Berinvestasi dalam rekayasa kekacauan sebagai kebutuhan strategis

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Berinvestasi dalam rekayasa kekacauan sebagai kebutuhan strategis

Table of Contents

Berinvestasi dalam rekayasa kekacauan sebagai kebutuhan strategis

Adrian Hornsby, Amazon Web Services

Januari 2025 (sejarah dokumen)

Praktik rekayasa kekacauan menggunakan gangguan terkontrol untuk mengidentifikasi masalah sistem dan peluang untuk mencegah pemadaman dan insiden lainnya. Rekayasa kekacauan telah menjadi penting untuk meningkatkan sistem yang tangguh, tetapi adopsi yang meluas menghadapi rintangan seputar kesalahpahaman, ketahanan budaya, sumber daya, dan cara mengukur nilai bisnis. Menetapkan tujuan awal membantu memulai upaya rekayasa kekacauan, sementara mengukur laba atas investasi (ROI) membenarkan investasi lanjutan — terutama di tengah tekanan ekonomi.

Dokumen strategi ini menguraikan pendekatan holistik untuk menangkap perbaikan operasional kuantitatif dan manfaat organisasi kualitatif. Tujuan utamanya adalah untuk memperlakukan rekayasa kekacauan sebagai kebutuhan strategis yang mirip dengan keamanan siber dan bukan sebagai latihan pembenaran biaya yang berkelanjutan.

Biaya downtime dan munculnya chaos engineering

Information Technology Intelligence Consulting (ITIC) memperkirakan bahwa 90 persen perusahaan menghadapi biaya melebihi $300.000 per jam downtime, dengan 41 persen melebihi $1-5 juta per jam. Selain pendapatan yang hilang segera, downtime dapat menyebabkan masalah jangka panjang, termasuk kegagalan kepatuhan, penurunan harga saham, biaya mitigasi yang signifikan, dan bahkan kerusakan merek.

Sementara downtime umumnya dikaitkan dengan sistem online yang menghasilkan pendapatan, dampak negatifnya jauh melampaui itu. Semua bisnis dan organisasi besar, terlepas dari model pendapatan utama mereka, sangat bergantung pada ketersediaan sistem internal mereka, seperti SDM dan penggajian.

Downtime yang mempengaruhi layanan internal inti ini dapat menghambat kemampuan perusahaan untuk berfungsi, yang menyebabkan gangguan operasional yang substansif dan dampak keuangan. Masalah yang dihasilkan dapat mencakup yang berikut:

- Keterlambatan dalam membayar karyawan dan vendor
- Ketidakmampuan untuk memproses pesanan atau transaksi pelanggan
- Pelanggaran data sensitif yang diizinkan oleh sistem keamanan yang dikompromikan
- Kehilangan produktivitas dan peluang pendapatan
- Hukuman peraturan untuk ketidakpatuhan
- Kerusakan reputasi merek

Rekayasa kekacauan sengaja memperkenalkan gangguan yang dikendalikan. Menggunakan rekayasa kekacauan untuk memahami atau memverifikasi respons sistem terhadap gangguan telah menjadi praktik penting untuk meningkatkan ketahanan sistem. Chaos engineering memungkinkan organisasi Anda untuk secara proaktif mengungkap masalah, memvalidasi mekanisme ketahanan, dan pada akhirnya mengurangi risiko downtime yang tidak direncanakan dan biaya terkait. Manfaat rekayasa kekacauan meliputi:

- Mengekspos utang teknis
- Melatih otot operasional
- Membangun kepercayaan dalam sistem
- Mengidentifikasi titik kegagalan

- Meningkatkan pemantauan dan observabilitas
- Mendukung pembelajaran berbasis eksperimen
- Memberikan ketahanan yang lebih baik untuk mengurangi waktu henti

Ketika sistem menjadi lebih kompleks dan harapan pelanggan meningkat, rekayasa kekacauan semakin penting. Gartner merekomendasikan chaos engineering sebagai praktik penting bagi organisasi untuk mengurangi downtime yang tidak direncanakan dan meningkatkan ketahanan.

Tantangan adopsi rekayasa kekacauan

Meskipun rekayasa kekacauan merupakan praktik yang semakin penting untuk meningkatkan ketahanan sistem, pengadopsiannya dapat menghadapi hambatan berikut:

- Kesalahpahaman tentang risiko — Kesalahpahaman umum adalah bahwa rekayasa kekacauan hanya dilakukan di lingkungan produksi, yang mengarah pada kekhawatiran tentang risiko yang berlebihan. Persepsi ini berasal dari kurangnya pemahaman tentang sifat sistematis dan terkontrol dari praktik rekayasa kekacauan. Sebagaimana dicatat dalam AWS Well-Architected Framework, lakukan simulasi kesalahan terlebih dahulu di lingkungan non-produksi.
- Jangka panjang untuk nilai bisnis - Manfaat rekayasa kekacauan bertambah secara bertahap, sehingga sulit untuk mengukur nilai bisnis dan membenarkan investasi awal. ROI yang lebih lambat membuat sulit bagi organisasi untuk memprioritaskan dan tetap dengan rekayasa kekacauan.
- Kesenjangan keterampilan dan keahlian — Rekayasa kekacauan membutuhkan seperangkat keterampilan dan keahlian unik yang mungkin tidak tersedia dalam organisasi Anda. Membangun atau memperoleh keahlian ini dapat menjadi penghalang yang signifikan, terutama bagi organisasi yang baru dalam praktik dan mereka yang memiliki sumber daya terbatas.

