkap dan dapat diakses untuk referensi masa depan	
Manajemen aset proyek: Memfasilitasi pembagian sumber daya terkait proyek dengan mudah	Tim pengembangan
Manajemen aset proyek: Menerapkan kontrol versi untuk konten bersama sehingga anggota tim dapat melacak perubahan, kembali ke versi sebelumnya, dan berkolaborasi pada pembaruan konten	Tim pengembangan

Kasus penggunaan AI generatif untuk DevSecOps

DevSecOps Alat bertenaga AI mengotomatiskan banyak aspek dari jalur pengiriman perangkat lunak. Misalnya, mereka dapat melakukan tinjauan kode cerdas, mendeteksi potensi bug, mendeteksi kerentanan keamanan, dan mengidentifikasi masalah kinerja dalam waktu dekat saat pengembang menulis kode. AI menghasilkan dan menjalankan rangkaian pengujian yang komprehensif, dan secara otomatis memperbaruinya saat basis kode berkembang. Pendekatan AI-augmented ini untuk DevSecOps mempercepat jalur pengiriman dan secara signifikan meningkatkan keamanan dan keandalan perangkat lunak yang dikirimkan.

Tabel berikut menunjukkan kasus DevSecOps penggunaan yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan pengiriman berkelanjutan: Seluruh jaringan pipa penyebaran otomatis	DevOps insinyur
DevOps dan pengiriman berkelanjutan: Menerima umpan balik hampir real-time tentang kualitas kode dan potensi masalah	Pengembang perangkat lunak

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan pengiriman berkelanjutan: Menerima masalah keamanan dan rekomendasi remediasi yang hampir real-time	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Terima kode mendekati waktu nyata dan saran praktik terbaik	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Otomatiskan tugas berulang dan integrasikan perintah ke dalam skrip	DevOps insinyur
DevOps dan pengiriman berkelanjutan: Buat kode dan hasilkan artefak secara otomatis setelah setiap kode komit	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Bangun kode sesuai dengan standar dan kerangka kerja organisasi	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Secara otomatis menjalankan pengujian unit pada setiap komit untuk menangkap kesalahan di awal proses pengembangan	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Analisis cakupan pengujian unit untuk memastikan bahwa semua jalur kode kritis diuji	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Kelola cabang dan gabungkan perubahan	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Kelola versi kode dan artefak	Pengembang perangkat lunak

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan pengiriman berkelanjutan: Simpan dan kelola artefak dan dependensi build	DevOps insinyur
DevOps dan pengiriman berkelanjutan: Selesaikan dan ambil dependensi selama proses pembuatan	Pengembang perangkat lunak
DevOps dan pengiriman berkelanjutan: Hasilkan dan jalankan tes integrasi untuk memastikan bahwa komponen bekerja sama seperti yang diharapkan	Insinyur uji
DevOps dan pengiriman berkelanjutan: Gunakan layanan tiruan selama tes integrasi untuk mensimulasikan interaksi dengan sistem eksternal	Insinyur uji
DevOps dan pengiriman berkelanjutan: Benchmark kinerja aplikasi di bawah beban yang berbeda	Insinyur kinerja
DevOps dan pengiriman berkelanjutan: Simulasikan skenario lalu lintas tinggi untuk menguji skalabilitas dan waktu respons aplikasi	Insinyur kinerja
DevOps dan pengiriman berkelanjutan: Uji kemampuan sistem untuk pulih dari kegagalan , seperti kerusakan server atau pemadaman jaringan	Insinyur keandalan situs
DevOps dan pengiriman berkelanjutan: Lakukan rekayasa kekacauan	Insinyur keandalan situs
DevOps dan pengiriman berkelanjutan: Jalankan tes untuk memverifikasi bahwa aplikasi memenuhi persyaratan bisnis	Insinyur QA

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan pengiriman berkelanjutan: Lakukan pengujian penerimaan pengguna	Pemilik produk
DevOps dan pengiriman berkelanjutan: Pindai dependensi untuk kerentanan dan masalah kepatuhan lisensi	Insinyur keamanan
DevOps dan pengiriman berkelanjutan: Pantau dan kelola dependensi open source untuk memastikan bahwa dependensi tersebut mutakhir dan aman	Insinyur keamanan
DevOps dan pengiriman berkelanjutan: Menghasilkan dan memelihara tagihan bahan perangkat lunak (SBOM) untuk melacak semua komponen dan dependensi	Insinyur keamanan
DevOps dan pengiriman berkelanjutan: Gunakan SBOM untuk melakukan audit untuk kepatuhan terhadap peraturan	Petugas kepatuhan
DevOps dan pengiriman berkelanjutan: Buat catatan rilis	Manajer rilis
DevOps dan pengiriman berkelanjutan: Merencanakan dan mengoordinasikan rilis	Manajer rilis
DevOps dan pengiriman berkelanjutan: Menerapkan prosedur operasi standar untuk manajemen rollback dan rilis	Manajer rilis
DevOps dan pengiriman berkelanjutan: Gunakan bendera fitur untuk mengaktifkan atau menonaktifkan fitur dalam produksi tanpa menerapkan kode baru	Manajer produk

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan pengiriman berkelanjutan: Jalankan A/B pengujian menggunakan flag fitur untuk mengukur dampak berbagai fitur pada perilaku pengguna	Manajer produk
DevOps dan pengiriman berkelanjutan: Menganalisis dan memantau kegagalan pipa	DevOps insinyur
DevOps dan pengiriman berkelanjutan: Membuat dan mengelola sumber daya infrastru ktur	DevOps insinyur
DevOps dan keamanan: Pindai repositori kode untuk rahasia hardcode	DevOps insinyur
DevOps dan keamanan: Terapkan deteksi mendekati waktu nyata untuk segera mengingatkan pengembang jika rahasia dikomit ke repositori	DevOps insinyur
DevOps dan keamanan: Menegakkan pemantauan kualitas kode berkelanjutan	Pengembang perangkat lunak
DevOps dan keamanan: Mendeteksi dan menandai indikator potensi kerentanan keamanan dalam kode	Pengembang perangkat lunak
DevOps dan keamanan: Menerapkan pengujian otomatis untuk Open Worldwide Application Security Project (OWASP) 10 risiko keamanan teratas untuk memastikan bahwa aplikasi mematuhi praktik keamanan standar industri	Insinyur keamanan

