Kerangka kerja, platform, protokol, dan alat AI Agentic di AWS

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Kerangka kerja, platform, protokol, dan alat AI Agentic di AWS

Table of Contents

Kerangka kerja, platform, protokol, dan alat AI Agentic di AWS

Aaron Sempf, Ansley Verzosa, dan Joshua Samuel, Amazon Web Services ()AWS

Januari 2026 (sejarah dokumen)

Agentic AI adalah paradigma yang kuat di persimpangan AI, sistem terdistribusi, dan rekayasa perangkat lunak. Ini adalah kelas sistem cerdas yang terdiri dari agen perangkat lunak asinkron otonom yang menggunakan model AI dan terintegrasi dengan alat dan sumber daya. Agen pameran, dapat memahami konteks, alasan atas tujuan, membuat keputusan, dan mengambil tindakan yang disengaja atas nama pengguna atau sistem. Agen-agen ini beroperasi secara independen, seringkali secara kolaboratif, dalam lingkungan terdistribusi dan dirancang untuk mengejar tujuan yang didelegasikan dengan kecerdasan, memori, dan niat yang tertanam.

Pada AWS, organisasi dapat memanfaatkan AI agen untuk mengotomatiskan alur kerja yang kompleks, meningkatkan proses pengambilan keputusan, dan menciptakan sistem yang lebih responsif. Panduan ini memberikan informasi tentang komponen utama yang diperlukan untuk membangun solusi AI agen yang efektif:

- Kerangka kerja memprofilkan kerangka kerja AI agen saat ini, termasuk ulasan tentang manfaat dan kasus penggunaannya. Pelajari bagaimana kerangka kerja ini mengurangi pengangkatan berat yang tidak berdiferensiasi di seluruh pola, protokol, dan alat. Pahami kriteria pemilihan utama untuk memilih kerangka kerja yang tepat untuk kebutuhan Anda.
- Platform memberikan gambaran umum tentang platform AI agen (agen terkelola, orkestrasi sumber terbuka, dan hibrida) dan pertimbangan untuk pemilihan atau desain.
- Protokol mengeksplorasi protokol komunikasi standar penting untuk interaksi agen. Agent-to-agentprotokol yang muncul, seperti open source Model Context Protocol (MCP) dan Agent2Agent (A2A), bersama dengan implementasi proprietary lainnya. Temukan bagaimana protokol umum memungkinkan protokol yang berbeda untuk berinteraksi dengan mulus.
- Tools menyediakan informasi tentang alat berbasis protokol (seperti MCP), framework-native tools, dan meta-tools. Organizations dapat membangun toolkit yang terintegrasi dengan sistem kunci dalam alur kerja mereka, memungkinkan alur kerja agen pengguna akhir dan berbasis server.

Audiens yang dituju

Panduan ini ditujukan untuk arsitek, pengembang, dan pemimpin teknologi yang ingin memanfaatkan kekuatan agen perangkat lunak berbasis AI dalam aplikasi cloud-native modern.

Tujuan

Panduan ini membantu Anda melakukan hal berikut:

- Bandingkan kerangka kerja AI agen yang berbeda untuk memilih yang paling tepat untuk kasus penggunaan Anda.
- Pelajari tentang platform AI agen yang menyediakan kemampuan untuk mengubah agen individu menjadi sistem adaptif yang terkoordinasi.
- Memahami keuntungan dari protokol terbuka untuk membangun arsitektur AI agen yang berkelanjutan.
- Buat strategi integrasi alat yang tepat saat membangun sistem agen.

Tentang seri konten ini

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

Kerangka Kerja

Fondasi AI agen pada AWS memeriksa pola inti dan alur kerja yang memungkinkan perilaku otonom yang diarahkan pada tujuan. Inti penerapan pola-pola ini terletak pada pilihan kerangka kerja. Kerangka kerja adalah fondasi perangkat lunak dari kode yang telah ditulis sebelumnya yang menyediakan lingkungan terstruktur dan fungsionalitas umum untuk membangun dan mengelola, alat, dan kemampuan orkestrasi yang diperlukan untuk membangun agen AI otonom siap produksi.

Kerangka kerja AI agen yang efektif menyediakan beberapa kemampuan penting yang mengubah interaksi model bahasa besar mentah (LLM) menjadi sistem terkoordinasi dan cerdas yang mampu melakukan penalaran, kolaborasi, dan tindakan:

- Orkestrasi agen mengoordinasikan arus informasi dan pengambilan keputusan di seluruh agen tunggal atau ganda untuk mencapai tujuan yang kompleks tanpa campur tangan manusia.
- Integrasi alat memungkinkan agen untuk berinteraksi dengan sistem eksternal APIs, dan sumber data untuk memperluas kemampuan mereka di luar pemrosesan bahasa. Untuk informasi selengkapnya, lihat Ikhtisar Alat dalam Strands Agents dokumentasi.
- Manajemen memori menyediakan keadaan persisten atau berbasis sesi untuk mempertahankan konteks lintas interaksi, penting untuk tugas yang berjalan lama atau adaptif. Kerangka kerja yang lebih canggih menggabungkan memori jangka panjang untuk menyimpan ringkasan dan preferensi pengguna, memungkinkan pengalaman agen yang dipersonalisasi dan sadar kontekstual. Untuk informasi selengkapnya, lihat Cara berpikir tentang kerangka kerja agen di LangChain Blog.
- Definisi alur kerja mendukung pola terstruktur seperti rantai, perutean, paralelisasi, dan loop refleksi yang memungkinkan penalaran otonom yang canggih.
- Penyebaran dan pemantauan memfasilitasi transisi dari pengembangan ke produksi dengan observabilitas untuk sistem otonom. Untuk informasi selengkapnya, lihat pengumuman ketersediaan AgentCore umum Amazon Bedrock.

Kemampuan ini diimplementasikan dengan berbagai pendekatan dan penekanan di seluruh lanskap kerangka kerja, masing-masing menawarkan keuntungan berbeda untuk kasus penggunaan agen otonom dan konteks organisasi yang berbeda.

Bagian ini membuat profil dan membandingkan kerangka kerja terkemuka untuk membangun solusi AI agen, dengan fokus pada kekuatan, keterbatasan, dan kasus penggunaan ideal untuk operasi otonom:

- Agen Helai
- LangChain dan LangGraph
- CrewAI
- AutoGen
- ???
- Membandingkan kerangka kerja AI agen

> **Note**
>
> Bagian ini mencakup kerangka kerja yang secara khusus mendukung agensi AI dan tidak mencakup antarmuka frontend atau AI generatif tanpa agensi.

Strands Agents

Strands Agentsadalah SDK open-source yang awalnya dirilis oleh AWS, seperti yang dijelaskan dalam AWS Open Source Blog. Strands Agentsdirancang untuk membangun agen AI otonom dengan pendekatan model-first Ini menyediakan kerangka kerja yang fleksibel dan dapat diperluas yang dirancang untuk bekerja dengan mulus Layanan AWS sambil tetap terbuka untuk integrasi dengan komponen pihak ketiga. Strands Agents sangat ideal untuk membangun solusi yang sepenuhnya otonom.

Fitur utama Strands Agents

Strands Agentstermasuk fitur utama berikut:

- Desain model-first — Dibangun di sekitar konsep bahwa model fondasi adalah inti dari kecerdasan agen, memungkinkan penalaran otonom yang canggih. Untuk informasi selengkapnya, lihat Agent Loop dalam Strands Agents dokumentasi.
- Pola kolaborasi multi-agen — Model koordinasi bawaan seperti pola Swarm, Graph, dan Workflow yang memungkinkan kolaborasi dan tata kelola yang dapat diskalakan di seluruh jaringan agen terdistribusi. Untuk informasi selengkapnya, lihat Pola Multi-agen dalam dokumentasi Strands Agents.
- Integrasi MCP — Dukungan asli untuk Model Context Protocol (MCP), memungkinkan penyediaan konteks standar untuk LLMs operasi otonom yang konsisten.

- Layanan AWS integrasi - Koneksi tanpa batas ke Amazon Bedrock,, AWS Lambda AWS Step Functions, dan lainnya Layanan AWS untuk alur kerja otonom yang komprehensif. Untuk informasi lebih lanjut, lihat Roundup AWS Mingguan (AWS Blog).
- Pemilihan model foundation - Mendukung berbagai model pondasi termasuk Anthropic Claude, Amazon Nova (Premier, Pro, Lite, dan Micro) di Amazon Bedrock, dan lainnya untuk mengoptimalkan kemampuan penalaran otonom yang berbeda. Untuk informasi selengkapnya, lihat Amazon Bedrock di Strands Agents dokumentasi.
- Integrasi LLM API - Integrasi fleksibel dengan antarmuka layanan LLM yang berbeda termasuk Amazon Bedrock, OpenAI, dan lainnya untuk penyebaran produksi. Untuk informasi selengkapnya, lihat Amazon Bedrock Basic Usage dalam Strands Agents dokumentasi.
- Kemampuan multimodal — Dukungan untuk berbagai modalitas termasuk pemrosesan teks, ucapan, dan gambar untuk interaksi agen otonom yang komprehensif. Untuk informasi selengkapnya, lihat Amazon Bedrock Multimodal Support di dokumentasi. Strands Agents
- Ekosistem alat - Kumpulan alat yang kaya untuk Layanan AWS interaksi, dengan ekstensibilitas untuk alat khusus yang memperluas kemampuan otonom. Untuk informasi selengkapnya, lihat Ikhtisar Alat dalam Strands Agents dokumentasi.

Kapan harus menggunakan Strands Agents

Strands Agentssangat cocok untuk skenario agen otonom termasuk:

- Organizations yang membangun AWS infrastruktur yang menginginkan integrasi asli dengan Layanan AWS alur kerja otonom
- Tim yang memerlukan fitur keamanan, skalabilitas, dan kepatuhan tingkat perusahaan untuk sistem otonom produksi
- Proyek yang membutuhkan fleksibilitas dalam pemilihan model di berbagai penyedia untuk tugas otonom khusus
- Gunakan kasus yang memerlukan integrasi ketat dengan AWS alur kerja dan sumber daya yang ada untuk proses otonom ujung ke ujung

Pendekatan implementasi untuk Strands Agents

Strands Agentsmenyediakan pendekatan implementasi langsung untuk pemangku kepentingan bisnis, sebagaimana diuraikan dalam Panduan Memulai Cepat. Kerangka kerja ini memungkinkan organisasi untuk:

- Pilih model foundation seperti Amazon Nova (Premier, Pro, Lite, atau Micro) di Amazon Bedrock berdasarkan persyaratan bisnis tertentu.
- Tentukan alat khusus yang terhubung ke sistem perusahaan dan sumber data.
- Memproses beberapa modalitas termasuk teks, gambar, dan ucapan.
- Menyebarkan agen yang dapat secara mandiri menanggapi pertanyaan bisnis dan melakukan tugas.

Pendekatan implementasi ini memungkinkan tim bisnis untuk dengan cepat mengembangkan dan menyebarkan agen otonom tanpa keahlian teknis yang mendalam dalam pengembangan model AI.

Contoh dunia nyata dari Strands Agents

AWS Transform untuk penggunaan .NET Strands Agents untuk memperkuat kemampuan modernisasi aplikasinya, seperti yang dijelaskan dalam AWS Transform untuk .NET, layanan AI agen pertama untuk memodernisasi aplikasi.NET dalam skala besar (Blog).AWS Layanan produksi ini mempekerjakan beberapa agen otonom khusus. Para agen bekerja sama untuk menganalisis aplikasi.NET lama, merencanakan strategi modernisasi, dan mengeksekusi transformasi kode ke arsitektur cloud-native tanpa campur tangan manusia. AWS Transform untuk .NET menunjukkan kesiapan produksi Strands Agents untuk sistem otonom perusahaan.