Sisa dari dokumen strategi ini akan fokus sebagian besar pada tantangan kedua, yaitu untuk menunjukkan nilai bisnis dari rekayasa kekacauan.

Efek akumulasi dari rekayasa kekacauan

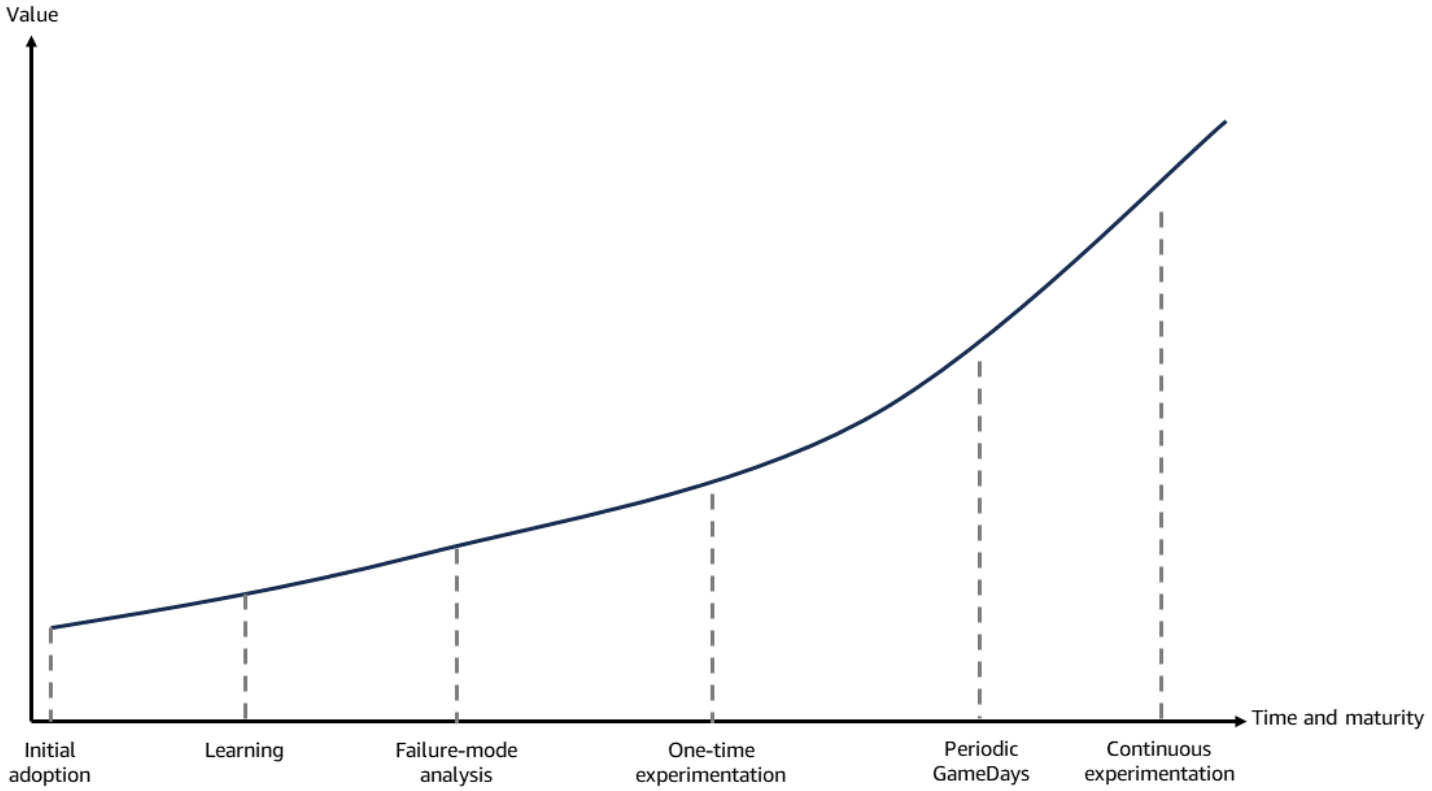
Tidak seperti proyek teknologi tradisional dengan tanggal mulai dan berakhir yang terdefinisi dengan baik, rekayasa kekacauan adalah praktik pembelajaran berkelanjutan dan peningkatan berkelanjutan terhadap ketahanan sistem. Manfaat senyawa rekayasa kekacauan dari waktu ke waktu.