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan keamanan: Secara teratur memperbarui dan mendidik pengembang tentang risiko OWASP dengan menginteg rasikan cek ke dalam proses pengembangan	Insinyur keamanan
DevOps dan keamanan: Pindai pustaka dan dependensi pihak ketiga untuk kerentanan keamanan yang diketahui	DevOps insinyur
DevOps dan keamanan: Pindai kode aplikasi dan infrastruktur untuk mendeteksi kerentanan	DevOps insinyur
DevOps dan keamanan: Analisis kode untuk kerentanan sebelum penerapan	Insinyur keamanan
DevOps dan keamanan: Menegakkan kebijakan keamanan dengan mencegah kode dengan kerentanan kritis digabungkan	Insinyur keamanan
DevOps dan keamanan: Menerapkan kontrol akses berbasis peran (RBAC) untuk membatasi akses ke sistem dan data sensitif dan untuk memastikan bahwa hanya personel yang berwenang yang dapat mengakses sumber daya penting	Insinyur keamanan
DevOps dan keamanan: Sesuaikan kontrol akses berdasarkan peran dan tanggung jawab dengan beradaptasi dengan perubahan dalam struktur tim	DevOps insinyur
DevOps dan keamanan: Uji aplikasi yang sedang berjalan untuk kerentanan keamanan dalam waktu dekat dengan mensimulasikan serangan pada lingkungan produksi	Insinyur keamanan

Subkapabilitas: Kasus penggunaan	Persona
DevOps dan keamanan: Terus memantau aplikasi yang digunakan untuk kerentanan keamanan	DevOps insinyur
DevOps dan keamanan: Jadwalkan pemindaian kerentanan reguler di semua lingkungan untuk mengidentifikasi dan mengatasi kelemahan keamanan	Insinyur keamanan
DevOps dan keamanan: Terapkan tambalan dan pembaruan berdasarkan hasil pemindaian kerentanan untuk membantu menjaga sistem yang aman	DevOps insinyur
Pemantauan kinerja aplikasi: Terus memantau kinerja aplikasi dalam waktu dekat untuk mendeteksi dan mendiagnosis masalah kinerja sebelum memengaruhi pengguna	Insinyur keandalan situs
Pemantauan kinerja aplikasi: Mendeteksi anomali kinerja, seperti lonjakan tiba-tiba dalam waktu respons atau peningkatan tingkat kesalahan, dan memulai peringatan	DevOps insinyur
Pemantauan kinerja aplikasi: Melacak permintaan saat mereka menyebar melalui sistem terdistribusi untuk mengidentifikasi kemacetan kinerja dan masalah latensi	DevOps insinyur
Pemantauan kinerja aplikasi: Gunakan penelusuran terdistribusi untuk menentukan layanan atau komponen yang tepat yang bertanggung jawab atas kegagalan atau penurunan kinerja	DevOps insinyur

Subkapabilitas: Kasus penggunaan	Persona
Agregasi dan analitik log: Agregat log dari berbagai sumber ke dalam sistem terpusat untuk memudahkan pencarian dan analisis untuk mengidentifikasi tren dan masalah	Insinyur keandalan situs
Agregasi dan analitik log: Menerapkan penguraian log otomatis untuk mengekstrak informasi yang relevan dan mendeteksi pola atau anomali yang mungkin mengindikasikan masalah	DevOps insinyur
Agregasi log dan analitik: Kumpulkan dan visualisasikan metrik kinerja utama	Insinyur keandalan situs
Agregasi dan analitik log: Pantau metrik terhadap perjanjian tingkat layanan yang telah ditentukan sebelumnya () SLAs	Manajer produk
Operasi AI: Mendeteksi insiden, menganalisis akar penyebab, dan memulai tindakan korektif tanpa campur tangan manusia	DevOps insinyur
Operasi AI: Memprediksi permintaan sumber daya masa depan dan mengoptimalkan perencanaan kapasitas untuk menghindari pemadaman	Insinyur keandalan situs
Peningkatan berkelanjutan: Memantau interaksi pengguna nyata dengan aplikasi untuk mengumpulkan wawasan tentang kinerja dan mengidentifikasi area untuk perbaikan	Desainer UX
Peningkatan berkelanjutan: Lacak kinerja aplikasi di berbagai wilayah geografis untuk memastikan pengalaman pengguna yang konsisten secara global	Manajer produk

Subkapabilitas: Kasus penggunaan	Persona
Pemantauan dasbor: Buat dasbor yang dapat disesuaikan untuk memvisualisasikan metrik, log, dan jejak penting dalam waktu dekat untuk memberikan pandangan komprehensif tentang kesehatan sistem	Insinyur keandalan situs
Pemantauan dasbor: Buat dasbor untuk tim yang berbeda (seperti tim pengembangan, operasi, dan produk) untuk memberikan wawasan yang relevan berdasarkan area fokus mereka	DevOps insinyur
Wawasan kinerja: Melakukan analisis rinci kinerja aplikasi untuk mengidentifikasi inefisien si dan mengoptimalkan kode atau infrastruktur	Pengembang perangkat lunak
Wawasan kinerja: Gunakan wawasan kinerja untuk meningkatkan kinerja aplikasi secara berulang dan mengoptimalkan pengalaman pengguna dari waktu ke waktu	Manajer produk

Kasus penggunaan AI generatif untuk operasi dan pemeliharaan

Setelah perangkat lunak digunakan, fokus beralih ke operasi dan pemeliharaan. AI generatif dapat meningkatkan pendekatan tradisional dengan menyediakan manajemen sistem yang lebih proaktif dan efisien. Alat operasi bertenaga AI terus memantau kinerja sistem dan memprediksi potensi masalah sebelum memengaruhi pengguna. Mereka melakukan analisis akar penyebab otomatis ketika masalah terjadi, yang secara signifikan mengurangi waktu rata-rata untuk penyelesaian. AI juga mengoptimalkan kinerja sistem dalam waktu dekat. Secara otomatis menyesuaikan konfigurasi berdasarkan perubahan pola beban dan perilaku pengguna. Misalnya, tim operasi mungkin menggunakan asisten AI untuk menghasilkan jadwal pemeliharaan prediktif, secara otomatis mengidentifikasi komponen yang kemungkinan gagal, dan menyarankan tindakan pencegahan. AI juga dapat membantu perencanaan kapasitas dengan menganalisis tren penggunaan dan memprediksi kebutuhan sumber daya masa depan dengan akurasi tinggi.