LangChain dan LangGraph

LangChainadalah salah satu kerangka kerja paling mapan dalam ekosistem AI agen. LangGraphmemperluas kemampuannya untuk mendukung alur kerja agen yang kompleks dan stateful seperti yang dijelaskan dalam Blog. LangChain Bersama-sama, mereka memberikan solusi komprehensif untuk membangun agen AI otonom yang canggih dengan kemampuan orkestrasi yang kaya untuk operasi independen.

Fitur utama LangChain dan LangGraph

LangChaindan LangGraph termasuk fitur-fitur utama berikut:

- Ekosistem komponen — Perpustakaan ekstensif komponen pra-bangun untuk berbagai kemampuan agen otonom, memungkinkan pengembangan agen khusus yang cepat. Untuk informasi selengkapnya, lihat Mulai cepat di LangChain dokumentasi.
- Pemilihan model foundation - Support untuk beragam model pondasi termasuk Anthropic Claude, Amazon Nova model (Premier, Pro, Lite, dan Micro) di Amazon Bedrock, dan lainnya untuk

kemampuan penalaran yang berbeda. Untuk informasi selengkapnya, lihat Input dan output dalam dokumentasi. LangChain

- Integrasi API LLM — Antarmuka standar untuk beberapa penyedia layanan model bahasa besar (LLM) termasuk Amazon Bedrock,OpenAI, dan lainnya untuk penerapan yang fleksibel. Untuk informasi lebih lanjut, lihat LLMsdi LangChain dokumentasi.
- Pemrosesan multimodal - Dukungan bawaan untuk pemrosesan teks, gambar, dan audio untuk memungkinkan interaksi agen otonom multimodal yang kaya. Untuk informasi selengkapnya, lihat Multimodalitas dalam dokumentasi. LangChain
- Alur kerja berbasis grafik — LangGraph memungkinkan mendefinisikan perilaku agen otonom yang kompleks sebagai mesin negara, mendukung logika keputusan yang canggih. Untuk informasi selengkapnya, lihat pengumuman LangGraphPlatform GA.
- Abstraksi memori — Beberapa opsi untuk manajemen memori jangka pendek dan jangka panjang, yang penting untuk agen otonom yang mempertahankan konteks dari waktu ke waktu. Untuk informasi selengkapnya, lihat Cara menambahkan memori ke chatbots dalam dokumentasi. LangChain
- Integrasi alat — Ekosistem integrasi alat yang kaya di berbagai layanan dan APIs, memperluas kemampuan agen otonom. Untuk informasi selengkapnya, lihat Alat dalam LangChain dokumentasi.
- LangGraph platform - Solusi penyebaran dan pemantauan terkelola untuk lingkungan produksi, mendukung agen otonom yang sudah berjalan lama. Untuk informasi selengkapnya, lihat pengumuman LangGraphPlatform GA.

Kapan harus menggunakan LangChain dan LangGraph

LangChaindan LangGraph sangat cocok untuk skenario agen otonom termasuk:

- Alur kerja penalaran multi-langkah kompleks yang membutuhkan orkestrasi canggih untuk pengambilan keputusan otonom
- Proyek yang membutuhkan akses ke ekosistem besar komponen prebuilt dan integrasi untuk beragam kemampuan otonom
- Tim dengan infrastruktur dan keahlian machine learning (ML) Python berbasis yang ada yang ingin membangun sistem otonom
- Gunakan kasus yang memerlukan manajemen negara yang kompleks di seluruh sesi agen otonom yang berjalan lama

Pendekatan implementasi untuk LangChain dan LangGraph

LangChaindan LangGraph memberikan pendekatan implementasi terstruktur untuk pemangku kepentingan bisnis, sebagaimana dirinci dalam dokumentasi. LangGraph Kerangka kerja ini memungkinkan organisasi untuk:

- Tentukan grafik alur kerja canggih yang mewakili proses bisnis.
- Buat pola penalaran multi-langkah dengan poin keputusan dan logika bersyarat.
- Mengintegrasikan kemampuan pemrosesan multimodal untuk menangani beragam tipe data.
- Menerapkan kontrol kualitas melalui mekanisme peninjauan dan validasi bawaan.

Pendekatan berbasis grafik ini memungkinkan tim bisnis untuk memodelkan proses keputusan yang kompleks sebagai alur kerja otonom. Tim memiliki visibilitas yang jelas ke dalam setiap langkah proses penalaran dan kemampuan untuk mengaudit jalur keputusan.

Contoh dunia nyata dari dan LangChain LangGraph

Vodafonetelah menerapkan agen otonom menggunakan LangChain (danLangGraph) untuk meningkatkan rekayasa data dan alur kerja operasinya, sebagaimana dirinci dalam studi kasus LangChain Enterprise mereka. Mereka membangun asisten AI internal yang secara mandiri memantau metrik kinerja, mengambil informasi dari sistem dokumentasi, dan menyajikan wawasan yang dapat ditindaklanjuti — semuanya melalui interaksi bahasa alami.

VodafoneImplementasinya menggunakan pemuat dokumen LangChain modular, integrasi vektor, dan dukungan untuk beberapa LLMs (OpenAI, LLaMA 3, danGemini) untuk membuat prototipe dan benchmark dengan cepat. Mereka kemudian digunakan LangGraph untuk menyusun orkestrasi multi-agen dengan menggunakan sub agen modular. Agen-agen ini melakukan tugas pengumpulan, pemrosesan, peringkasan, dan penalaran. LangGraphmengintegrasikan agen-agen ini APIs ke dalam sistem cloud mereka.

CrewAI

CrewAIadalah kerangka kerja sumber terbuka yang berfokus secara khusus pada orkestrasi multi-agen otonom, tersedia di. GitHub Ini memberikan pendekatan terstruktur untuk menciptakan tim agen otonom khusus yang berkolaborasi untuk menyelesaikan tugas-tugas kompleks tanpa campur tangan manusia. CrewAImenekankan koordinasi berbasis peran dan delegasi tugas.

Fitur utama CrewAI

CrewAImenyediakan fitur utama berikut:

- Desain agen berbasis peran — Agen otonom didefinisikan dengan peran, tujuan, dan cerita belakang tertentu untuk memungkinkan keahlian khusus. Untuk informasi selengkapnya, lihat Membuat Agen Efektif dalam CrewAI dokumentasi.
- Delegasi tugas — Mekanisme bawaan untuk menetapkan tugas secara mandiri ke agen yang sesuai berdasarkan kemampuan mereka. Untuk informasi selengkapnya, lihat Tugas dalam CrewAI dokumentasi.
- Kolaborasi agen — Kerangka kerja untuk komunikasi antar-agen otonom dan berbagi pengetahuan tanpa mediasi manusia. Untuk informasi selengkapnya, lihat Kolaborasi dalam CrewAI dokumentasi.
- Manajemen proses — Alur kerja terstruktur untuk eksekusi tugas otonom sekuensial dan paralel. Untuk informasi selengkapnya, lihat Proses dalam CrewAI dokumentasi.
- Pemilihan model foundation - Support untuk berbagai model foundation termasuk Anthropic Claude, Amazon Nova model (Premier, Pro, Lite, dan Micro) di Amazon Bedrock, dan lainnya untuk mengoptimalkan tugas penalaran otonom yang berbeda. Untuk informasi lebih lanjut, lihat LLMsdi CrewAI dokumentasi.
- Integrasi API LLM - Integrasi fleksibel dengan beberapa antarmuka layanan LLM termasuk Amazon Bedrock,OpenAI, dan penerapan model lokal. Untuk informasi selengkapnya, lihat Contoh Konfigurasi Penyedia dalam CrewAI dokumentasi.
- Dukungan multimodal — Kemampuan yang muncul untuk menangani teks, gambar, dan modalitas lainnya untuk interaksi agen otonom yang komprehensif. Untuk informasi selengkapnya, lihat Menggunakan Agen Multimodal dalam CrewAI dokumentasi.

Kapan harus menggunakan CrewAI

CrewAIsangat cocok untuk skenario agen otonom termasuk:

- Masalah kompleks yang mendapat manfaat dari keahlian khusus berbasis peran yang bekerja secara mandiri
- Proyek yang membutuhkan kolaborasi eksplisit antara beberapa agen otonom
- Gunakan kasus di mana dekomposisi masalah berbasis tim meningkatkan pemecahan masalah otonom

- Skenario yang membutuhkan pemisahan kekhawatiran yang jelas antara peran agen otonom yang berbeda

Pendekatan implementasi untuk CrewAI

CrewAImenyediakan implementasi berbasis peran pendekatan tim agen AI untuk pemangku kepentingan bisnis, sebagaimana dirinci dalam Memulai dalam dokumentasi. CrewAI Kerangka kerja ini memungkinkan organisasi untuk:

- Tentukan agen otonom khusus dengan peran, tujuan, dan bidang keahlian tertentu.
- Tetapkan tugas kepada agen berdasarkan kemampuan khusus mereka.
- Menetapkan dependensi yang jelas antara tugas untuk membuat alur kerja terstruktur.
- Mengatur kolaborasi antara beberapa agen untuk memecahkan masalah yang kompleks.

Pendekatan berbasis peran ini mencerminkan struktur tim manusia, sehingga intuitif bagi para pemimpin bisnis untuk memahami dan menerapkan. Organizations dapat membuat tim otonom dengan bidang keahlian khusus yang berkolaborasi untuk mencapai tujuan bisnis, mirip dengan bagaimana tim manusia beroperasi. Namun, tim otonom dapat bekerja terus menerus tanpa campur tangan manusia.

Contoh dunia nyata dari CrewAI

AWS telah menerapkan sistem multi-agen otonom menggunakan CreWai yang terintegrasi dengan Amazon Bedrock, sebagaimana dirinci dalam studi kasus yang CrewAI diterbitkan. AWS dan CrewAI mengembangkan kerangka kerja yang aman dan netral vendor. Arsitektur “aliran dan kru” CrewAI sumber terbuka terintegrasi dengan mulus dengan model fondasi Amazon Bedrock, sistem memori, dan pagar pembatas kepatuhan.

Elemen kunci dari implementasi meliputi:

- Cetak biru dan sumber terbuka — AWS dan CrewAI merilis desain referensi yang memetakan agen CrewAI ke model Amazon Bedrock dan alat observabilitas. Mereka juga merilis sistem contoh seperti kru audit AWS keamanan multi-agen, arus modernisasi kode, dan otomatisasi back-office consumer packaged goods (CPG).
- Integrasi tumpukan observabilitas — Solusi ini menyematkan pemantauan dengan Amazon CloudWatch, dan AgentOpsLangFuse, memungkinkan penelusuran dan debugging dari pembuktian konsep hingga produksi.

- Demonstrated Return on Investment (ROI) — Pilot awal menunjukkan peningkatan besar — eksekusi 70 persen lebih cepat untuk proyek modernisasi kode besar dan sekitar 90 persen pengurangan waktu pemrosesan untuk aliran back-office CPG.

AutoGen

AutoGenadalah kerangka kerja sumber terbuka yang dirilis awalnya olehMicrosoft. AutoGenberfokus pada mengaktifkan agen AI otonom percakapan dan kolaboratif. Ini menyediakan arsitektur yang fleksibel untuk membangun sistem multi-agen dengan penekanan pada interaksi asinkron, berbasis peristiwa antara agen untuk alur kerja otonom yang kompleks.