Ketika sistem berkembang dan tumbuh lebih kompleks, mode kegagalan baru muncul. Lebih banyak eksperimen chaos diperlukan untuk mengidentifikasi potensi masalah. Memperbaiki masalah dapat memakan waktu berbulan-bulan, terutama di perusahaan besar dengan sistem dan proses yang rumit, atau ketika kesalahan dimiliki oleh penyedia layanan eksternal.

Pergeseran budaya ke arah merangkul kegagalan sebagai kesempatan untuk belajar dan perbaikan tumbuh selama bertahun-tahun dan menjadi mendarah daging dalam organisasi. Investasi dalam mengotomatiskan eksperimen rekayasa kekacauan dan mengembangkan perkakas pendukung terus merampingkan dan meningkatkan praktik rekayasa kekacauan. Membangun pengetahuan kelembagaan dan pemahaman tentang ketahanan sistem ini adalah proses bertahap yang terakumulasi dari waktu ke waktu. Pengetahuan, proses, dan alat yang dikembangkan melalui rekayasa kekacauan meningkat nilainya saat praktik matang di samping sistem yang terus berkembang.

Diagram berikut menunjukkan bagaimana nilai meningkat dari waktu ke waktu saat adopsi kekacauan berlangsung melalui tahapan berikut:

- Adopsi awal
- Belajar
- Analisis mode kegagalan
- Eksperimen satu kali
- Berkala GameDays
- Eksperimen berkelanjutan



Seperti yang ditunjukkan dalam diagram, manfaat rekayasa kekacauan sering dimulai sebelum kesalahan apa pun disuntikkan ke dalam sistem. Proses perencanaan dan perancangan eksperimen chaos itu sendiri memberikan nilai langsung. Mengidentifikasi skenario kegagalan potensial, titik kegagalan tunggal, dan area ketidakpastian dalam sistem mengarah pada perbaikan.

Misalnya, menuliskan skenario kegagalan dan mendiskusikan potensi efek cascading, sebuah proses yang disebut mode kegagalan dan analisis efek (FMEA), membantu mengungkap kelemahan atau kesenjangan yang jelas yang mungkin telah diabaikan. Organisasi Anda dapat secara proaktif mengatasi masalah tersebut, bahkan sebelum membuat sistem mengalami gangguan yang disengaja. Untuk informasi lebih lanjut, lihat kerangka analisis Ketahanan.

Selain itu, peningkatan fokus pada observabilitas dan pemantauan sistem yang sering menyertai inisiatif rekayasa kekacauan mulai memberikan manfaat segera. Meningkatkan visibilitas ke dalam perilaku sistem dan mode kegagalan membantu tim lebih memahami kondisi operasi normal sistem. Visibilitas yang lebih besar juga membantu tim memahami bagaimana kondisi operasi menurun, beradaptasi, dan gagal ketika didorong ke batasnya.

Baik percobaan satu kali maupun GameDay mode periodik lebih merupakan pendekatan manual dibandingkan dengan mode eksperimen berkelanjutan. Mereka membutuhkan proses yang lebih

langsung dan eksplorasi, di mana para insinyur secara aktif membentuk dan menyempurnakan hipotesis melalui pengamatan dan eksperimen mereka.

Mode eksperimen berkelanjutan, di sisi lain, lebih otomatis. Mode ini berfokus pada menjalankan hipotesis yang disetujui dan divalidasi secara terkontrol dan berulang. Ini menggunakan otomatisasi dan integrasi dalam proses pengembangan melalui pipeline chaos khusus untuk membantu memastikan eksperimen yang konsisten dan berulang.

Inisiatif rekayasa kekacauan akar rumput

Perjalanan rekayasa kekacauan sering dimulai pada tingkat akar rumput, di mana tim teknik mengidentifikasi kebutuhan dan mulai bereksperimen dengan rekayasa kekacauan secara mandiri.

Dalam pendekatan akar rumput ini, tim bereksperimen, belajar, dan menyempurnakan praktik rekayasa kekacauan mereka. Nilai rekayasa chaos dapat ditunjukkan melalui hasil nyata berikut:

- Mengurangi insiden
- Observabilitas yang lebih baik
- Waktu pemulihan lebih cepat
- Ketahanan sistem yang ditingkatkan dan berkelanjutan

Inisiatif rekayasa kekacauan akar rumput biasanya muncul di bawah kondisi organisasi tertentu. Mereka membutuhkan lingkungan dengan tingkat otonomi teknik yang tinggi, di mana tim memiliki kebebasan untuk bereksperimen dan berinovasi tanpa hambatan birokrasi yang berlebihan. Keahlian lokal dalam rekayasa ketahanan atau sistem terdistribusi sangat penting, karena memberikan dasar teknis untuk memahami dan menerapkan eksperimen kekacauan. Yang paling penting, inisiatif ini sering bergantung pada juara chaos — individu yang bersemangat yang memahami nilai rekayasa kekacauan. Juara Chaos bersedia mengadvokasi adopsi rekayasa kekacauan, mendidik rekan-rekan mereka, dan mendorong eksperimen awal. Tanpa kebebasan organisasi, keahlian teknis, dan juara yang termotivasi, upaya rekayasa kekacauan akar rumput jarang berakar, terlepas dari potensi manfaatnya.