Tabel berikut menunjukkan kasus penggunaan operasi dan pemeliharaan yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Subkapabilitas: Kasus penggunaan	Persona
Manajemen insiden: Kelola insiden dalam waktu dekat dengan mengintegrasikan alat pemantauan dengan platform obrolan sehingga tim dapat mendeteksi, mendiskusikan, dan menyelesaikan masalah secara langsung dalam lingkungan obrolan	Insinyur keandalan situs
Manajemen insiden: Izinkan tim untuk memulai penerapan, menjalankan skrip, dan menjalank an perintah langsung dari antarmuka obrolan, yang merampingkan operasi	DevOps insinyur
Pemutakhiran kode: Tingkatkan dependens i dan pustaka kode untuk mengurangi upaya manual dan memastikan bahwa basis kode tetap up to date dengan versi terbaru	Pengembang perangkat lunak
Pengoptimalan kode: Tinjau kode untuk peluang pengoptimalan	Pengembang perangkat lunak
Optimalisasi kode: Identifikasi kemacetan dalam kode dan refactor atau optimalkan kode untuk meningkatkan kinerja	Pengembang perangkat lunak
Manajemen utang teknis: Log utang teknis sebagai bagian dari proses pengembangan	Manajer produk
Manajemen utang teknis: Memprioritaskan dan menangani utang teknis berdasarkan dampak, risiko, dan biaya, dan mengintegrasikannya ke dalam proses perencanaan sprint reguler	Pengembang perangkat lunak

Subkapabilitas: Kasus penggunaan	Persona
Manajemen utang teknis: Mengurangi utang teknis dalam kode aplikasi yang ada	Pengembang perangkat lunak
Manajemen perubahan: Menerapkan proses persetujuan perubahan yang memastikan bahwa semua perubahan kode ditinjau, diuji, dan disetujui oleh pemangku kepentingan yang diperlukan sebelum penerapan	Ubah manajer
Manajemen perubahan: Lakukan analisis dampak dari perubahan yang diusulkan	DevOps insinyur
Rekayasa balik: Menganalisis dan memahami struktur dan perilaku kode warisan	Arsitek solusi
Rekayasa balik: Jelaskan kode yang ada dan hasilkan dokumentasi	Pengembang perangkat lunak
Modernisasi kode: Terjemahkan kode dari satu bahasa pemrograman ke bahasa pemrograman lainnya	Pengembang perangkat lunak
Modernisasi kode: Modernisasi kode lama ke dalam bahasa pemrograman terbaru	Pengembang perangkat lunak
Optimalisasi kinerja: Terus memantau dan menyetel kinerja sistem dengan mengoptim alkan alokasi sumber daya, penyeimbangan beban, dan konfigurasi ulang aplikasi	Insinyur keandalan situs
Optimalisasi kinerja: Mengidentifikasi dan memfaktorkan ulang kode yang menyebabk an penurunan kinerja untuk meningkatkan kecepatan dan daya tanggap sistem	Pengembang perangkat lunak

Kasus penggunaan untuk asisten AI generatif dalam pengembangan perangkat lunak

Kemampuan asisten AI adalah jantung dari pengalaman pengembangan bertenaga AI generatif. Sistem cerdas dan sadar konteks ini berfungsi sebagai kolaborator virtual untuk semua anggota tim di seluruh SDLC. Bayangkan seorang pengembang mengerjakan sepotong kode yang kompleks. Mereka hanya dapat meminta bantuan asisten AI, dan dapat memberikan cuplikan kode yang relevan, menjelaskan algoritme yang rumit, atau bahkan menyarankan pengoptimalan berdasarkan konteks saat ini dan praktik terbaik. Asisten AI dapat membantu ITOps manajer memahami prosedur operasi standar berdasarkan dokumen internal. Dengan memberikan dukungan kontekstual instan, asisten AI secara signifikan mengurangi beban kognitif pada anggota tim. Ini membantu mereka fokus pada pemecahan masalah tingkat tinggi dan tugas-tugas kreatif. Kemampuan ini bertindak sebagai pengganda kekuatan yang meningkatkan produktivitas dan kualitas di semua tahap pengembangan perangkat lunak.

Tabel berikut menunjukkan kasus penggunaan yang dapat Anda tingkatkan dengan asisten AI dan persona yang diuntungkan.

Kasus penggunaan	Persona
Memberikan bantuan instan kepada tim pengembangan dengan menjawab pertanyaan, seperti tentang persyaratan, arsitektur, dan prosedur operasi standar	Tim pengembangan perangkat lunak
Cari atau ambil kutipan dari dokumentasi ekstensif atau buat ringkasan dengan menggunakan kueri bahasa alami	Tim pengembangan perangkat lunak
Meringkas dokumen teknis yang panjang, seperti dokumen persyaratan, dokumentasi desain arsitektur, dan proses internal	Tim pengembangan perangkat lunak
Pertahankan pustaka petunjuk yang dapat digunakan tim untuk tugas-tugas umum	Tim pengembangan perangkat lunak

Kasus penggunaan	Persona
Integrasikan AI generatif dengan mulus ke dalam alat dan sistem yang ada	Tim pengembangan perangkat lunak
Mengotomatiskan tugas di berbagai platform, alat, dan sistem internal	Tim pengembangan perangkat lunak
Buat gudang pengetahuan terpusat, termasuk praktik terbaik, informasi spesifik proyek, dan pengetahuan tim, yang dapat diakses oleh semua anggota tim	Tim pengembangan perangkat lunak
Mengambil pengetahuan yang relevan dari repositori berdasarkan konteks tugas	Tim pengembangan perangkat lunak
Lakukan tinjauan kode otomatis, analisis akar penyebab, sarankan perbaikan, deteksi potensi bug, dan lakukan pemecahan masalah	Pengembang perangkat lunak, DevOps insinyur, dan insinyur keandalan situs
Menganalisis data kinerja untuk mengident ifikasi tren dan pola yang dapat menginfor masikan keputusan tentang optimasi kinerja	Insinyur keandalan situs
Memberikan rekomendasi untuk meningkat kan efisiensi, mengurangi kompleksitas, dan meningkatkan keamanan	Pengembang perangkat lunak
Sarankan pengoptimalan untuk penggunaa n sumber daya cloud, seperti rekomendasi penskalaan atau strategi penghematan biaya	Pengembang perangkat lunak, DevOps insinyur, insinyur keandalan situs, dan arsitek solusi
Menghasilkan konten baru, seperti dokumenta si berdasarkan kode, panduan pengguna, atau rilis fitur produk	Tim pengembangan perangkat lunak