Fitur utama AutoGen

AutoGenmenyediakan fitur utama berikut:

- Agen percakapan — Dibangun di sekitar percakapan bahasa alami antara agen otonom, memungkinkan penalaran canggih melalui dialog. Untuk informasi selengkapnya, lihat Kerangka Percakapan Multi-agen dalam AutoGen dokumentasi.
- Arsitektur asinkron — Desain berbasis peristiwa untuk interaksi agen otonom non-pemblokiran, mendukung alur kerja paralel yang kompleks. Untuk informasi selengkapnya, lihat Memecahkan Beberapa Tugas dalam Urutan Obrolan Async dalam dokumentasi. AutoGen
- H uman-in-the-loop — Dukungan kuat untuk partisipasi manusia opsional dalam alur kerja agen otonom bila diperlukan. Untuk informasi selengkapnya, lihat Mengizinkan Umpan Balik Manusia di Agen dalam AutoGen dokumentasi.
- Pembuatan dan eksekusi kode — Kemampuan khusus untuk agen otonom yang berfokus pada kode yang dapat menulis dan menjalankan kode. Untuk informasi selengkapnya, lihat Eksekusi Kode dalam AutoGen dokumentasi.
- Perilaku yang dapat disesuaikan - Konfigurasi agen otonom yang fleksibel dan kontrol percakapan untuk beragam kasus penggunaan. Untuk informasi selengkapnya, lihat agentchat.conversable_agent di dokumentasi. AutoGen
- Pemilihan model foundation — Support untuk berbagai model foundation termasuk Anthropic Claude, Amazon Nova model (Premier, Pro, Lite, dan Micro) di Amazon Bedrock, dan lainnya untuk kemampuan penalaran otonom yang berbeda. Untuk informasi selengkapnya, lihat Konfigurasi LLM dalam AutoGen dokumentasi.

- Integrasi API LLM - Konfigurasi standar untuk beberapa antarmuka layanan LLM termasuk Amazon Bedrock,, dan. OpenAI Azure OpenAI Untuk informasi selengkapnya, lihat oai.openai_utils di Referensi API. AutoGen
- Pemrosesan multimodal — Dukungan untuk pemrosesan teks dan gambar untuk memungkinkan interaksi agen otonom multimodal yang kaya. Untuk informasi lebih lanjut, lihat Terlibat dengan Model Multimodal: GPT-4V dalam dokumentasi. AutoGen AutoGen

Kapan harus menggunakan AutoGen

AutoGensangat cocok untuk skenario agen otonom termasuk:

- Aplikasi yang membutuhkan aliran percakapan alami antara agen otonom untuk penalaran yang kompleks
- Proyek yang membutuhkan operasi otonom penuh dan kemampuan pengawasan manusia opsional
- Gunakan kasus yang melibatkan pembuatan kode otonom, eksekusi, dan debugging tanpa campur tangan manusia
- Skenario yang membutuhkan pola komunikasi agen otonom yang fleksibel dan asinkron

Pendekatan implementasi untuk AutoGen

AutoGenmenyediakan pendekatan implementasi percakapan untuk pemangku kepentingan bisnis, sebagaimana dirinci dalam Memulai dalam dokumentasi. AutoGen Kerangka kerja ini memungkinkan organisasi untuk:

- Buat agen otonom yang berkomunikasi melalui percakapan bahasa alami.
- Menerapkan interaksi asinkron yang digerakkan oleh peristiwa antara beberapa agen.
- Gabungkan operasi otonom penuh dengan pengawasan manusia opsional bila diperlukan.
- Mengembangkan agen khusus untuk berbagai fungsi bisnis yang berkolaborasi melalui dialog.

Pendekatan percakapan ini membuat penalaran sistem otonom transparan dan dapat diakses oleh pengguna bisnis. Pengambil keputusan dapat mengamati dialog antara agen untuk memahami bagaimana kesimpulan dicapai dan secara opsional berpartisipasi dalam percakapan ketika penilaian manusia diperlukan.

Contoh dunia nyata dari AutoGen

Magentic-Oneadalah sistem multi-agen generalis open source yang dirancang untuk menyelesaikan tugas multi-langkah yang kompleks secara mandiri di berbagai lingkungan, seperti yang dijelaskan dalam blog AI Frontiers. Microsoft Pada intinya adalah agen Orchestrator, yang menguraikan tujuan tingkat tinggi dan melacak kemajuan dengan menggunakan buku besar terstruktur. Agen ini mendelegasikan subtugas kepada agen khusus (sepertiWebSurfer,, FileSurferCoder, danComputerTerminal) dan beradaptasi secara dinamis dengan perencanaan ulang bila diperlukan.

Sistem ini dibangun di atas AutoGen kerangka kerja dan model-agnostik, default ke GPT-4o. Ini mencapai kinerja mutakhir di seluruh tolok ukur seperti,, dan —semuanya tanpa penyetelan khusus tugas. GAIA AssistantBench WebArena Selain itu, mendukung ekstensibilitas modular dan evaluasi yang ketat melalui saran. AutoGenBench

LlamaIndex

LlamaIndexadalah kerangka data yang dirancang khusus untuk menghubungkan model bahasa besar (LLMs) dengan sumber data eksternal untuk memungkinkan Retrieval Augmented Generation (RAG) dan aplikasi AI agen yang canggih. Kerangka kerja ini menyediakan abstraksi dan alur kerja pengembangan yang dipercepat untuk sistem agen, pola orkestrasi khusus, dan integrasi sistem yang mengurangi solusi AI berbasis pengetahuan. time-to-production

Fitur utama LlamaIndex

LlamaIndexmenyediakan serangkaian kemampuan komprehensif yang membuatnya sangat cocok untuk aplikasi AI agen perusahaan:

- Arsitektur data-sentris — Unggul dalam menelan, mengindeks, dan mengambil informasi dari lebih dari 100 format data termasuk, dokumen Word, spreadsheet PDFs, Microsoft dan banyak lagi. Kerangka kerja ini mengubah data perusahaan menjadi basis pengetahuan yang dapat dikueri yang dioptimalkan untuk agen AI. Lihat informasi yang lebih lengkap dalam dokumentasi LlamaIndex.
- Penerapan siap produksi - LlamaIndex menawarkan kerangka kerja sumber terbuka dan layanan terkelola melaluiLlamaCloud, menyediakan fitur tingkat perusahaan termasuk kontrol keamanan, skalabilitas, integrasi observabilitas, dan fleksibilitas penerapan. Untuk informasi selengkapnya, lihat dokumentasi LlamaIndex kerangka kerja.
- Pemrosesan dokumen lanjutan — LlamaCloud menyediakan kemampuan penguraian dokumen, ekstraksi, pengindeksan, dan pengambilan yang menangani tata letak kompleks, tabel bersarang,

konten multi-modal, dan bahkan catatan tulisan tangan. Penguraian canggih ini memungkinkan agen untuk bekerja secara efektif dengan dokumen perusahaan dunia nyata yang berisi bagan, diagram, dan pemformatan kompleks. Lihat informasi yang lebih lengkap dalam dokumentasi LlamaCloud.

- Orkestrasi alur kerja — LlamaAgents menyediakan mesin orkestrasi async-first yang digerakkan oleh peristiwa untuk membangun sistem agen multi-langkah. Alur kerja mendukung pola kompleks termasuk loop, eksekusi paralel, percabangan bersyarat, dan dimulainya kembali stateful, menjadikannya ideal untuk interaksi agen yang canggih. Untuk informasi selengkapnya, lihat dokumentasi LlamaIndex alur kerja.
- Kemampuan pengambilan agen — Mode pengambilan lanjutan termasuk pencarian hibrida, pencarian semantik, dan perutean otomatis yang secara cerdas menentukan strategi pengambilan terbaik untuk setiap kueri. Kerangka kerja ini mendukung pengambilan komposit di beberapa basis pengetahuan dengan reranking untuk meningkatkan akurasi. Untuk informasi selengkapnya, lihat dokumentasi LlamaIndex RAG.
- Observabilitas dan evaluasi — LlamaIndex terintegrasi dengan berbagai alat observabilitas dan evaluasi. Kemampuan integrasi ini membantu Anda melacak dan men-debug aplikasi Anda, mengevaluasi kinerjanya, dan memantau biaya. Untuk informasi selengkapnya, lihat dokumentasi Tracing dan Debugging dan LlamaIndex Evaluating.

Kapan harus menggunakan LlamaIndex

LlamaIndexsangat cocok untuk skenario AI agen yang menekankan alur kerja intensif data dan manajemen pengetahuan:

- Aplikasi dokumen berat yang mengharuskan agen untuk memproses, menganalisis, dan mengekstrak wawasan dari sejumlah besar dokumen perusahaan seperti kontrak, laporan, manual, dan pengajuan peraturan
- Pembuatan prototipe cepat ke skenario produksi di mana organisasi ingin dengan cepat membangun dan menyebarkan agen yang berpusat pada dokumen tanpa overhead manajemen infrastruktur yang luas
- Arsitektur RAG-first yang memprioritaskan akurasi pengambilan dan relevansi konteks, terutama ketika bekerja dengan dokumen multi-modal yang kompleks yang berisi tabel, gambar, dan data terstruktur
- Alur kerja dokumen multi-agen yang memerlukan agen khusus untuk berbagai aspek pemrosesan dokumen, seperti penguraian, analisis, ringkasan, dan pemeriksaan kepatuhan

Pendekatan implementasi untuk LlamaIndex

LlamaIndex menyediakan blok bangunan tingkat rendah dan abstraksi tingkat tinggi yang mengakomodasi pendekatan implementasi yang berbeda:

- Perkembangan pesat aplikasi RAG fungsional hanya dalam beberapa baris kode dengan menggunakan LlamaIndex tingkat tinggi APIs. Pendekatan ini dapat LlamaIndex diakses oleh tim bisnis dan pengembang yang baru mengenal AI agen.
- Integrasi perusahaan melalui LlamaHub sistem perusahaan populer termasuk SharePoint, Amazon Simple Storage Service (Amazon S3), database, dan. APIs Pendekatan ini memungkinkan integrasi tanpa batas dengan infrastruktur data yang ada.
- Opsi penyebaran yang fleksibel antara penerapan yang di-host mandiri sumber terbuka untuk kontrol maksimum, atau layanan LlamaCloud terkelola untuk mengurangi overhead operasional dan fitur perusahaan.
- Aplikasi dapat dimulai dengan mesin kueri sederhana dan secara progresif menambahkan kemampuan agen, orkestrasi multi-agen, dan alur kerja yang kompleks seiring dengan berkembangnya persyaratan.

Contoh dunia nyata dari LlamaIndex

Contoh ini berfokus pada anak perusahaan dari perusahaan kedirgantaraan yang berspesialisasi dalam navigasi penerbangan dan solusi operasi. Mereka perlu mengatasi tantangan yang berkembang yang melibatkan uji coba chatbot AI yang tidak terkoordinasi. Uji coba menghasilkan pekerjaan berulang, siklus pengembangan yang panjang, hambatan kepatuhan, dan implementasi yang terisolasi di seluruh organisasi.