Peran tujuan dalam adopsi rekayasa kekacauan

Adalah umum untuk tujuan awal muncul secara organik dari upaya rekayasa kekacauan akar rumput dalam suatu organisasi. Didorong oleh kebutuhan untuk mengatasi masalah berulang mereka sendiri, tim atau kelompok ini sering mengeksplorasi praktik rekayasa kekacauan tanpa persetujuan eksplisit atau prioritas dari tingkat yang lebih tinggi.

Tim dapat menggunakan hasil ini untuk membangun kasus yang menarik untuk adopsi organisasi yang lebih luas, yang secara efektif menjadi tempat pembuktian bagi tim lain.

Setelah manfaat dari upaya akar rumput menjadi terlalu signifikan untuk diabaikan, tim-tim ini dapat meningkatkan upaya dan pengetahuan mereka menjadi kepemimpinan dan menetapkan tujuan. Peningkatan visibilitas ini dapat memfasilitasi adopsi tujuan ketahanan di seluruh organisasi dan mengarah pada dukungan dan sumber daya yang diperlukan untuk implementasi rekayasa kekacauan.

Tujuan, terutama yang didorong oleh kepemimpinan dan ditetapkan sebagai respons terhadap pemadaman yang signifikan, memainkan peran penting dalam mengkatalisasi adopsi praktik rekayasa kekacauan. Jenis tujuan yang umum meliputi yang berikut:

- Tujuan ketersediaan untuk mengidentifikasi dan mengurangi titik kegagalan tunggal (SPOF)
- Tujuan pemulihan layanan untuk meningkatkan kemampuan untuk pulih dari gangguan atau kegagalan
- Tujuan pengalaman pengguna untuk memenuhi tujuan tingkat layanan tertentu () SLOs
- Sasaran berbasis metrik untuk melacak kemajuan dalam mengurangi risiko ketersediaan yang diketahui dan menerapkan langkah-langkah ketahanan yang direkomendasikan
- Tujuan regulasi dan kepatuhan untuk menunjukkan ketahanan operasional

Untuk informasi lebih lanjut tentang beberapa jenis tujuan ini dan bagaimana Amazon dan organisasi lain telah menggunakan tujuan selama adopsi rekayasa kekacauan, lihat Lampiran A.

Tujuan-tujuan ini berfungsi sebagai pembenaran yang meyakinkan dan memberikan pendekatan yang ditargetkan dan dapat ditindaklanjuti untuk mendorong adopsi rekayasa kekacauan. Pada awalnya, tujuan berfungsi sebagai proxy untuk metrik ROI tradisional. Sasaran menawarkan alasan yang meyakinkan ketika perhitungan ROI ketahanan yang dapat diukur mungkin sulit untuk diperoleh. Tanpa tujuan seperti itu di awal adopsi, praktik rekayasa kekacauan berisiko gagal untuk menunjukkan efektivitasnya dan mendapatkan dukungan organisasi yang lebih luas.

Pergeseran dari tujuan ke pengukuran ROI

Ketika praktik matang dan tujuan awal tercapai, fokus akhirnya bergeser dari menetapkan tujuan ke arah mengukur manfaat finansial nyata dari rekayasa chaos — laba atas investasi (ROI). Pergeseran ini berasal dari dua alasan utama:

- Pertimbangan ekonomi
- Melestarikan pengalaman dan kepercayaan pelanggan

Pertimbangan ekonomi

Pada saat pertumbuhan ekonomi dan keuangan yang sehat, perusahaan sering tidak memerlukan pembenaran ekstensif untuk menetapkan tujuan spesifik untuk strategi rekayasa kekacauan. Namun, perubahan dalam lanskap keuangan telah menyebabkan banyak organisasi mengevaluasi kembali investasi mereka, dan implementasi rekayasa kekacauan perlu memberikan ROI terkuantifikasi.

Perusahaan-perusahaan ini sekarang ditugaskan untuk menetapkan metrik ROI tradisional yang jelas untuk menunjukkan nilai dan dampak praktik rekayasa kekacauan. Tantangan ini semakin diperumit oleh paradoks pencegahan. Paradoks pencegahan terjadi ketika pencegahan insiden yang berhasil membuat lebih sulit untuk membenarkan investasi, karena pemangku kepentingan cenderung meremehkan bencana yang dihindari. Bahkan organisasi dengan budaya keunggulan operasional yang mendarah daging menghadapi tekanan untuk menggunakan metrik ROI untuk membenarkan adopsi berkelanjutan dari rekayasa kekacauan.