Kasus penggunaan AI generatif untuk analitik dan wawasan

Kemampuan analitik dan wawasan membantu mengubah sejumlah besar data menjadi wawasan yang dapat ditindaklanjuti yang mendorong pengambilan keputusan dan peningkatan berkelanjutan. Dengan menggunakan AI generatif, kemampuan ini memproses data dari berbagai sumber, termasuk repositori kode, alat manajemen proyek, dan platform kolaborasi tim, untuk memberikan pandangan holistik tentang proses pengembangan dan produktivitas tim. AI generatif melampaui metrik tradisional untuk menawarkan analisis prediktif dan preskriptif. Ini dapat memperkirakan masalah potensial dan menyarankan perbaikan yang ditargetkan. Misalnya, dapat menganalisis pola dalam komit kode, tingkat resolusi bug, dan kecepatan pengiriman fitur untuk mengidentifikasi tim berkinerja tinggi, menentukan kemacetan, dan menyarankan pengoptimalan proses. Selain itu, dapat memberikan wawasan tentang dinamika tim dan kinerja individu. Wawasan ini membantu para pemimpin membuat keputusan berdasarkan data tentang distribusi beban kerja, kebutuhan pelatihan, dan komposisi tim. Dengan menyajikan wawasan ini melalui dasbor interaktif, kemampuan ini memberdayakan pemangku kepentingan di semua tingkatan untuk membuat keputusan yang tepat, mengoptimalkan proses, dan terus meningkatkan produktivitas tim, yang mengarah pada pengiriman perangkat lunak berkualitas tinggi yang lebih cepat.

Tabel berikut menunjukkan kasus penggunaan analitik yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Kasus penggunaan	Persona
Memantau produktivitas individu dan tim	Manajer pengembangan
Analisis tren produktivitas untuk mendeteks i potensi burnout sehingga Anda dapat mengambil tindakan proaktif untuk menjaga kesejahteraan dan produktivitas tim	Manajer pengembangan
Lacak seberapa sering perubahan kode diterapkan ke produksi untuk mengukur kecepatan dan kelincahan proses pengemban gan	Manajer produk
Menganalisis data frekuensi penyebaran untuk mengidentifikasi periode aktivitas penyebaran	Manajer produk

Kasus penggunaan	Persona
rendah yang mungkin menunjukkan inefisiensi proses atau kendala sumber daya	
Ukur waktu antara komit kode untuk penerapan untuk mengidentifikasi peluang untuk merampingkan proses pengembangan dan penerapan	Manajer pengembangan
Lacak persentase penerapan yang mengakibatkan kegagalan yang memerlukan perbaikan segera untuk menilai keandalan proses rilis	Insinyur keandalan situs
Gunakan metrik tingkat kegagalan perubahan untuk mengidentifikasi area kode yang sering menyebabkan masalah untuk memandu upaya refactoring dan pengujian yang ditargetkan	Pengembang perangkat lunak
Pantau berapa lama waktu yang dibutuhkan untuk memulihkan layanan setelah pemadaman atau insiden sehingga Anda dapat mengurangi waktu henti dan meningkatkan ketahanan sistem secara keseluruhan	Insinyur keandalan situs
Menganalisis tren waktu pemulihan untuk meningkatkan proses respons insiden dan mendorong pemulihan lebih cepat dari kegagalan sistem	DevOps insinyur
Buat dasbor khusus yang menggabungkan metrik utama, seperti frekuensi penerapan, waktu tunggu, dan tingkat kegagalan perubahan, untuk memberikan pandangan komprehensif tentang pengembangan dan kesehatan operasional	Manajer produk

Kasus penggunaan	Persona
Buat dasbor yang disesuaikan dengan kebutuhan tim yang berbeda untuk memberikan wawasan terfokus ke bidang tanggung jawab spesifik mereka, seperti pengembangan, operasi, atau bisnis	Manajer produk
Melacak indikator kinerja kunci bisnis (KPIs), seperti dampak pendapatan, kepuasan pelanggan, dan pangsa pasar, untuk menyelaraskan upaya pengembangan dengan tujuan bisnis yang lebih luas	Manajer produk
Menganalisis dampak fitur baru pada bisnis KPIs untuk menilai keberhasilan mereka dan memandu pengembangan produk masa depan	Analis bisnis
Pantau metrik kualitas kode, seperti kompleksitas kode, cakupan pengujian, dan kepadatan bug, untuk memastikan bahwa basis kode tetap dapat dipertahankan dan aman	Pengembang perangkat lunak
Identifikasi area basis kode yang memerlukan refactoring untuk mendorong keberlanjutan jangka panjang dan mengurangi utang teknis	Arsitek solusi

Kasus penggunaan AI generatif untuk manajemen pengetahuan

Dalam setiap organisasi pengembangan perangkat lunak, pengetahuan adalah aset penting. Kemampuan manajemen pengetahuan, didukung oleh AI generatif, meningkatkan cara aset ini ditangkap, diatur, dan digunakan. Sistem manajemen pengetahuan tradisional sering mengandung terlalu banyak informasi, mengandung konten yang sudah ketinggalan zaman, atau sulit dicari untuk menemukan informasi yang relevan dengan cepat.

AI generatif mengatasi tantangan ini secara langsung. Secara otomatis menghasilkan, dan memperbarui dokumentasi berdasarkan perubahan kode, percakapan, dan artefak proyek. Ini

memastikan bahwa basis pengetahuan tetap terkini tanpa memerlukan upaya manual dari anggota tim. Lebih penting lagi, AI membuat pengetahuan ini dapat diakses dengan cara yang intuitif. Anggota tim dapat mengajukan pertanyaan dalam bahasa alami, dan AI dapat memberikan jawaban yang relevan. AI dapat menarik dari berbagai sumber, seperti dokumentasi resmi, komentar kode, utas diskusi, dan bahkan sumber daya eksternal. Misalnya, anggota tim baru yang mencoba memahami komponen tertentu dapat bertanya kepada AI, “Bagaimana cara kerja modul otentikasi?” AI kemudian akan memberikan penjelasan singkat dan tautan ke bagian kode yang relevan, diagram arsitektur, dan perubahan terbaru. Bahkan bisa menyesuaikan informasi ini berdasarkan peran anggota tim dan tingkat keahlian.

Kemampuan ini mempercepat orientasi, mengurangi pertanyaan berulang, dan mempromosikan berbagi pengetahuan di seluruh organisasi. Ini membantu melestarikan pengetahuan kelembagaan, membuatnya lebih mudah bagi tim untuk mempertahankan dan mengembangkan sistem yang kompleks dari waktu ke waktu.