Mereka mengembangkan kerangka agen terpadu, solusi berbasis template yang dapat digunakan kembali yang dibangun di atas kerangka kerja LlamaIndex sumber terbuka yang membuat pembuatan agen jauh lebih efisien. Mereka membandingkan beberapa kerangka kerja yang bersaing, baik berorientasi rantai maupun berbasis grafik. Pada akhirnya, mereka LlamaIndex memilih tiga keunggulan penting: desainnya yang fleksibel, komponen modular, dan kontrol orkestrasi siap produksi.

Platform ini mengurangi waktu pengembangan dan penyebaran agen sebesar 87% dari 512 menjadi 64 jam. Pengurangan ini dicapai dengan memungkinkan tim untuk membangun agen dengan sekitar 50 baris kode dan file konfigurasi JSON. Tim memanfaatkan kerangka kerja terpadu dengan

keamanan bawaan, kepatuhan, dan akses sistem istimewa. Untuk detail lebih lanjut, lihat studi kasus LlamaIndex pelanggan.

Membandingkan kerangka kerja AI agen

Saat memilih kerangka kerja AI agen untuk pengembangan agen otonom, pertimbangkan bagaimana setiap opsi selaras dengan persyaratan spesifik Anda. Pertimbangkan tidak hanya kemampuan teknisnya tetapi juga kecocokan organisasinya, termasuk keahlian tim, infrastruktur yang ada, dan persyaratan pemeliharaan jangka panjang. Banyak organisasi mungkin mendapat manfaat dari pendekatan hibrida, memanfaatkan beberapa kerangka kerja untuk berbagai komponen ekosistem AI otonom mereka.

Tabel berikut membandingkan tingkat kematangan (terkuat, kuat, memadai, atau lemah) dari setiap kerangka kerja di seluruh dimensi teknis utama. Untuk setiap kerangka kerja, tabel juga mencakup informasi tentang opsi penerapan produksi dan kompleksitas kurva pembelajaran.

Kerangka	AWS integrasi	Dukungan multi-agen otonom	Kompleksitas alur kerja otonom	Kemampuan multimodal	Pemilihan model pondasi	Integrasi API LLM	Penyebaran produksi	Kurva belajar
AutoGen	Lemah	Kuat	Kuat	Memadai	Memadai	Kuat	Lakukan sendiri (DIY)	Curam
CrewAI	Lemah	Kuat	Memadai	Lemah	Memadai	Memadai	DIY	Sedang
LangChain / LangGraph	Memadai	Kuat	Terkuat	Terkuat	Terkuat	Terkuat	Platform atau DIY	Curam
LlamaIndex	Memadai	Memadai	Kuat	Memadai	Kuat	Kuat	Platform atau DIY	Sedang
Strands Agents	Terkuat	Kuat	Terkuat	Kuat	Kuat	Terkuat	DIY	Sedang

Pertimbangan dalam memilih kerangka kerja AI agen

Saat mengembangkan agen otonom, pertimbangkan faktor-faktor kunci berikut:

- AWS Integrasi infrastruktur — Organizations yang banyak diinvestasikan AWS akan mendapat manfaat paling besar dari integrasi asli Strands Agents dengan Layanan AWS untuk alur kerja otonom. Untuk informasi lebih lanjut, lihat Roundup AWS Mingguan (AWS Blog).
- Pemilihan model foundation - Pertimbangkan kerangka kerja mana yang memberikan dukungan terbaik untuk model fondasi pilihan Anda (misalnya, model Amazon Nova di Amazon Bedrock atau Anthropic Claude), berdasarkan persyaratan penalaran agen otonom Anda. Untuk informasi lebih lanjut, lihat Membangun Agen Efektif di Anthropic situs web.
- Integrasi API LLM - Evaluasi kerangka kerja berdasarkan integrasinya dengan antarmuka layanan model bahasa besar (LLM) pilihan Anda (misalnya, Amazon Bedrock atauOpenAI) untuk penerapan produksi. Untuk informasi selengkapnya, lihat Antarmuka Model dalam Strands Agents dokumentasi.
- Persyaratan multimodal — Untuk agen otonom yang perlu memproses teks, gambar, dan ucapan, pertimbangkan kemampuan multimodal dari setiap kerangka kerja. Untuk informasi selengkapnya, lihat Multimodalitas dalam dokumentasi. LangChain
- Kompleksitas alur kerja otonom - Alur kerja otonom yang lebih kompleks dengan manajemen negara yang canggih mungkin mendukung kemampuan mesin negara bagian yang canggih. LangGraph
- Kolaborasi tim otonom — Proyek yang membutuhkan kolaborasi otonom berbasis peran eksplisit antara agen khusus dapat memperoleh manfaat dari arsitektur berorientasi tim. CrewAI
- Paradigma pengembangan otonom — Tim yang lebih menyukai pola percakapan dan asinkron untuk agen otonom mungkin lebih menyukai arsitektur berbasis peristiwa. AutoGen
- Pendekatan terkelola atau berbasis kode - Organizations yang menginginkan pengalaman yang dikelola sepenuhnya dengan pengkodean minimal harus mempertimbangkan Amazon Bedrock Agents. Organizations yang memerlukan kustomisasi lebih dalam mungkin lebih suka Strands Agents atau kerangka kerja lain dengan kemampuan khusus yang lebih selaras dengan persyaratan agen otonom tertentu.
- Kesiapan produksi untuk sistem otonom — Pertimbangkan opsi penyebaran, kemampuan pemantauan, dan fitur perusahaan untuk agen otonom produksi.

Platform

Platform AI Agentic menyediakan lapisan runtime, orkestrasi, dan integrasi dasar yang diperlukan untuk menerapkan, menskalakan, dan mengelola sistem agen tingkat produksi. Kerangka kerja mendefinisikan bagaimana agen dibangun dan protokol mengatur bagaimana mereka berkomunikasi. Platform menyediakan lingkungan di mana agen-agen ini beroperasi, berkolaborasi, dan berkembang dengan aman dalam skala besar.

Platform agen menggabungkan eksekusi model, manajemen konteks, integrasi alat, observabilitas, dan kemampuan tata kelola ke dalam lingkungan terpadu. Platform ini memungkinkan organisasi untuk beralih dari eksperimen ke penyebaran skala perusahaan.

Di bagian ini:

Mengapa platform penting

Platform AI agen sangat penting bagi organisasi yang ingin mengoperasionalkan sistem otonom dalam produksi. Mereka menawarkan kemampuan berikut:

- Menyediakan orkestrasi runtime untuk hosting, penskalaan, dan agen koordinasi.
- Kelola status, konteks, dan memori di seluruh alur kerja multi-agen.
- Menawarkan kontrol keamanan, identitas, dan tata kelola yang selaras dengan standar perusahaan.
- Integrasikan dengan ekosistem perkakas dan sistem eksternal melalui standar APIs atau protokol.
- Aktifkan observabilitas dan auditabilitas di seluruh interaksi agen dan arus peristiwa.
- Mendukung interoperabilitas lintas model, memungkinkan agen untuk menggunakan beberapa model pondasi dalam satu lingkungan.

Kemampuan ini mengubah agen individu menjadi sistem adaptif yang terkoordinasi yang dapat beroperasi dengan andal dalam batas perusahaan dan peraturan.

Jenis platform AI agen

Platform AI agen biasanya termasuk dalam satu atau lebih kategori berikut:

- Agen terkelola - Platform yang dikelola sepenuhnya menyediakan infrastruktur, memori, dan kemampuan orkestrasi bawaan. Mereka mengurangi overhead operasional dan mempercepat waktu produksi.
- Orkestrasi sumber terbuka — Platform agen sumber terbuka menawarkan fleksibilitas dan transparansi bagi organisasi yang lebih menyukai lingkungan yang dapat disesuaikan atau penerapan lokal.
- Hybrid Enterprise — Platform hybrid mengintegrasikan komponen yang dikelola dan di-host sendiri, menggabungkan skalabilitas layanan yang dikelola cloud dengan kontrol sistem perusahaan.

Pertimbangan pemilihan platform

Saat memilih atau merancang platform AI agen, organisasi harus mempertimbangkan hal berikut:

- Integration depth — Mengevaluasi seberapa baik platform terintegrasi dengan sumber data, alat, dan protokol yang ada.
- Skalabilitas — Pastikan platform dapat menskalakan secara dinamis untuk mendukung beban kerja otonom dan kolaborasi multi-agen.
- Keamanan dan kepatuhan - Menilai privasi data, enkripsi, dan fitur tata kelola terhadap persyaratan organisasi dan regional.
- Ekstensibilitas — Pilih platform dengan arsitektur modular yang memungkinkan alat, model, atau agen baru ditambahkan dari waktu ke waktu.
- Observabilitas — Lebih suka platform yang menyediakan telemetri terperinci, keterlacakan, dan log audit untuk interaksi agen.
- Efisiensi biaya - Pertimbangkan model tanpa server atau berbasis penggunaan untuk mengoptimalkan biaya untuk beban kerja variabel.

Agen Batuan Dasar Amazon

Amazon Bedrock Agents adalah layanan terkelola penuh yang memungkinkan Anda membangun dan mengonfigurasi agen otonom dalam aplikasi Anda. Ini dapat mengatur interaksi antara model dasar, sumber data, aplikasi perangkat lunak, dan percakapan pengguna. Pendekatannya yang efisien untuk membuat agen tidak mengharuskan Anda menyediakan kapasitas, mengelola infrastruktur, atau menulis kode khusus.

Fitur utama dari Amazon Bedrock Agents

Amazon Bedrock Agents mencakup fitur-fitur utama berikut:

- Layanan yang dikelola sepenuhnya - Manajemen infrastruktur lengkap tanpa perlu menyediakan kapasitas atau mengelola sistem yang mendasarinya. Untuk informasi selengkapnya, lihat Mengotomatiskan tugas di aplikasi Anda menggunakan agen AI di dokumentasi Amazon Bedrock.
- Pengembangan berbasis API — Tentukan dan jalankan agen melalui panggilan API sederhana dengan menentukan model, instruksi, alat, dan parameter konfigurasi. Untuk informasi selengkapnya, lihat Membuat dan mengonfigurasi agen secara manual di dokumentasi Amazon Bedrock.
- Grup tindakan — Tentukan tindakan spesifik yang dapat dilakukan agen Anda dengan membuat grup tindakan dengan skema API. Untuk informasi selengkapnya, lihat Menggunakan grup tindakan untuk menentukan tindakan yang akan dilakukan agen Anda di dokumentasi Amazon Bedrock.
- Integrasi basis pengetahuan — Terhubung dengan mulus ke Pangkalan Pengetahuan Amazon Bedrock untuk meningkatkan respons agen dengan data organisasi Anda. Untuk informasi selengkapnya, lihat Pembuatan respons tambahan untuk agen Anda dengan basis pengetahuan di dokumentasi Amazon Bedrock.
- Template prompt lanjutan - Sesuaikan perilaku agen melalui templat prompt untuk pra-pemrosesan, orkestrasi, pembuatan respons basis pengetahuan, dan pasca-pemrosesan. Untuk informasi selengkapnya, lihat Meningkatkan akurasi agen menggunakan templat prompt lanjutan di Amazon Bedrock di dokumentasi Amazon Bedrock.
- Penelusuran dan pengamatan - Lacak proses step-by-step penalaran agen menggunakan kemampuan penelusuran bawaan. Untuk informasi selengkapnya, lihat Lacak proses step-by-step penalaran agen menggunakan jejak di dokumentasi Amazon Bedrock.

- Pembuatan versi dan alias — Buat beberapa versi agen Anda dan terapkan melalui alias untuk peluncuran terkontrol. Untuk informasi selengkapnya, lihat Menerapkan dan menggunakan agen Amazon Bedrock di aplikasi Anda di dokumentasi Amazon Bedrock.