Melestarikan pengalaman dan kepercayaan pelanggan

Mempertahankan ketahanan yang didorong oleh tujuan dapat menjadi tantangan dalam jangka panjang. Setelah tujuan awal seperti mencapai target waktu pemulihan terpenuhi, membenarkan investasi rekayasa kekacauan berkelanjutan menjadi sulit sampai pemadaman besar berikutnya. Aliran dan pasang surut investasi menciptakan siklus gigi gergaji reaktif. Untuk setiap pemadaman baru, investasi dalam ketahanan melonjak dengan tujuan baru mengatasi akar masalahnya. Setelah tujuan baru terpenuhi, investasi turun hingga insiden berikutnya, memulai kembali loop reaktif.

Pemadaman yang mendorong pendekatan reaktif ini berdampak negatif pada pelanggan. Pertanyaan kuncinya: Berapa banyak pemadaman besar yang akan ditoleransi pelanggan sebelum mereka meninggalkan penyedia layanan demi pesaing yang lebih tangguh?

Mengukur ROI rekayasa kekacauan

Saat ini, sangat sedikit sumber daya yang diterbitkan menyediakan metodologi komprehensif atau data dunia nyata untuk mengukur laba atas investasi jangka panjang (ROI) untuk rekayasa kekacauan.

Dalam paper The Business Case for Chaos Engineering, Netflix menawarkan persamaan berharga untuk menghitung ROI untuk rekayasa kekacauan. Persamaan ini memberikan titik awal bagi organisasi yang memulai perjalanan rekayasa kekacauan mereka.

Persamaan tersebut mengharuskan Anda memperkirakan biaya berikut secara akurat:

- Pemadaman yang dapat dicegah dan tidak dapat dicegah
- Biaya implementasi program rekayasa kekacauan
- Biaya kerusakan yang disebabkan oleh kekacauan

Kerusakan akibat kekacauan mengacu pada dampak negatif atau gangguan yang disebabkan oleh sengaja menyuntikkan kesalahan atau kondisi turbulen ke dalam sistem sebagai bagian dari eksperimen rekayasa kekacauan. Persamaan ini membutuhkan estimasi biaya pemadaman yang dapat dicegah dan tidak dapat dicegah, biaya implementasi program rekayasa kekacauan, dan biaya kerusakan yang disebabkan oleh kekacauan.

Menentukan dengan pasti masalah mana yang bisa dicegah oleh program rekayasa kekacauan adalah tugas yang sulit. Ini membutuhkan analisis hipotetis yang melibatkan melihat akar penyebab masalah dan berspekulasi bagaimana eksperimen rekayasa kekacauan mungkin membantu mengidentifikasi mereka. Analisis ini menantang karena sistem modern sangat kompleks, dengan banyak saling ketergantungan dan interaksi antara berbagai komponen, layanan, dan perpustakaan pihak ketiga. Selain itu, kesalahan dalam sistem seringkali tidak deterministik, dan kondisi yang menyebabkan kesalahan bisa sulit untuk dipahami sepenuhnya di belakang.

Meskipun pendekatan yang disarankan oleh Netflix memiliki beberapa keterbatasan, ini berfungsi sebagai fondasi yang baik bagi organisasi yang mulai mengeksplorasi rekayasa kekacauan. Persamaan ini dapat memandu Anda dalam memperkirakan biaya dan potensi manfaat, yang membantu Anda membuat keputusan tentang penerapan program semacam itu. Namun, seiring kemajuan organisasi lebih jauh dalam perjalanan rekayasa kekacauan mereka, penting untuk memperluas penilaian ROI untuk menggabungkan perspektif yang lebih holistik.

Pendekatan holistik ini tidak hanya akan menangkap manfaat langsung dari pengurangan pemadaman dan biaya rekayasa tetapi juga menyoroti efek transformatif jangka panjang pada ketahanan organisasi secara keseluruhan. Ini menangkap manfaat peracikan dan efek organisasi yang lebih luas dari rekayasa kekacauan untuk memberikan representasi yang lebih akurat dari nilai sebenarnya dan dampak rekayasa kekacauan.

Pendekatan holistik untuk kuantifikasi ROI

Penilaian ROI holistik harus memperhitungkan tidak hanya untuk ukuran kuantitatif tetapi juga faktor kualitatif. Pendekatan holistik membutuhkan data dunia nyata dari organisasi yang mempraktikkan rekayasa kekacauan dalam skala besar dalam jangka waktu yang lebih lama. Anda dapat menggunakan data mulai dari proyek dan sasaran akar rumput melalui data ROI pendekatan persamaan apa pun yang Anda kumpulkan.