Tabel berikut menunjukkan kasus penggunaan manajemen pengetahuan yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Kasus penggunaan	Persona
Buat platform terpadu yang memudahkan untuk mengakses semua pengetahuan terkait proyek	Tim pengembangan perangkat lunak
Menangkap pengetahuan dari berbagai kegiatan pembangunan	Tim pengembangan perangkat lunak
Menyediakan fungsionalitas pencarian lanjutan untuk dengan cepat menemukan pengetahuan yang relevan dalam repositori	Tim pengembangan perangkat lunak
Personalisasi modul dan jalur pembelajaran untuk tim	Tim pengembangan perangkat lunak

Kasus penggunaan AI generatif untuk ekstensibilitas

Ekstensibilitas memungkinkan integrasi tanpa batas dengan alat dan alur kerja yang ada sambil memungkinkan organisasi untuk menyesuaikan sistem AI dengan kebutuhan spesifik mereka.

Kemampuan ini menyediakan antarmuka yang kuat APIs SDKs, dan dapat disesuaikan yang memfasilitasi integrasi fungsionalitas AI ke dalam alat pengembangan dan manajemen proyek yang populer. Misalnya, organisasi dapat meningkatkan Jira dengan fitur yang didukung AI untuk prioritas tiket otomatis, estimasi upaya, dan perencanaan sprint. Anda dapat menambah jaringan pipa Jenkins dengan AI untuk pengoptimalan build cerdas dan pemilihan tes prediktif.

Selain itu, ekstensibilitas memungkinkan integrasi mendalam dengan lingkungan pengembangan terintegrasi (IDEs), sistem kontrol versi, dan platform peninjauan kode. AI dapat membantu membuat kode, mengotomatiskan ulasan kode, dan menghasilkan dokumentasi kontekstual.

Kemampuan ini juga mendukung pelatihan dan penyempurnaan model AI pada data spesifik organisasi. Ini membantu AI memahami pola pengkodean khusus perusahaan, preferensi arsitektur, dan pengetahuan domain. Hasilnya adalah bantuan yang lebih relevan dan sadar konteks di semua alat terintegrasi. Dengan memberikan tingkat fleksibilitas dan integrasi ini, ekstensibilitas memastikan bahwa pengalaman pengembangan yang didukung AI berkembang bersama organisasi. Ini dapat beradaptasi dengan perubahan teknologi dan kebutuhan bisnis sambil meningkatkan rantai alat dan alur kerja yang ada dengan mulus.

Tabel berikut menunjukkan kasus penggunaan ekstensibilitas yang dapat Anda tingkatkan dengan AI generatif dan persona yang bertanggung jawab atas kasus penggunaan tersebut.

Kasus penggunaan	Persona
Integrasikan alat pihak ketiga ke dalam lingkungan pengembangan	DevOps insinyur
Buat alur kerja otomatisasi khusus yang disesuaikan dengan proses pengembangan unik tim	DevOps insinyur
Connect ke berbagai APIs layanan	DevOps insinyur
Buat konektor untuk alat lintas platform	DevOps insinyur

Praktik terbaik untuk menggunakan AI generatif dalam pengembangan perangkat lunak

Bagian ini menjelaskan praktik terbaik untuk mengintegrasikan AI generatif ke dalam siklus hidup pengembangan perangkat lunak (SDLC). Dari menerapkan rantai alat dan DevSecOps jaringan pipa yang mulus hingga mendorong kolaborasi dan mengotomatiskan tugas berulang, panduan ini membantu Anda memanfaatkan kekuatan AI untuk meningkatkan proses dan pengalaman pengembangan Anda. Dengan mengikuti praktik terbaik ini, tim pengembangan perangkat lunak dapat membuka tingkat efisiensi, inovasi, dan kualitas baru dalam pekerjaan mereka.

Bagian ini membahas praktik terbaik berikut:

- Menerapkan rantai alat end-to-end terintegrasi yang mulus
- Menerapkan end-to-end CI/CD pipa untuk DevSecOps
- Mengadopsi alat dan praktik kolaboratif
- Mengotomatiskan tugas berulang
- Secara teratur meninjau dan mengulangi pengalaman pengembangan
- Mengadopsi praktik manajemen proyek yang efektif
- Menerapkan manajemen pengetahuan
- Menyediakan ekstensibilitas dan kustomisasi
- Mengoptimalkan operasi
- Menggunakan wawasan berbasis data
- Mengadopsi pendekatan berbasis platform

Menerapkan rantai alat end-to-end terintegrasi yang mulus

Menerapkan rantai alat end-to-end terintegrasi yang mulus adalah praktik terbaik dasar untuk menciptakan pengalaman pengembangan yang didukung AI generatif. Ide intinya adalah membangun ekosistem alat dan platform yang kohesif yang dapat digunakan tim perangkat lunak Anda di seluruh SDLC. Tim dapat menggunakan rantai alat untuk merencanakan, mengidealkan, membuat kode, membangun, menguji, menyebarkan, dan mengelola operasi yang sedang berlangsung. Dengan mengintegrasikan kemampuan AI generatif ke dalam rantai alat ini, Anda

memastikan bahwa bantuan AI tersedia di setiap tahap. Integrasi ini mengurangi atau menghilangkan handoff manual, mengurangi peralihan konteks, dan membantu aliran data dan artefak dengan lancar di antara fase pengembangan yang berbeda. Misalnya, cuplikan kode yang dihasilkan AI dari lingkungan pengembangan terintegrasi (IDE) Anda dapat mengalir dengan mulus ke sistem kontrol versi Anda, dan analitik bertenaga AI dari platform penerapan Anda dapat menginformasikan alat manajemen proyek Anda. Ini menciptakan loop umpan balik berkelanjutan yang meningkatkan proses pengembangan Anda.

Menerapkan end-to-end CI/CD pipa untuk DevSecOps

Untuk membangun rantai alat terintegrasi ini, menerapkan integrasi end-to-end berkelanjutan dan penerapan berkelanjutan (CI/CD) pipeline for DevSecOps. This AI-powered pipeline is a critical component that streamlines your software delivery processes. It helps you release new applications and updates more quickly and reliably. By embedding security practices throughout the entire SDLC, you can identify and address vulnerabilities much earlier, which reduces the overall cost and risk. The pipeline should incorporate AI at every stage, from continuous integration and testing to security checks and deployment. For instance, you can use AI to analyze code commits in near real time so that you can predict potential integration issues before they occur. In the CI/CDpipeline, Anda juga dapat menggunakan AI generatif untuk memperbarui kebijakan keamanan secara otomatis berdasarkan intelijen ancaman terbaru.