Kapan menggunakan Amazon Bedrock Agents

Amazon Bedrock Agents sangat cocok untuk skenario agen otonom termasuk:

- Organizations yang menginginkan pengalaman yang dikelola sepenuhnya untuk membangun dan menyebarkan agen tanpa mengelola infrastruktur
- Proyek yang membutuhkan pengembangan dan penyebaran agen yang cepat melalui konfigurasi daripada kode
- Kasus penggunaan yang mendapat manfaat dari integrasi ketat dengan kemampuan Amazon Bedrock lainnya seperti Pangkalan Pengetahuan dan Pagar Pembatas
- Tim tanpa sumber daya internal untuk membangun agen dari awal tetapi membutuhkan kemampuan otonom siap produksi

Pendekatan implementasi untuk Amazon Bedrock Agents

Amazon Bedrock Agents menyediakan pendekatan implementasi berbasis konfigurasi untuk pemangku kepentingan bisnis. Layanan ini memungkinkan organisasi untuk:

- Tentukan agen melalui panggilan Konsol Manajemen AWS atau API tanpa menulis kode yang rumit.
- Buat grup tindakan yang menentukan APIs dan operasi yang dapat dilakukan agen.
- Hubungkan basis pengetahuan untuk memberikan informasi spesifik domain kepada agen.
- Uji dan ulangi perilaku agen melalui antarmuka visual.

Pendekatan terkelola ini memungkinkan tim bisnis untuk dengan cepat mengembangkan dan menyebarkan agen otonom tanpa keahlian teknis yang mendalam dalam pengembangan model AI atau manajemen infrastruktur.

Contoh dunia nyata dari Amazon Bedrock Agents

Solusi operasi keuangan (FinOps) yang dijelaskan dalam posting AWS blog ini menggunakan kerangka kerja multi-agen Amazon Bedrock untuk membuat asisten manajemen biaya cloud berbasis

AI. Model yayasan Amazon Nova yang hemat biaya mendukung solusi di mana agen FinOps Supervisor pusat mendelegasikan tugas kepada agen khusus. Agen ini mengambil dan menganalisis data AWS pengeluaran dengan menggunakan AWS Cost Explorer dan menghasilkan rekomendasi penghematan biaya dengan menggunakan. AWS Trusted Advisor

Sistem ini mencakup akses pengguna yang aman melalui Amazon Cognito, front-end yang dihosting, dan grup AWS Lambda tindakan untuk AWS Amplify analisis dan peramalan waktu nyata. Tim keuangan dapat mengajukan pertanyaan bahasa alami seperti “Berapa biaya saya pada Februari 2025?” Sistem merespons dengan perincian terperinci, saran pengoptimalan, dan prakiraan — semuanya dalam arsitektur tanpa server yang dapat diskalakan yang digunakan dengan menggunakan. AWS CloudFormation

Batuan Dasar Amazon AgentCore

Amazon Bedrock AgentCore adalah platform agen untuk membangun, menyebarkan, dan mengoperasikan agen berkemampuan tinggi dengan aman dalam skala menggunakan kerangka kerja, model, atau protokol apa pun. Menggunakan AgentCore, Anda dapat melakukan hal berikut, semua tanpa manajemen infrastruktur:

- Membangun agen lebih cepat.
- Memungkinkan agen untuk mengambil tindakan di seluruh alat dan data.
- Jalankan agen dengan aman dengan latensi rendah dan runtime yang diperpanjang.
- Memantau agen dalam produksi.

AgentCore menghilangkan pengangkatan berat yang tidak berdiferensiasi dari membangun infrastruktur agen khusus, memungkinkan Anda mempercepat agen Anda ke produksi. Layanannya dapat digunakan bersama atau secara independen dan kompatibel dengan kerangka kerja apa pun, termasukCrewAI,LangGraph,LlamaIndex, danStrands Agents. AgentCorejuga kompatibel dengan model foundation apa pun yang tersedia di dalam atau di luar Amazon Bedrock, memberikan fleksibilitas tertinggi.

AgentCore terdiri dari beberapa layanan utama:

- Amazon Bedrock AgentCore Runtime - Menyediakan lingkungan yang aman, tanpa server, dan dapat diskalakan untuk meng-host dan menjalankan agen Anda, tanpa perlu mengelola infrastruktur apa pun yang diperlukan untuk menerapkan dan menjalankan agen atau alat AI.

- Amazon Bedrock AgentCore Memory - Menawarkan sistem memori terkelola, memungkinkan agen untuk mempertahankan konteks dari interaksi untuk percakapan yang lebih personal dan koheren dengan mempertahankan pengetahuan langsung dan jangka panjang.
- Amazon Bedrock AgentCore Gateway — Menyederhanakan proses pembuatan, pengamanan, dan menemukan alat yang tepat untuk agen. Dengan AgentCore Gateway, pengembang dapat mengonversi APIs, fungsi Lambda, dan layanan yang ada menjadi alat yang kompatibel dengan Model Context Protocol (MCP) dan membuatnya tersedia untuk agen.
- Amazon Bedrock AgentCore Identity - Menyediakan identitas agen yang aman dan terukur serta layanan manajemen akses yang mempercepat pengembangan agen AI. Dengan AgentCore Identity, Anda dapat menetapkan identitas unik dan dapat diverifikasi ke agen, memungkinkan kontrol akses yang halus dan interaksi yang didukung agen yang aman dengan sistem perusahaan.
- Alat AgentCore bawaan Amazon Bedrock - Memungkinkan Anda menggunakan alat bawaan untuk meningkatkan alur kerja pengembangan dan pengujian Anda. Gunakan alat ini untuk berinteraksi dengan aplikasi Anda secara efektif, memungkinkan agen AI untuk menulis dan mengeksekusi kode dengan aman di lingkungan kotak pasir. Gunakan alat browser untuk memungkinkan agen AI berinteraksi dengan situs web dalam skala besar.
- Amazon Bedrock AgentCore Observability — Memberikan kemampuan pencatatan dan pemantauan, memberi Anda visibilitas real-time ke kinerja dan perilaku agen Anda untuk memfasilitasi debugging dan pengoptimalan.

Fitur utama AgentCore

AgentCore termasuk fitur utama berikut:

- Dikelola sepenuhnya dan dapat diperluas - AgentCore adalah layanan yang dikelola sepenuhnya, yang berarti AWS menangani infrastruktur dan pemeliharaan yang mendasarinya. Ini juga dapat diperluas, yang memungkinkan Anda untuk menyesuaikan dan meningkatkan fungsionalitas agen Anda. Untuk informasi selengkapnya, lihat Memulai AgentCore Runtime di AgentCore dokumentasi.
- Memori jangka panjang dan jangka pendek — Memberikan interaksi yang lebih personal dan relevan dengan melengkapi agen dengan sistem memori untuk mengingat konteks dari percakapan saat ini dan pengetahuan jangka panjang. Untuk informasi selengkapnya, lihat Memulai AgentCore Memori di AgentCore dokumentasi.

- Pengembangan dan integrasi alat yang disederhanakan - Memungkinkan agen Anda menemukan dan menggunakan alat melalui satu titik akhir yang aman. Ubah sumber daya perusahaan Anda dengan cepat menjadi alat yang siap agen hanya dengan beberapa baris kode, membebaskan pengembang untuk fokus membangun kemampuan unik. Untuk informasi selengkapnya, lihat Memulai AgentCore Gateway di AgentCore dokumentasi.
- Infrastruktur yang aman dan terukur - AgentCore menyediakan lingkungan yang aman dan terukur untuk menyebarkan dan mengoperasikan agen. Ini mencakup fitur untuk identitas dan manajemen akses, enkripsi data, dan keamanan jaringan. Untuk informasi selengkapnya, lihat Memulai AgentCore Identitas dalam AgentCore dokumentasi.
- Integrasi dengan berbagai alat - Memungkinkan Anda mengintegrasikan agen Anda dengan berbagai alat, termasuk penerjemah kode dan alat browser yang dapat Anda buat dengan menggunakan alat AgentCore bawaan. Untuk informasi selengkapnya, lihat Memulai dengan Penerjemah AgentCore Kode dan Memulai AgentCore Browser di AgentCore dokumentasi.
- Pengamatan dan pemantauan komprehensif — Dapatkan visibilitas mendalam ke agen Anda dengan alat komprehensif untuk melacak, men-debug, dan memantau kinerja mereka dalam produksi. Visualisasikan seluruh jalur eksekusi agen untuk mengaudit alasannya dan menyelesaikan kegagalan. Gunakan dasbor real-time dan data telemetri standar untuk melacak metrik operasional utama. Untuk informasi selengkapnya, lihat Menambahkan observabilitas ke AgentCore resource Amazon Bedrock Anda di dokumentasi. AgentCore

Kapan harus menggunakan AgentCore

AgentCore sangat cocok untuk skenario agen otonom termasuk:

- Organizations yang ingin mempercepat pengembangan dan menurunkan biaya operasional dengan layanan terkelola penuh yang menangani infrastruktur, keamanan, alat bawaan, observabilitas, dan penskalaan
- Proyek yang membutuhkan fleksibilitas dengan layanan modular yang bekerja sama atau independen dan kompatibel dengan kerangka kerja apa pun, seperti CrewAI atauLangGraph, dan model fondasi apa pun dari sumber mana pun
- Gunakan kasus yang membutuhkan agen percakapan stateful yang perlu mempertahankan konteks dan belajar dari interaksi masa lalu untuk memberikan tanggapan yang dipersonalisasi dan relevan
- Agen memungkinkan untuk melakukan tugas-tugas kompleks melalui integrasi sederhana dengan beragam aplikasi, sumber data, dan APIs

Pendekatan implementasi untuk AgentCore

AgentCore dirancang untuk organisasi yang ingin memindahkan agen AI dari bukti konsep, dibangun menggunakan kerangka kerja open source atau agen kustom, ke produksi. Dengan AgentCore, organisasi dapat melakukan hal berikut:

- Menyebarkan agen secara aman pada infrastruktur tanpa server, mendukung kerangka kerja dan model apa pun, dengan isolasi sesi dan identitas bawaan serta manajemen akses untuk end-to-end keamanan dan kepatuhan. Buat agen AgentCore Runtime dengan cepat untuk kerangka kerja agen terkemuka dengan menggunakan toolkit starter.
- Tingkatkan agen dengan mengintegrasikan memori persisten untuk retensi konteks, menyederhanakan pengembangan dan integrasi alat melalui AgentCore Gateway. Manfaatkan alat browser bawaan dan penerjemah kode untuk alur kerja tingkat lanjut.
- Lacak, debug, dan pantau agen AI dalam produksi menggunakan dasbor observabilitas yang didukung oleh Amazon CloudWatch Application Insights danOpenTelemetry, melacak metrik utama AgentCore sumber daya (runtime, memori, gateway, dan alat).
- Mempercepat penyebaran dan inovasi dengan layanan modular yang dikelola sepenuhnya, blok yang dapat dikomposisi bersama-sama atau secara independen, dengan kerangka kerja agen dan penyedia model apa pun. Fleksibilitas ini membantu organisasi beralih dari prototipe ke produksi lebih cepat.

Pendekatan terkelola ini memungkinkan organisasi untuk dengan cepat dan aman membangun, menyebarkan, dan menjalankan agen AI tingkat perusahaan dan sistem multi-agen pada skala apa pun.