Pengukuran kuantitatif fokus pada kuantitas atau frekuensi. Pengukurannya objektif, dan dapat dianalisis secara statistik. Contohnya termasuk survei, eksperimen, dan analisis data. Langkah-langkah kuantitatif dapat mencakup hal-hal berikut:

- Metrik insiden
- Biaya
- Perbaikan
- Kepatuhan
- Tingkat adopsi
- Kepuasan pelanggan

Melacak langkah-langkah kuantitatif dapat menunjukkan manfaat operasional langsung dari rekayasa kekacauan.

Ukuran kualitatif bersifat deskriptif, dan fokus pada pemahaman pengalaman dan pendapat. Mereka sering subjektif, dan mereka tidak dapat dengan mudah diukur secara numerik. Untuk rekayasa kekacauan, langkah-langkah kualitatif menangkap dampak organisasi yang lebih luas. Langkah-langkah kualitatif dapat mencakup hal-hal berikut:

- Kepercayaan karyawan
- Pergeseran budaya
- Kolaborasi

- Efektivitas pelatihan
- Retensi bakat
- Reputasi merek
- Keunggulan kompetitif

Dengan mempertimbangkan dampak keuangan kuantitatif dan manfaat organisasi kualitatif, Anda dapat membuat keputusan yang lebih tepat tentang investasi rekayasa kekacauan yang berkelanjutan sambil menumbuhkan budaya ketahanan.

Untuk informasi lebih lanjut tentang langkah-langkah ini dan kerangka klasifikasi insiden terkait, lihat Lampiran B dan Lampiran C.

Transisi dari ROI ke chaos engineering sebagai kebutuhan strategis

Meskipun tergoda untuk memantau ROI, tantangan dalam mengukur nilai chaos engineering sering membuat organisasi memprioritaskan efisiensi jangka pendek langsung daripada investasi ketahanan strategis. Pendekatan ini mengabaikan rekayasa kekacauan sebagai pendorong utama ketahanan dan keunggulan kompetitif untuk menghindari pemadaman. Nilai sebenarnya dari chaos engineering adalah mencegah kegagalan masa depan. Chaos engineering mendukung kelangsungan bisnis jangka panjang.

Alih-alih berfokus pada ROI, perlakukan rekayasa kekacauan seperti keamanan siber. Seperti yang dijelaskan dalam artikel Forbes Keamanan Siber Sebagai Investasi Strategis: Bagaimana Optimalisasi ROI Dapat Menyebabkan Masa Depan yang Lebih Aman, keamanan siber tidak boleh dipandang sebagai pusat biaya atau pengeluaran wajib bagi organisasi karena pola pikir itu gagal mengenali nilai strategis yang dapat diberikan oleh langkah-langkah keamanan siber yang kuat dari waktu ke waktu. Sebaliknya, penulis berpendapat bahwa dengan mengubah perspektif untuk memperlakukan keamanan siber sebagai investasi jangka panjang yang mendorong keunggulan kompetitif, organisasi dapat membuka jalan baru untuk inovasi, efisiensi operasional, dan diferensiasi dalam pasar masing-masing. Dengan mengadopsi pendekatan ini, penulis menyimpulkan bahwa Chief Information Security Officers (CISOs) dapat lebih mengamankan pembelian dan pendanaan kepemimpinan. Mereka kemudian dapat memposisikan perusahaan mereka untuk melampaui pesaing dalam lanskap cyber yang semakin berisiko. Penciptaan nilai strategis jangka panjang dari keamanan siber ini sejajar dengan peningkatan berkelanjutan yang melekat dalam praktik rekayasa kekacauan.

Sementara keamanan melindungi kemampuan organisasi untuk mengoperasikan dan melindungi aset, chaos engineering membantu memastikan ketersediaan, keandalan, dan pemulihan sistem dan layanan inti. Untuk mewujudkan nilai jangka panjang dan keunggulan kompetitif, perlakukan rekayasa kekacauan sebagai kemampuan inti dan keharusan strategis, bukan sebagai inisiatif yang membutuhkan pembenaran konstan.

Diagram berikut menunjukkan evolusi rekayasa kekacauan dari akar rumput ke tujuan dan ROI, menjadi strategi.

Pada tingkat akar rumput, tim individu biasanya bereksperimen secara independen, didorong oleh kebutuhan lokal. Eksperimen ini diperjuangkan oleh para insinyur yang bersemangat yang menunjukkan nilai melalui insiden yang berkurang dan peningkatan observabilitas.

Ketika upaya ini terbukti berhasil, tim dapat meningkatkan pembelajaran mereka menjadi kepemimpinan. Dengan visibilitas ini, upaya beralih ke fase yang didorong oleh tujuan. Organisasi menetapkan tujuan formal untuk ketahanan dan pemulihan, didukung oleh sumber daya dan dukungan untuk implementasi yang lebih luas.