Mengadopsi alat dan praktik kolaboratif

Saat Anda meningkatkan infrastruktur pembangunan Anda, jangan lupakan elemen manusia. Pengembangan perangkat lunak secara inheren merupakan upaya kolaboratif. Ini melibatkan tim lintas fungsi yang terdiri dari pengembang, desainer, manajer produk, Scrum Masters, analis bisnis, dan pemangku kepentingan lainnya. Orang-orang ini bekerja secara kolektif untuk membawa ide-ide membuahkan hasil. Dengan menggunakan alat kolaboratif modern dan menumbuhkan budaya komunikasi terbuka dan berbagi pengetahuan, Anda dapat secara signifikan meningkatkan produktivitas dan efektivitas tim pengembangan perangkat lunak Anda. Dalam pengalaman pengembangan perangkat lunak bertenaga AI Anda, alat ini mengambil dimensi baru. Anda dapat mengintegrasikan AI ke dalam platform kolaborasi untuk memfasilitasi komunikasi dan berbagi pengetahuan yang lebih efektif di antara anggota tim. Asisten AI dapat menjawab pertanyaan umum, meringkas diskusi, atau bahkan menengahi konflik. AI generatif dapat meningkatkan proses peninjauan kode dengan secara otomatis menyarankan perbaikan atau mengidentifikasi potensi masalah. Selain itu, Anda dapat menggunakan AI untuk membuat dokumentasi dinamis dan sadar

konteks yang diperbarui dalam waktu dekat saat proyek berkembang sehingga semua anggota tim memiliki akses ke informasi terkini dan relevan.

Mengotomatiskan tugas berulang

Dengan menggunakan AI generatif untuk menangani aktivitas rutin dan memakan waktu, Anda membebaskan tim perangkat lunak Anda untuk fokus pada karya kreatif bernilai tinggi yang mendorong inovasi dan memberikan dampak bisnis. Contoh tugas berulang termasuk menghasilkan kode boilerplate, membuat data uji, menulis dokumentasi, atau bahkan menyusun rencana proyek awal. Dengan membongkar tugas-tugas ini ke AI, anggota tim dapat fokus pada pekerjaan yang lebih kreatif dan strategis. Misalnya, alat penyelesaian kode yang didukung AI dapat secara signifikan mempercepat proses pengkodean dengan menyarankan cuplikan kode yang relevan berdasarkan konteks dan pola pengkodean. Demikian pula, AI generatif dapat secara otomatis membuat dan memperbarui dokumentasi teknis saat kode berubah. Ini membuat dokumentasi tetap terkini dan mengurangi upaya manual yang biasanya diperlukan untuk tugas ini. Dalam pengujian, AI dapat menghasilkan kasus uji komprehensif berdasarkan persyaratan dan analisis kode, yang meningkatkan cakupan pengujian dan mengurangi kemungkinan kasus tepi yang diabaikan. Dengan mengotomatiskan tugas-tugas berulang ini secara cerdas, AI generatif mempercepat jadwal pengembangan, meningkatkan konsistensi, dan mengurangi kesalahan manusia. Hasilnya adalah output perangkat lunak berkualitas lebih tinggi.

Secara teratur meninjau dan mengulangi pengalaman pengembangan

Pengalaman pengembangan perangkat lunak Anda sendiri harus diperlakukan sebagai produk yang membutuhkan penyempurnaan berkelanjutan. Ini melibatkan pembentukan proses sistematis untuk meninjau dan mengulangi secara teratur pada semua aspek siklus hidup pengembangan, alat, dan praktik. Lakukan penilaian berkala dari seluruh rantai alat, alur kerja, dan proses. Kumpulkan umpan balik dari semua anggota tim di berbagai peran, termasuk manajer produk, desainer, arsitek, pengembang, penguji, dan personel operasi. Minta mereka untuk mengidentifikasi titik nyeri, kemacetan, dan peluang untuk peningkatan. Misalnya, tim dapat melakukan tinjauan triwulanan atas kinerja CI/CD pipeline mereka dan menganalisis metrik seperti waktu pembuatan, frekuensi penerapan, dan tingkat kesalahan untuk mengidentifikasi area untuk pengoptimalan. Karena kemampuan AI generatif terus berkembang pesat, sangat penting untuk secara konsisten mengevaluasi alat dan fitur baru yang didukung AI yang mungkin lebih merampingkan alur kerja atau menambah kemampuan di semua peran dalam SDLC.

Mengadopsi praktik manajemen proyek yang efektif

Untuk mengatur upaya pengembangan perangkat lunak kompleks Anda secara efektif, adopsi praktik manajemen proyek yang diperkuat AI. Dalam konteks ini, manajemen proyek yang efektif melampaui metodologi tradisional. Ini mencakup pendekatan AI-augmented yang meningkatkan perencanaan, pelaksanaan, dan pemantauan di seluruh SDLC. Kerangka kerja tangkas mempromosikan fleksibilitas, kolaborasi, dan iterasi cepat, dan Anda dapat menggunakan AI generatif untuk mengoptimalkan proses ini. Misalnya, AI generatif dapat menganalisis data proyek historis untuk perkiraan yang lebih akurat, secara otomatis menghasilkan dan memprioritaskan cerita pengguna berdasarkan tujuan bisnis dan umpan balik pelanggan, dan memberikan wawasan cerdas tentang kinerja tim. Alat manajemen proyek yang didukung AI dapat memprediksi hambatan potensial dan menyarankan penugasan tugas yang optimal berdasarkan keterampilan dan beban kerja anggota tim. Dengan mengintegrasikan kemampuan yang didukung AI ke dalam praktik manajemen proyek, Anda dapat mencapai visibilitas yang lebih besar, membuat keputusan berbasis data lebih cepat, dan memastikan bahwa anggota tim selaras dan bekerja secara efisien menuju tujuan bersama.

Menerapkan manajemen pengetahuan

Saat pengalaman pengembangan perangkat lunak bertenaga AI Anda matang, terapkan sistem manajemen pengetahuan yang kuat. Sistem manajemen pengetahuan yang kuat membantu Anda menangkap, mengatur, dan memberikan akses ke wawasan, praktik terbaik, dan solusi yang berharga. Semua anggota tim di seluruh SDLC harus memiliki akses mudah ke sistem. Gunakan AI generatif untuk menciptakan basis pengetahuan yang dinamis dan cerdas yang berkembang bersama organisasi Anda. Misalnya, AI dapat secara otomatis menghasilkan dan memperbarui dokumentasi berdasarkan perubahan kode, percakapan, dan artefak proyek sehingga informasi tetap terkini tanpa intervensi manual. AI generatif juga dapat memperkuat kemampuan pencarian cerdas dan membantu anggota tim dengan cepat menemukan informasi yang relevan dengan menggunakan kueri bahasa alami, bahkan jika mereka tidak tahu terminologi yang tepat. Selain itu, AI generatif dapat secara proaktif memunculkan informasi yang relevan kepada anggota tim berdasarkan tugas atau tantangan mereka saat ini. Ini bertindak sebagai mentor virtual yang meningkatkan pengambilan keputusan dan pemecahan masalah di semua peran. Dengan menerapkan sistem manajemen pengetahuan yang didukung AI, Anda dapat memecah silo, mempercepat orientasi, mengurangi pekerjaan yang berlebihan, dan menumbuhkan budaya pembelajaran dan inovasi berkelanjutan di seluruh tim pengembangan perangkat lunak Anda.