Contoh dunia nyata AgentCore

AWS Telah mengamati bahwa salah satu bank terbesar di Amerika Latin telah digunakan AI/ML selama bertahun-tahun untuk memberikan pengalaman perbankan digital yang sangat personal dan aman. Bank memperluas layanan AI agen dengan menggunakan AgentCore untuk menyediakan pelanggan dengan interaksi intuitif, keamanan yang ditingkatkan, dan otomatisasi yang lebih besar. Menurut CTO, AgentCore diharapkan dapat mendukung upaya mereka untuk memenuhi komitmen pelanggan dalam skala besar. AgentCore memberikan pengembang mereka alat dan fleksibilitas untuk membangun dan mengelola agen, sambil membantu memastikan kepatuhan terhadap peraturan keuangan.

Protokol

Agen AI memerlukan protokol komunikasi standar untuk berinteraksi dengan agen dan layanan lain. Organizations menerapkan arsitektur agen menghadapi tantangan signifikan seputar interoperabilitas, independensi vendor, dan pemeriksaan masa depan investasi mereka.

Bagian ini membantu Anda menavigasi lanskap agent-to-agent protokol dengan fokus pada standar terbuka yang memaksimalkan fleksibilitas dan interoperabilitas. (Untuk informasi tentang agent-to-tool protokol, lihat Strategi integrasi alat nanti dalam panduan ini.)

Bagian ini menyoroti Model Context Protocol (MCP), standar terbuka yang awalnya dikembangkan pada Anthropic tahun 2024. Hari ini, AWS secara aktif mendukung MCP melalui kontribusi untuk pengembangan dan implementasi protokol. AWS Berkolaborasi dengan kerangka kerja agen open-source terkemuka, termasuk,, dan LangGraph CrewAILlamaIndex, untuk membentuk masa depan komunikasi antar-agen pada protokol. Untuk informasi selengkapnya, lihat Protokol Terbuka untuk Interoperabilitas Agen Bagian 1: Komunikasi Antar Agen di MCP (Blog).AWS

Di bagian ini:

- Mengapa pemilihan protokol penting
- Agent-to-agent protokol
- Memilih protokol agen
- Strategi implementasi untuk protokol agen
- Memulai dengan MCP
- ???

Mengapa pemilihan protokol penting

Pemilihan protokol secara fundamental membentuk bagaimana Anda dapat membangun dan mengembangkan arsitektur agen AI Anda. Dengan memilih protokol yang mendukung portabilitas antar kerangka kerja agen, Anda mendapatkan fleksibilitas untuk menggabungkan berbagai sistem agen dan alur kerja untuk memenuhi kebutuhan spesifik Anda.

Protokol terbuka memungkinkan Anda mengintegrasikan agen di beberapa kerangka kerja. Misalnya, gunakan LangChain untuk prototyping cepat dan implementasikan sistem produksi denganStrands Agents, berkomunikasi melalui protokol umum, seperti MCP atau protokol Agent2Agent (A2A). Fleksibilitas ini mengurangi ketergantungan pada penyedia AI tertentu, menyederhanakan integrasi

dengan sistem yang ada, dan memungkinkan Anda meningkatkan kemampuan agen dari waktu ke waktu.

Protokol yang dirancang dengan baik juga menetapkan pola keamanan yang konsisten untuk otentikasi dan otorisasi di seluruh ekosistem agen Anda. Yang terpenting, portabilitas protokol menjaga kebebasan Anda untuk mengadopsi kerangka kerja dan kemampuan agen baru saat muncul. Memilih protokol terbuka melindungi investasi Anda dalam pengembangan agen sambil mempertahankan interoperabilitas dengan sistem pihak ketiga.

Keuntungan dari protokol terbuka

Saat menerapkan ekstensi Anda sendiri atau membangun sistem agen kustom, protokol terbuka menawarkan keuntungan yang menarik:

- Dokumentasi dan transparansi — Biasanya menyediakan dokumentasi yang komprehensif dan implementasi transparan
- Dukungan komunitas — Akses ke komunitas pengembang yang lebih luas untuk pemecahan masalah dan praktik terbaik
- Jaminan interoperabilitas - Jaminan yang lebih baik bahwa ekstensi Anda akan bekerja di berbagai implementasi
- Kompatibilitas di masa mendatang - Mengurangi risiko melanggar perubahan atau penghentian
- Pengaruh pada pembangunan — Kesempatan untuk berkontribusi pada evolusi protokol

Agent-to-agent protokol

Tabel berikut memberikan ikhtisar protokol agen yang memungkinkan beberapa agen untuk berkolaborasi, mendelegasikan tugas, dan berbagi informasi.

Protokol	Ideal untuk	Pertimbangan-pertimbangan
Komunikasi antar agen MCP	Organizations mencari pola kolaborasi agen yang fleksibel	• Perpanjangan untuk Model Context Protocol (MCP) yang diusulkan oleh AWS yang dibangun di atas fondasi yang ada untuk komunikasi agent-to-agent

		• Memungkinkan kolaboras i agen yang mulus dengan keamanan OAuth berbasis
Protokol A2A	Ekosistem agen lintas platform	• Didukung oleh Google • Standar yang lebih baru dengan adopsi yang lebih terbatas dibandingkan dengan MCP

Memutuskan di antara opsi protokol

Saat menerapkan agent-to-agent komunikasi, sesuaikan persyaratan komunikasi spesifik Anda dengan kemampuan protokol yang sesuai. Pola interaksi yang berbeda memerlukan fitur protokol yang berbeda. Tabel berikut menguraikan pola komunikasi umum dan merekomendasikan pilihan protokol yang paling sesuai untuk setiap skenario.

Pola	Deskripsi	Pilihan protokol yang ideal
Permintaan dan tanggapan sederhana	Interaksi satu kali antar agen	MCP dengan aliran stateless
Dialog stateful	Percakapan yang sedang berlangsung dengan konteks	MCP dengan manajemen sesi
Kolaborasi multi-agen	Interaksi kompleks antara beberapa agen	MCP antar-agen atau AutoGen
Alur kerja berbasis tim	Tim agen hierarkis dengan peran yang ditentukan	MCP antar-agen,, atau CrewAI AutoGen

Di luar pola komunikasi, beberapa faktor teknis dan organisasi dapat memengaruhi pemilihan protokol Anda. Tabel berikut menguraikan pertimbangan utama yang dapat membantu Anda mengevaluasi protokol mana yang paling sesuai dengan persyaratan implementasi spesifik Anda.

Pertimbangan	Deskripsi	Contoh

Model keamanan	Persyaratan otentikasi dan otorisasi	OAuth 2.0 di MCP
Lingkungan penyebaran	Dimana agen akan menjalankan dan berkomunikasi	Mesin terdistribusi atau tunggal
Kompatibilitas ekosistem	Integrasi dengan kerangka kerja agen yang ada	LangChain atau Strands Agents
Kebutuhan skalabilitas	Pertumbuhan yang diharapkan dalam interaksi agen	Kemampuan streaming MCP

Memilih protokol agen

Untuk sebagian besar organisasi yang membangun sistem agen produksi, Model Context Protocol (MCP) menawarkan fondasi komunikasi yang paling komprehensif dan didukung dengan baik. agent-to-agent MCP mendapat manfaat dari kontribusi pengembangan aktif dari AWS dan komunitas open-source.

Memilih protokol agen yang tepat penting bagi organisasi yang ingin menerapkan AI agen secara efektif. Pertimbangan berbeda berdasarkan konteks organisasi.

Pertimbangan pemilihan protokol agen

Organizations harus mempertimbangkan praktik terbaik berikut saat memilih protokol untuk sistem AI agen:

- Prioritaskan standar terbuka — Organizations harus mengadopsi protokol terbuka seperti MCP untuk membantu memastikan interoperabilitas jangka panjang, ekstensibilitas, dan untuk mengurangi risiko penguncian vendor.
- Menyeimbangkan kecepatan dan fleksibilitas — Startup dan pengadopsi awal dapat dimulai dengan protokol kepemilikan yang didukung dengan baik untuk pengembangan yang cepat tetapi harus menentukan jalur migrasi ke standar terbuka saat sistem matang.
- Menerapkan lapisan abstraksi — Perusahaan harus menerapkan abstraksi protokol untuk menyederhanakan migrasi, mengaktifkan adopsi hibrida, dan strategi integrasi masa depan.

- Menekankan keamanan dan kepatuhan — Organizations dalam industri yang diatur harus memilih protokol dengan otentikasi, enkripsi, dan kemampuan audit yang kuat untuk memenuhi persyaratan tata kelola dan kepatuhan.
- Evaluasi kematangan ekosistem — Semua organisasi harus menilai kesehatan, adopsi, dan dukungan masyarakat dari setiap protokol untuk memastikan keberlanjutan dan meminimalkan utang teknis.
- Terlibat dalam pengembangan standar - Organizations harus berpartisipasi dalam badan standar atau komunitas sumber terbuka untuk membantu membentuk evolusi protokol dan memengaruhi praktik terbaik.
- Perhitungan kedaulatan data — Pemerintah dan sektor yang diatur harus memastikan pilihan protokol selaras dengan persyaratan residensi dan kedaulatan data di seluruh wilayah penyebaran.
- Manfaatkan layanan terkelola — Jika memungkinkan, gunakan implementasi protokol agen yang dikelola atau tanpa server untuk mengurangi kompleksitas operasional dan mempercepat penyebaran.

Strategi implementasi untuk protokol agen

Untuk menerapkan protokol agen secara efektif di seluruh organisasi Anda, pertimbangkan langkah-langkah strategis berikut:

1. Mulai dengan penyelarasan standar - Mengadopsi protokol terbuka yang sudah mapan jika memungkinkan.
2. Buat lapisan abstraksi — Menerapkan adaptor antara sistem Anda dan protokol tertentu.
3. Berkontribusi pada standar terbuka - Berpartisipasi dalam komunitas pengembangan protokol.
4. Pantau evolusi protokol —Tetap terinformasi tentang standar dan pembaruan yang muncul.
5. Uji interoperabilitas secara teratur — Verifikasi bahwa implementasi Anda tetap kompatibel.

Memulai dengan MCP

AWS secara aktif mendukung Model Context Protocol (MCP) melalui kontribusi untuk pengembangan dan implementasi protokol. AWS Berkolaborasi dengan kerangka kerja agen open-source terkemuka, termasuk,, dan LangGraph CrewAILlamaIndex, untuk membentuk masa depan komunikasi antar-agen pada protokol.

Untuk mengimplementasikan MCP dalam arsitektur agen Anda, lakukan tindakan berikut:

1. Jelajahi implementasi MCP dalam kerangka kerja seperti SDK. Strands Agents
2. Tinjau dokumentasi teknis Protokol Konteks Model.
3. Baca Protokol Terbuka untuk Interoperabilitas Agen Bagian 1: Komunikasi Antar Agen di MCP (AWS Blog) untuk mempelajari tentang interoperabilitas agen.
4. Bergabunglah dengan komunitas MCP untuk memengaruhi evolusi protokol.

MCP menyediakan lapisan komunikasi yang memungkinkan agen berinteraksi dengan data dan layanan eksternal dan juga dapat digunakan untuk memungkinkan agen berinteraksi dengan agen lain. Implementasi transportasi HTTP Streamable protokol memberi pengembang serangkaian pola interaksi yang komprehensif tanpa harus menemukan kembali roda. Pola-pola ini mendukung request/response arus stateless dan manajemen sesi stateful dengan persisten. IDs

Dengan mengadopsi protokol terbuka seperti MCP, Anda memposisikan organisasi Anda untuk membangun sistem agen yang tetap fleksibel, interoperable, dan mudah beradaptasi seiring berkembangnya teknologi AI. Untuk informasi tentang implementasi agent-to-tool protokol, lihat Strategi integrasi alat nanti dalam panduan ini.