Akhirnya, rekayasa kekacauan matang melampaui membutuhkan pembenaran ROI yang konstan untuk diakui sebagai kebutuhan strategis, mirip dengan keamanan siber. Pada tahap ini, chaos engineering menjadi terintegrasi penuh ke dalam proses organisasi. Implementasi berfokus pada ketahanan jangka panjang daripada metrik jangka pendek. Chaos engineering diperlakukan sebagai kemampuan inti yang penting untuk mempertahankan keunggulan kompetitif dan kepercayaan pelanggan.

Mengintegrasikan rekayasa kekacauan ke dalam organisasi Anda

Untuk meningkatkan rekayasa kekacauan ke tingkat kepentingan yang sama dengan keamanan, pertimbangkan saran berikut:

- Menetapkan rekayasa kekacauan sebagai praktik yang tidak dapat dinegosiasikan - Sama seperti keamanan siber dianggap sebagai persyaratan mendasar bagi organisasi, lihat rekayasa kekacauan sebagai praktik wajib untuk memastikan ketahanan dan keandalan sistem. Integrasikan rekayasa kekacauan ke dalam proses, alat, dan budaya organisasi Anda, daripada menganggapnya sebagai aktivitas opsional atau diskresioner. Untuk informasi selengkapnya, lihat panduan kerangka kerja siklus hidup Ketahanan.
- Pembelian dan dukungan tingkat eksekutif yang aman — Seperti halnya inisiatif keamanan, upaya rekayasa kekacauan harus memiliki dukungan aktif dan dukungan aktif dari kepemimpinan eksekutif. Ini termasuk mengalokasikan sumber daya, anggaran, dan personel khusus untuk menerapkan dan mempertahankan praktik rekayasa kekacauan di seluruh organisasi.

- Menerapkan tata kelola dan pengawasan — Mirip dengan kerangka kerja tata kelola CISO dan keamanan, buat tim teknik kekacauan khusus atau Chief Resilience Officer. Tim atau peran ini bertanggung jawab untuk mengawasi dan mengoordinasikan upaya rekayasa kekacauan di berbagai tim dan unit bisnis.
- Integrasikan rekayasa kekacauan ke dalam siklus pengembangan dan operasi — Sama seperti praktik keamanan yang diintegrasikan ke dalam pengembangan perangkat lunak dan proses penyebaran, membuat rekayasa kekacauan menjadi bagian yang mulus dari pengembangan perangkat lunak dan siklus hidup pengiriman.
- Lakukan latihan dan simulasi rekayasa chaos reguler - Mirip dengan simulasi pelanggaran keamanan dan latihan respons insiden, lakukan eksperimen rekayasa kekacauan reguler untuk memvalidasi kemampuan respons insiden dan mengidentifikasi titik buta potensial secara proaktif.
- Gunakan rekayasa kekacauan untuk memelihara runbook - Seperti halnya melakukan tinjauan keamanan, gunakan eksperimen rekayasa kekacauan untuk memvalidasi efektivitas dan akurasi runbook untuk respons dan pemulihan insiden. Selain itu, eksperimen rekayasa chaos dapat berfungsi sebagai simulasi realistis bagi insinyur on-call untuk berlatih menjalankan prosedur runbook. Simulasi membantu para insinyur mempertahankan memori otot operasional dan kesiapan mereka untuk menangani insiden dunia nyata.
- Menumbuhkan budaya ketahanan - Seperti halnya pelatihan kesadaran keamanan, berinvestasi dalam pendidikan teknik kekacauan dan inisiatif berbagi pengetahuan untuk menumbuhkan budaya ketahanan. Sertakan program pelatihan, kolaborasi lintas fungsi, dan insentif untuk tim yang mengadopsi praktik rekayasa kekacauan.
- Ukur dan laporkan metrik ketahanan - Pantau metrik ketahanan secara teratur dan laporkan kepada pemangku kepentingan. Gunakan metrik kuantitatif dan kualitatif yang dibahas dalam dokumen ini sebagai titik awal.
- Perlakukan ketahanan sebagai keunggulan kompetitif - Tindakan keamanan siber dapat memberikan keunggulan kompetitif. Demikian pula, lihat kemampuan rekayasa kekacauan dan ketahanan Anda sebagai pembeda yang membantu Anda menawarkan layanan yang lebih andal dan dapat dipercaya kepada pelanggan Anda.

Mendapatkan buy-in eksekutif

Rekayasa kekacauan sering tidak memiliki pemilik yang jelas dalam tanggung jawab tradisional C-suite. CEO peduli dengan pertumbuhan, profitabilitas, dan kepemimpinan pasar. CFO berfokus pada kinerja keuangan, pengendalian biaya, dan manajemen risiko. CTO memprioritaskan strategi teknologi, peta jalan produk, dan keunggulan teknik. CISO mengawasi keamanan dan kepatuhan.