Menyediakan ekstensibilitas dan kustomisasi

Untuk memaksimalkan manfaat AI generatif dalam pengembangan perangkat lunak, pastikan alat dan platform bertenaga AI Anda dapat diperluas dan dapat disesuaikan. Ini membantu Anda menyesuaikan kemampuan AI dengan kebutuhan spesifik, alur kerja, dan tumpukan teknologi Anda. Misalnya, Anda dapat menyempurnakan model AI pada basis kode dan dokumentasi Anda sendiri, membuat alat bertenaga AI khusus untuk tugas tertentu, atau mengintegrasikan kemampuan AI ke dalam alat dan proses yang ada. Ekstensibilitas ini membantu Anda mengembangkan pengalaman pengembangan yang didukung AI untuk memenuhi kebutuhan organisasi yang berubah. Ini juga membantu Anda mengoptimalkan pengalaman untuk domain atau jenis proyek tertentu.

Mengoptimalkan operasi

AI generatif memainkan peran penting dalam mengoptimalkan operasi dan pemeliharaan perangkat lunak. Optimalkan operasi dengan mengintegrasikan kemampuan AI ke dalam alat dan proses operasional Anda. Misalnya, gunakan AI generatif untuk menganalisis data log dalam waktu dekat, memprediksi potensi kegagalan sistem, dan mengotomatiskan tugas pemeliharaan rutin. AI generatif juga dapat membantu analisis akar penyebab dengan menghubungkan peristiwa di seluruh sistem terdistribusi yang kompleks. Ini meningkatkan keandalan sistem, mengurangi waktu henti, dan membebaskan tim operasi Anda untuk fokus pada inisiatif yang lebih strategis.

Menggunakan wawasan berbasis data

Gunakan wawasan berbasis data di seluruh perjalanan pengembangan yang didukung AI Anda. Menerapkan sistem untuk mengumpulkan, menganalisis, dan menindaklanjuti data dari semua tahapan SDLC. Ini termasuk metrik kode, hasil pengujian, data penyebaran, umpan balik pengguna, dan kinerja operasional. Gunakan AI generatif untuk mengungkap pola dan wawasan yang mungkin tidak terlihat oleh pengamat manusia. Kemudian, masukkan kembali wawasan ini ke dalam proses pengembangan Anda untuk menginformasikan semuanya mulai dari keputusan arsitektur hingga prioritas fitur.

Mengadopsi pendekatan berbasis platform

Untuk sepenuhnya menyadari manfaat AI generatif dalam pengembangan perangkat lunak, adopsi pendekatan berbasis platform. Buat platform komprehensif dan terintegrasi yang menggabungkan kemampuan AI di semua aspek SDLC. Platform harus memberikan pengalaman pengguna yang

konsisten, manajemen dan data terpusat, dan integrasi yang mulus antara berbagai alat dan proses. Ini membuat manfaat AI tersedia secara seragam di seluruh organisasi Anda, mengurangi biaya pengelolaan beberapa alat AI yang berbeda, dan memberikan dasar untuk peningkatan dan perluasan kemampuan AI yang berkelanjutan.

Mengukur keberhasilan AI generatif dalam pengembangan perangkat lunak

Untuk secara efektif mengukur efek penerapan pengalaman pengembangan perangkat lunak bertenaga AI generatif, Anda perlu membuat serangkaian metrik komprehensif yang mencakup berbagai dimensi siklus hidup pengembangan perangkat lunak (SDLC) Anda. Metrik ini harus menangkap peningkatan segera dalam efisiensi dan produktivitas dan juga mencerminkan keuntungan jangka panjang dalam kualitas perangkat lunak, kepuasan tim, dan nilai bisnis.

Lakukan hal berikut untuk secara efektif menggunakan metrik yang direkomendasikan di bagian ini:

1. Tetapkan garis dasar — Sebelum Anda terjun ke dalam penerapan pengalaman pengembangan yang didukung AI, luangkan waktu untuk mengumpulkan data komprehensif tentang kinerja Anda saat ini di seluruh metrik ini. Ini memberikan titik awal yang jelas dan membantu Anda membuat perbandingan yang berarti nanti.
2. Tetapkan target realistis — Dengan baseline Anda di tangan, tetapkan target peningkatan yang dapat dicapai untuk setiap metrik. Jadilah ambisius tetapi realistis. Ingatlah bahwa kemajuan berkelanjutan seringkali bersifat inkremental.
3. Terapkan pemantauan berkelanjutan — Gunakan alat otomatis untuk terus mengumpulkan dan menganalisis data untuk metrik ini di lingkungan Anda. Pemantauan waktu dekat membantu Anda memantau kemajuan dan dengan cepat mengidentifikasi masalah atau peluang apa pun.
4. Lakukan tinjauan rutin — Jadwalkan sesi tinjauan triwulanan atau dua tahunan di mana Anda dan tim Anda menilai kemajuan Anda secara menyeluruh terhadap target. Gunakan sesi ini untuk mengidentifikasi area untuk perbaikan lebih lanjut dan merayakan kesuksesan Anda.
5. Iterasi dan sesuaikan — Berdasarkan wawasan yang Anda peroleh, terus perbaiki implementasi AI generatif Anda dan sesuaikan target seperlunya.

Bagian ini menjelaskan kategori metrik berikut:

- Kecepatan penyebaran
- Kualitas kode
- Efisiensi operasional
- Produktivitas dan kepuasan tim
- Dampak bisnis

Kecepatan penyebaran

Pertimbangkan untuk mengukur metrik kecepatan penerapan berikut.

Metrik	Deskripsi
Waktu ke pasar	Ukur pengurangan waktu dari konsepsi ide hingga penyebaran produksi
Kecepatan sprint	Lacak peningkatan poin cerita yang diselesai kan per sprint oleh tim Anda
Frekuensi komit kode	Pantau peningkatan komit kode, yang menunjukkan percepatan siklus pengemban gan
Tarik waktu resolusi permintaan	Menilai penurunan waktu yang dibutuhka n untuk meninjau dan menggabungkan perubahan kode di repositori Anda
Kecepatan rilis	Ukur peningkatan jumlah rilis per kuartal atau tahun

Kualitas kode

Pertimbangkan untuk mengukur metrik kualitas kode berikut.