Memulai dengan A2A

Protokol Agent2Agent (A2A) memungkinkan kolaborasi terdesentralisasi antar agen melalui lapisan semantik bersama. Alih-alih merutekan semua pekerjaan melalui orkestrator pusat, A2A memungkinkan agen untuk menemukan satu sama lain, mengiklankan kemampuan mereka, menegosiasikan tugas, dan berbagi konteks menggunakan protokol berbasis JSON yang ringan. Setiap agen menerbitkan manifes kemampuan.

Contoh berikut menunjukkan manifes kemampuan A2A yang disederhanakan yang mengiklankan tindakan yang didukung agen, input yang diperlukan, dan metadata operasional untuk memungkinkan penemuan dan negosiasi tugas:

```
{
  "can": ["summarize.text", "extract.keywords"],
  "needs": ["document.input"],
  "meta": { "version": "1.0.3", "latencyMs": 120 }
}
```

Model ini memungkinkan pencocokan kemampuan dinamis, delegasi tugas tengah, dan kolaborasi lintas organisasi. Agen dapat mengatur diri sendiri di sekitar tugas, membentuk kelompok kerja sementara, dan beradaptasi ketika kemampuan baru masuk atau keluar dari sistem.

A2A mendukung interaksi mulai dari permintaan stateless sederhana hingga sesi negosiasi multi-langkah, termasuk:

- peer-to-peerPesan langsung untuk kolaborasi latensi rendah
- Negosiasi tugas semantik, di mana agen memilih rekan yang paling cocok
- Penemuan berbasis kemampuan, memungkinkan pembagian kerja yang muncul
- Penahan sesi untuk interaksi multi-langkah stateful

Dengan mengadopsi protokol agen-asli terbuka seperti A2A, organisasi menciptakan sistem AI yang modular, interoperable, dan mampu berkolaborasi lintas batas. A2A memastikan bahwa ekosistem agen tetap fleksibel dan dapat berkembang saat agen, tim, atau sistem eksternal baru diperkenalkan, tanpa memerlukan lapisan orkestrasi yang kaku atau kopling sebelumnya.

Untuk mengimplementasikan protokol A2A dalam arsitektur agen Anda, lakukan tindakan berikut:

1. Tinjau Spesifikasi Protokol A2A — Baca versi terbaru Spesifikasi Protokol Agent2Agent (A2A) untuk mempelajari bagaimana kemampuan memanifestasikan, alur negosiasi, dan jabat tangan agen beroperasi.
2. Jelajahi runtime yang kompatibel dengan A2A — Evaluasi kerangka kerja seperti Strands Agents SDK atau lapisan runtime khusus yang mendukung manifes dan negosiasi kemampuan gaya A2A. peer-to-peer
3. Terapkan manifes kemampuan untuk agen Anda — Tentukan setiap agen`can`,`needs`, dan `meta` bidang untuk memungkinkan penemuan, perjodohan, dan kolaborasi tingkat niat.
4. Bereksperimenlah dengan pola negosiasi A2A — Gunakan loop permintaan—tawaran-terima, kueri kemampuan terstruktur, atau penemuan berbasis gosip untuk memahami bagaimana agen bernalar tentang siapa yang harus menangani tugas.
5. Uji A2A dalam lingkungan infrastruktur campuran — Gabungkan negosiasi sejawat A2A dengan perutean acara yang asli melalui AWS Amazon untuk mengevaluasi pola koordinasi hibrida. EventBridge
6. Bergabunglah dengan komunitas A2A — Terlibat dengan kelompok kerja terbuka untuk tetap up to date dengan ekstensi, rekomendasi keamanan, dan peningkatan interoperabilitas lintas vendor, dan berkontribusi pada pengembangan protokol.

Alat

Agen AI memberikan nilai dengan berinteraksi dengan alat eksternal APIs, dan sumber data untuk melakukan tugas yang bermanfaat. Strategi integrasi alat yang tepat secara langsung memengaruhi kemampuan agen Anda, postur keamanan, dan fleksibilitas jangka panjang.

Bagian ini membantu Anda menavigasi lanskap integrasi alat dengan fokus pada standar terbuka yang memaksimalkan kebebasan dan fleksibilitas Anda. Bagian ini menyoroti Model Context Protocol (MCP) untuk integrasi alat dan meninjau alat khusus kerangka kerja dan meta-tool khusus yang meningkatkan alur kerja agen.

Di bagian ini:

- Kategori alat
- Alat berbasis protokol
- Alat asli kerangka kerja
- Alat meta
- Strategi integrasi alat
- Praktik terbaik keamanan untuk integrasi alat

Kategori alat

Sistem agen bangunan melibatkan tiga kategori utama alat.

Alat berbasis protokol

Alat berbasis protokol menggunakan protokol standar untuk komunikasi: agent-to-tool

- Alat MCP — Buka alat standar yang bekerja di seluruh kerangka kerja dengan opsi eksekusi lokal dan jarak jauh.
- OpenAIpemanggilan fungsi — Alat berpemilik yang khusus untuk OpenAI model.
- Anthropicalat — Variasi pada OpenAI fungsi yang memanggil alat berpemilik yang khusus untuk model Anthropic Claude.

Alat asli kerangka kerja

Alat Framework-native dibangun langsung ke dalam kerangka kerja agen tertentu:

- Strands Agents alat - Ringan, quick-to-implement alat yang khusus untuk Strands Agents kerangka kerja.
- LangChainalat Python berbasis alat yang terintegrasi erat dengan LangChain ekosistem.
- LlamaIndexalat — Alat yang dioptimalkan untuk pengambilan dan pemrosesan data di dalamnyaLlamaIndex.

Alat meta

Meta-tools meningkatkan alur kerja agen tanpa langsung mengambil tindakan eksternal:

- Alat alur kerja - Kelola alur eksekusi agen, logika percabangan, dan manajemen status.
- Alat grafik agen - Mengkoordinasikan beberapa agen dalam alur kerja yang kompleks.
- Alat memori — Menyediakan penyimpanan persisten dan pengambilan informasi di seluruh sesi agen.
- Alat refleksi — Memungkinkan agen untuk menganalisis dan meningkatkan kinerja mereka sendiri.

Alat berbasis protokol

Saat mempertimbangkan alat berbasis protokol, Model Context Protocol (MCP) memberikan fondasi yang paling komprehensif dan fleksibel untuk integrasi alat. Sebagaimana dinyatakan dalam posting blog AWS Open Source tentang interoperabilitas agen, AWS telah merangkul MCP sebagai protokol strategis, secara aktif berkontribusi pada pengembangannya.

Tabel berikut menjelaskan opsi untuk penerapan alat MCP.

Model penyebaran	Deskripsi	Ideal untuk	Implementasi
Berbasis stdio lokal	Alat berjalan dalam proses yang sama dengan agen	Pengembangan, pengujian, dan alat sederhana	Cepat diimpleme ntasikan tanpa overhead jaringan

Acara terkirim server lokal (SSE) berbasis	Alat berjalan secara lokal tetapi berkomunikasi melalui HTTP	Alat lokal yang lebih kompleks dengan pemisahan kekhawatiran	Isolasi yang lebih baik tetapi latensi masih rendah
Remote HTTP Streamable	Alat berjalan di server jarak jauh	Lingkungan produksi dan alat bersama	Dapat diskalakan dan dikelola secara terpusat

MCP resmi SDKs tersedia untuk membangun alat MCP:

- PythonSDK — Implementasi komprehensif dengan dukungan protokol penuh
- TypeScriptSDK —JavaScript/TypeScriptimplementasi untuk aplikasi web
- JavaSDK - Implementasi Java untuk aplikasi perusahaan

Ini SDKs menyediakan blok bangunan untuk membuat alat yang kompatibel dengan MCP dalam bahasa pilihan Anda, dengan implementasi spesifikasi protokol yang konsisten.

Selain itu, AWS telah menerapkan MCP di Strands AgentsSDK. Strands AgentsSDK menyediakan cara mudah untuk membuat dan menggunakan alat yang kompatibel dengan MCP. Dokumentasi komprehensif tersedia di Strands Agents GitHub repositori. Untuk kasus penggunaan yang lebih sederhana atau ketika bekerja di luar Strands Agents kerangka kerja, MCP resmi SDKs menawarkan implementasi langsung protokol dalam berbagai bahasa.

Fitur keamanan alat MCP

Fitur keamanan alat MCP meliputi:

- OAuth 2.0/2.1 otentikasi — otentikasi standar industri
- Pelingkupan izin - Kontrol akses berbutir halus untuk alat
- Penemuan kemampuan alat — Penemuan dinamis alat yang tersedia
- Penanganan kesalahan terstruktur - Pola kesalahan yang konsisten

Memulai dengan alat MCP

Untuk menerapkan MCP untuk integrasi alat, lakukan tindakan berikut:

1. Jelajahi Strands AgentsSDK untuk implementasi MCP siap produksi.
2. Tinjau dokumentasi teknis MCP untuk memahami konsep inti.
3. Gunakan contoh praktis yang dijelaskan dalam posting Blog AWS Open Source ini.
4. Mulailah dengan alat lokal sederhana sebelum melanjutkan ke alat jarak jauh.
5. Bergabunglah dengan komunitas MCP untuk memengaruhi evolusi protokol.

Jelajahi AgentCore Gateway

Amazon Bedrock AgentCore Gateway menyediakan cara yang mudah dan aman bagi pengembang untuk membangun, menerapkan, menemukan, dan terhubung ke alat MCP, dan titik akhir target lainnya dalam skala besar. Dengan AgentCore Gateway, pengembang dapat mengonversi APIs, AWS Lambda fungsi, dan layanan yang ada menjadi alat yang kompatibel dengan MCP. Kemudian, hanya dengan beberapa baris kode, mereka dapat membuat alat ini tersedia untuk agen melalui titik akhir AgentCore Gateway. AgentCore Gateway mendukungOpenAPI,Smithy, dan Lambda sebagai tipe input, dan merupakan satu-satunya solusi yang menyediakan otentikasi ingress komprehensif dan otentikasi keluar dalam layanan yang dikelola sepenuhnya.

Alat asli kerangka kerja

Meskipun Model Context Protocol (MCP) memberikan fondasi yang paling fleksibel, alat kerangka kerja asli menawarkan keuntungan untuk kasus penggunaan tertentu.

Strands AgentsSDK menawarkan alat Python berbasis yang dicirikan oleh desainnya yang ringan yang membutuhkan overhead minimal untuk operasi sederhana. Mereka memungkinkan implementasi cepat dan memungkinkan pengembang untuk membuat alat hanya dengan beberapa baris kode. Selain itu, mereka terintegrasi erat untuk bekerja dengan mulus dalam Strands Agents kerangka kerja.

Contoh berikut menunjukkan cara membuat alat cuaca sederhana menggunakanStrands Agents. Pengembang dapat dengan cepat mengubah Python fungsi menjadi alat yang dapat diakses agen dengan overhead kode minimal dan secara otomatis menghasilkan dokumentasi yang sesuai dari docstring fungsi.

```
#Example of a simple Strands native tool

@tool
```

```
def weather(location: str) -> str:

"""Get the current weather for a location""" #

Implementation here

return f"The weather in {location} is sunny."
```

Untuk pembuatan prototipe cepat atau kasus penggunaan sederhana, alat kerangka kerja asli dapat mempercepat pengembangan. Namun, untuk sistem produksi, alat MCP memberikan interoperabilitas yang lebih baik dan fleksibilitas future daripada framework-native tools.