Metrik	Deskripsi
Kepadatan cacat	Ukur pengurangan bug perangkat lunak
Cakupan kode	Lacak peningkatan persentase cakupan pengujian di seluruh basis kode Anda
Utang teknis	Pantau penurunan utang teknis yang teridenti fikasi dari waktu ke waktu

Metrik	Deskripsi
Skor analisis kode statis	Menilai peningkatan kualitas kode berdasarkan alat analisis otomatis Anda

Efisiensi operasional

Pertimbangkan untuk mengukur metrik efisiensi operasional berikut.

Metrik	Deskripsi
Frekuensi penyebaran	Ukur peningkatan jumlah penerapan yang berhasil
Waktu rata-rata untuk pemulihan (MTTR)	Lacak pengurangan jumlah waktu yang diperlukan untuk pulih dari kegagalan sistem
Ubah tingkat kegagalan	Pantau penurunan persentase perubahan yang mengakibatkan kegagalan dalam penerapan Anda

Produktivitas dan kepuasan tim

Pertimbangkan untuk mengukur metrik produktivitas dan kepuasan tim berikut.

Metrik	Deskripsi
Peningkatan produktivitas	Pantau peningkatan persentase produktivitas untuk setiap tugas
Skor kepuasan	Lakukan survei rutin untuk mengukur peningkat an moral dan kepuasan kerja tim Anda
Efisiensi berbagi pengetahuan	Ukur pengurangan waktu yang dihabiskan tim Anda untuk mencari informasi atau mengajukan pertanyaan berulang

Metrik	Deskripsi
Waktu on-boarding	Lacak penurunan waktu yang dibutuhkan anggota tim baru untuk menjadi produktif

Dampak bisnis

Pertimbangkan untuk mengukur metrik dampak bisnis berikut.

Metrik	Deskripsi
Tingkat adopsi fitur	Ukur peningkatan keterlibatan pengguna dengan fitur baru yang telah Anda rilis
Skor kepuasan pelanggan	Lacak peningkatan umpan balik dan peringkat pengguna Anda
Dampak pendapatan (langsung dan tidak langsung)	Menilai peningkatan pendapatan yang dikaitkan dengan peningkatan kecepatan rilis atau peningkatan produktivitas

Kesimpulan

Dokumen strategi ini memberikan gambaran umum tentang pengalaman pengembangan perangkat lunak bertenaga AI generatif. Ini mengeksplorasi lima dimensi dalam kerangka 5-I —Investigate, Integrate, Interact, Iterate, dan Impact. Dimensi ini menyediakan peta jalan strategis untuk mengintegrasikan AI generatif di seluruh siklus hidup pengembangan perangkat lunak (SDLC). Ini juga menjelaskan kemampuan dasar yang diperlukan untuk berhasil mengimplementasikan kerangka kerja ini. Kemampuan mencakup bidang-bidang seperti manajemen proyek DevSecOps, asisten AI, manajemen pengetahuan, dan banyak lagi. Ini memberikan praktik terbaik untuk dipertimbangkan saat mengintegrasikan AI generatif, dan ini membantu Anda menggunakan metrik untuk mengukur dampak AI generatif terhadap pengalaman pengembangan perangkat lunak Anda.

Integrasi AI generatif ke dalam proses pengembangan perangkat lunak merupakan perubahan paradigma yang berpotensi mempercepat inovasi, meningkatkan kualitas, dan meningkatkan produktivitas. Namun, penting untuk menyadari bahwa ini bukan implementasi satu kali. Ini adalah evolusi berkelanjutan yang membutuhkan upaya berkelanjutan dan penyempurnaan berkelanjutan.

Saat Anda memulai perjalanan ini, kami sarankan Anda memulai dengan penilaian menyeluruh tentang kemampuan dan kesiapan organisasi Anda saat ini. Alat AWS Penilaian adalah alat penilaian pengembangan perangkat lunak bertenaga AI yang dapat membantu Anda mengidentifikasi area prioritas dan membuat peta jalan implementasi yang disesuaikan.

Sumber daya

Setelah Anda mengidentifikasi area prioritas utama, sumber daya berikut dapat membantu Anda menerapkan peta jalan Anda:

AWS dokumentasi

- Mengotomatiskan operasi AWS infrastruktur dengan menggunakan Amazon Bedrock (Panduan AWS Preskriptif)
- Praktik terbaik dengan Amazon Q Developer untuk pembuatan kode in-line dan asisten (Panduan AWS Preskriptif)
- Kembangkan asisten berbasis obrolan otomatis sepenuhnya dengan menggunakan agen dan basis pengetahuan Amazon Bedrock (AWS Panduan Preskriptif)
- Mengubah model operasi pengembangan dan pemeliharaan aplikasi AWS dengan AI generatif (Panduan AWS Preskriptif)

- Gunakan Amazon Q Developer sebagai asisten pengkodean untuk meningkatkan produktivitas Anda (PanduanAWS Preskriptif)

AWS posting blog dan tutorial

- Postingan blog Amazon Q
- Mempercepat Siklus Hidup Pengembangan Perangkat Lunak Anda dengan Amazon Q (AWS posting blog)
- Membangun Agen AI Arsitek AWS Solusi: Memanfaatkan Amazon Bedrock untuk Arsitektur dan Penerapan Otomatis (video)AWS
- Operasi teknologi bertenaga AI generatif (AWS posting blog)
- Modernisasi aplikasi Java Anda dengan Amazon Q Developer (AWS posting blog)
- Gunakan Amazon Bedrock untuk menghasilkan, mengevaluasi, dan memahami kode dalam pipeline pengembangan perangkat lunak Anda (posting AWS blog)

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	Tidak berlaku	April 18, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada 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 kekritisan dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat Klasifikasi data.

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat bahasa definisi database.

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

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

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

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

pertukaran data elektronik (EDI)

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

enkripsi

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

kunci enkripsi

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

endianness

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

titik akhir

Lihat titik akhir layanan.

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam 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 ()OUs. Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub 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.

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

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

aplikasi idle

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

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

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

persistensi poliglot

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

penilaian portofolio

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

predikat

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

predikat pushdown

Teknik 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 kredensil.

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

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

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 metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat Apa yang ada di rahasia Secrets Manager? dalam dokumentasi Secrets Manager.

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

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

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations 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.