Tabel berikut memberikan ikhtisar alat khusus kerangka kerja lainnya.

Kerangka	Jenis alat	Keuntungan	Pertimbangan-pertimbangan
AutoGen	Definisi fungsi	Dukungan multi-agen yang kuat	Microsoftekosistem
LangChain	Pythonkelas	Ekosistem besar alat pra-bangun	Kerangka kerja terkunci
LlamaIndex	Fungsi Python	Dioptimalkan untuk operasi data	Terbatas untuk LlamaIndex

Alat meta

Meta-tools tidak berinteraksi langsung dengan sistem eksternal. Sebaliknya, mereka meningkatkan kemampuan agen dengan menerapkan pola agen. Bagian ini membahas alur kerja, grafik agen, dan meta-tools memori.

Meta-tools alur kerja

Alur kerja meta-tools mengelola alur eksekusi agen:

- Manajemen negara - Pertahankan konteks di berbagai interaksi agen
- Logika percabangan - Aktifkan jalur eksekusi bersyarat
- Mekanisme coba lagi — Tangani kegagalan dengan strategi coba ulang yang canggih

Contoh kerangka kerja dengan meta-tools alur kerja termasuk LangGraphdan kemampuan alur kerja. Strands Agents

Meta-tools grafik agen

Meta-tools grafik agen mengoordinasikan beberapa agen yang bekerja bersama:

- Delegasi tugas - Menetapkan subtugas ke agen khusus
- Agregasi hasil - Menggabungkan output dari beberapa agen
- Resolusi konflik — Menyelesaikan ketidaksepakatan antar agen

Kerangka kerja menyukai AutoGendan CrewAImengkhususkan diri dalam koordinasi grafik agen.

Alat meta memori

Meta-tools memori menyediakan penyimpanan dan pengambilan persisten:

- Riwayat percakapan - Pertahankan konteks di seluruh sesi
- Basis pengetahuan — Menyimpan dan mengambil informasi khusus domain
- Toko vektor - Aktifkan kemampuan pencarian semantik

Sistem sumber daya MCP menyediakan cara standar untuk mengimplementasikan meta-alat memori yang bekerja di berbagai kerangka kerja agen.

Strategi integrasi alat

Pilihan strategi integrasi alat Anda secara langsung memengaruhi apa yang dapat dicapai agen Anda dan seberapa mudah sistem Anda dapat berkembang. Prioritaskan protokol terbuka seperti Model Context Protocol (MCP) sementara secara strategis menggunakan framework-native tools dan meta-tools. Dengan begitu, Anda dapat membangun ekosistem alat yang tetap fleksibel dan kuat seiring kemajuan teknologi AI.

Pendekatan strategis berikut untuk integrasi alat memaksimalkan fleksibilitas sambil memenuhi kebutuhan mendesak organisasi Anda:

1. Mengadopsi MCP sebagai fondasi Anda - MCP menyediakan cara standar untuk menghubungkan agen ke alat dengan fitur keamanan yang kuat. Mulailah dengan MCP sebagai protokol alat utama Anda untuk:

- Alat strategis yang akan digunakan di beberapa implementasi agen.
- Alat sensitif keamanan yang memerlukan otentikasi dan otorisasi yang kuat.
- Alat yang membutuhkan eksekusi jarak jauh di lingkungan produksi.

2. Gunakan alat framework-native bila perlu — Pertimbangkan alat framework-native untuk:
   - Prototyping cepat selama pengembangan awal.
   - Alat non-kritis sederhana dengan persyaratan keamanan minimal.
   - Fungsionalitas khusus kerangka kerja yang memanfaatkan kemampuan unik.
3. Menerapkan meta-tools untuk alur kerja yang kompleks — Tambahkan meta-tools untuk menyempurnakan arsitektur agen Anda:
   - Mulai sederhana dengan pola alur kerja dasar.
   - Tambahkan kompleksitas saat kasus penggunaan Anda matang.
   - Standarisasi antarmuka antara agen dan meta-tools.
4. Rencanakan evolusi — Bangun dengan mempertimbangkan fleksibilitas masa depan:
   - Antarmuka alat dokumen secara independen dari implementasi.
   - Buat lapisan abstraksi antara agen dan alat.
   - Menetapkan jalur migrasi dari protokol eksklusif ke protokol terbuka.

Praktik terbaik keamanan untuk integrasi alat

Integrasi alat secara langsung memengaruhi postur keamanan Anda. Bagian ini menguraikan praktik terbaik untuk dipertimbangkan untuk organisasi Anda.

Autentikasi dan otorisasi

Manfaatkan kontrol akses yang kuat berikut:

- Gunakan OAuth 2.0/2.1 — Menerapkan otentikasi standar industri untuk alat jarak jauh.
- Terapkan hak istimewa paling sedikit — Berikan alat hanya izin yang mereka butuhkan.
- Putar kredensyal — Perbarui kunci API dan token akses secara teratur.

Perlindungan data

Untuk membantu melindungi data, lakukan langkah-langkah berikut:

- Validasi input dan output - Menerapkan validasi skema untuk semua interaksi alat.
- Enkripsi data sensitif — Gunakan TLS untuk semua komunikasi alat jarak jauh.
- Menerapkan minimisasi data — Hanya berikan informasi yang diperlukan ke alat.

Pemantauan dan audit

Pertahankan visibilitas dan kontrol dengan menggunakan mekanisme ini:

- Catat semua pemanggilan alat — Pertahankan jejak audit yang komprehensif.
- Memantau anomali — Mendeteksi pola penggunaan alat yang tidak biasa.
- Menerapkan pembatasan tarif - Mencegah penyalahgunaan melalui panggilan alat yang berlebihan.

Model keamanan Model Context Protocol (MCP) mengatasi masalah ini secara komprehensif. Untuk informasi selengkapnya, lihat Pertimbangan keamanan dalam dokumentasi MCP.

Kesimpulan

Lanskap AI agen terus berkembang pesat, menawarkan kepada organisasi cara baru yang kuat untuk membangun sistem yang cerdas dan otonom. Panduan ini telah mengeksplorasi tiga komponen penting untuk implementasi yang sukses: kerangka kerja yang menyediakan fondasi, platform yang menyediakan lingkungan, protokol yang memungkinkan komunikasi, dan alat yang memperluas kemampuan.

Saat kerangka kerja matang, Anda dapat mengharapkan peningkatan interoperabilitas, standardisasi di sekitar protokol seperti Model Context Protocol (MCP), dan kemampuan orkestrasi yang lebih canggih untuk agen otonom. Organizations yang membangun keahlian dengan kerangka kerja ini hari ini akan memiliki posisi yang baik untuk membangun agen yang semakin otonom dan cerdas yang memberikan nilai bisnis yang signifikan.

Platform menyediakan lingkungan eksekusi, tata kelola, dan siklus hidup tempat sistem agen beroperasi. Mereka menangani masalah seperti identitas, batas keamanan, observabilitas, manajemen memori, landasan sesi, dan interaksi yang aman dengan alat dan data. Di AWS lingkungan, platform seperti runtime agen terkelola dan layanan orkestrasi memungkinkan organisasi untuk menyebarkan, memantau, mengembangkan, dan mengatur agen otonom dan sistem agen dalam skala besar. Platform menjembatani kerangka kerja dasar dengan persyaratan operasional dunia nyata.

Pilihan protokol agen merupakan keputusan strategis yang menyeimbangkan kebutuhan pembangunan segera dengan fleksibilitas jangka panjang dan interoperabilitas. Dengan memprioritaskan protokol terbuka dan menciptakan lapisan abstraksi yang sesuai, organisasi dapat membangun sistem agen yang tetap dapat beradaptasi dengan teknologi yang berkembang sambil memenuhi persyaratan bisnis saat ini.

Bagi sebagian besar organisasi, MCP merupakan fondasi yang kuat karena standarnya yang terbuka, ekosistem yang berkembang, dukungan untuk pola agent-to-agent komunikasi, dan kemampuan integrasi alat. AWS telah merangkul MCP dan Agent2Agent (A2A) sebagai protokol strategis, secara aktif berkontribusi pada pengembangan mereka dan menerapkannya di seluruh layanan seperti SDK. Strands Agents Dengan menggunakan MCP atau A2A bersama dengan alat dan meta-tool framework-native yang sesuai, Anda dapat membangun sistem agen yang memberikan nilai langsung sambil tetap dapat beradaptasi dengan inovasi masa depan.

Sumber daya

Gunakan sumber daya berikut AWS dan lainnya yang terkait dengan pengembangan agen otonom.

AWS Blog

- Amazon Bedrock AgentCore Memory: Membangun agen sadar konteks
- Praktik terbaik untuk membangun aplikasi AI generatif yang kuat dengan Amazon Bedrock Agents - Bagian 1
- Praktik terbaik untuk membangun aplikasi AI generatif yang kuat dengan Amazon Bedrock Agents - Bagian 2
- Bangun jaringan pipa RAG yang kuat dengan dan LlamaIndex Amazon Bedrock
- Bangun agen AI tepercaya dengan Amazon Bedrock Observability AgentCore
- Evaluasi tanggapan RAG dengan Amazon Bedrock, LlamaIndex dan RAGAS
- Memperkenalkan Penerjemah AgentCore Kode Batuan Dasar Amazon
- Memperkenalkan Amazon Bedrock AgentCore Gateway: Mengubah pengembangan alat agen AI perusahaan
- Memperkenalkan Amazon Bedrock AgentCore Identity: Mengamankan AI agen dalam skala besar
- MemperkenalkanStrands Agents, SDK Agen AI Sumber Terbuka
- Protokol Terbuka untuk Interoperabilitas Agen Bagian 1: Komunikasi Antar Agen di MCP
- Meluncurkan dan menskalakan agen dan alat Anda dengan aman di Amazon Bedrock AgentCore Runtime
- AWS Transform untuk .NET, layanan AI agen pertama untuk memodernisasi aplikasi.NET dalam skala besar
- AWS Roundup Mingguan: Strands Agents

AWS Bimbingan Preskriptif

- Mengoperasionalkan AI agen pada AWS
- Yayasan AI agen pada AWS
- Pola dan alur kerja Agentic AI di AWS
- Membangun arsitektur tanpa server untuk AI agen AWS

- Membangun arsitektur multi-tenant untuk AI agen di AWS
- Keamanan untuk AI agen aktif AWS
- Pengambilan opsi dan arsitektur Augmented Generation di AWS

AWS sumber daya

- Dokumentasi Amazon Bedrock
- Dokumentasi Amazon Bedrock AgentCore
- Perangkat AgentCore Pemula Amazon Bedrock (repositori) GitHub
- Dokumentasi Amazon Nova
- AWS Server MCP (GitHubrepositori)

Sumber daya lainnya

- AutoGendokumentasi (Microsoft)
- Membangun agen yang efektif (Anthropic)
- CrewAI GitHubrepositori
- Dokumentasi LangChain
- LangGraphplatform
- Dokumentasi LlamaIndex
- Dokumentasi Protokol Konteks Model
- Dokumentasi Strands Agents
- Strands AgentsIkhtisar Alat
- Strands AgentsPanduan Memulai Cepat

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
Bagian baru	Bagian Platform Ditambahkan	Januari 16, 2026
Publikasi awal	—	Juli 14, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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