Dengan tidak ada eksekutif tunggal yang benar-benar memiliki ketahanan, seringkali sulit untuk mendapatkan dukungan dan dukungan. Namun kegagalan sistem berdampak pada pendapatan, kepuasan pelanggan, dan reputasi merek, yang menjadi perhatian CEO dan CFO. CTO dan CISO ditugaskan untuk menerapkan langkah-langkah ketahanan, tetapi mereka mungkin tidak memiliki mandat organisasi. Ambiguitas ini dapat menghalangi investasi strategis dan menyelaraskan organisasi menuju strategi ketahanan bersama.

Ambiguitas ini juga membuat sulit untuk mendapatkan dukungan eksekutif untuk inisiatif ketahanan seperti rekayasa kekacauan. Bagaimanapun, para pemimpin tingkat C menyulap banyak prioritas strategis: pertumbuhan, inovasi, pengalaman pelanggan, kepatuhan, dan banyak lagi.

Untuk secara efektif mengkomunikasikan nilai rekayasa kekacauan kepada eksekutif tingkat-C, pertimbangkan pendekatan berikut:

- Tentukan masalah utama dan penggerak keputusan eksekutif C-suite Anda.

  Misalnya, apakah eksekutif C-suite khawatir tentang churn pelanggan, kepatuhan terhadap peraturan, pengurangan biaya, atau tekanan kompetitif? Posisikan chaos engineering sebagai pengganda kekuatan yang sejalan dengan tantangan dan tujuan unik perusahaan.
- Identifikasi tujuan bersama dan hasil strategis.

  Bagaimana strategi chaos engineering Anda mendukung strategi pertumbuhan organisasi secara keseluruhan, pengalaman pelanggan, peluang pasar, dan efisiensi operasional? Prioritaskan inisiatif berdasarkan tujuan, dampak bisnis, ROI, dan risiko tidak melakukan inisiatif.
- Komunikasikan efektivitas strategi rekayasa kekacauan Anda dalam istilah yang dapat diukur dengan menggunakan indikator ketahanan utama.

  Mulailah dengan empat indikator ketahanan utama ini: ketersediaan, waktu untuk mendeteksi, waktu untuk merespons, dan waktu untuk pulih. Ikat ini langsung dengan hasil bisnis seperti pendapatan, penghematan biaya, dan reputasi merek.
- Jangan tersesat dalam detail teknis.

  Fokus pada sentimen keseluruhan dan dampak bisnis yang terukur. C-suite peduli dengan hasil yang mendorong pertumbuhan, meningkatkan kepercayaan pelanggan, dan mendorong inovasi.

Paradoks pencegahan

Ketika kesalahan berhasil dikurangi sebelum terwujud, menjadi tantangan untuk meyakinkan para pemangku kepentingan tentang nilai dan perlunya tindakan pencegahan yang diambil. Fenomena ini dikenal sebagai paradoks pencegahan. Paradoks pencegahan adalah hambatan terbesar untuk mengintegrasikan rekayasa kekacauan sebagai kebutuhan strategis, dan itu berasal dari bias yang melekat dalam kognisi manusia.

Bug Y2K berfungsi sebagai ilustrasi yang bagus dari paradoks ini. Bertahun-tahun persiapan dan miliaran dolar diinvestasikan untuk memperbarui komputer di seluruh dunia. Namun, transisi yang mulus ke tahun 2000 ditafsirkan oleh banyak orang sebagai bukti sifat berlebihan dari kekhawatiran Y2K. Keberhasilan upaya pencegahan yang dilakukan jarang diakui.

Paradoks pencegahan ini terus menantang organisasi yang berinvestasi dalam rekayasa kekacauan saat ini. Ketika potensi pemadaman berhasil dihindari melalui tindakan proaktif, tidak adanya bencana secara paradoks dapat menyulitkan untuk membenarkan sumber daya yang dihabiskan untuk pencegahan.

Akar penyebab fenomena ini terletak pada cara pikiran kita terhubung untuk memproses informasi. Proses kognitif manusia diarahkan untuk merespons dan mengingat peristiwa aktual dan hasil yang terlihat. Ketika bencana dicegah, tidak ada narasi dramatis untuk dipegang atau dibagikan. Aspek lain dari paradoks pencegahan adalah bias melihat ke belakang. Setelah nonevent, individu cenderung menyimpulkan bahwa tidak ada yang terjadi, jadi itu bukan masalah nyata. Kemungkinan bahwa tindakan pencegahan yang tepat mencegah masalah nyata tidak dikenali. Titik buta psikologis ini menciptakan tantangan abadi bagi organisasi. Semakin sukses Anda dalam pencegahan dan ketahanan, semakin banyak upaya Anda tampak tidak perlu dalam retrospeksi.

Untuk mengatasi paradoks pencegahan, organisasi Anda dapat mengambil langkah-langkah spesifik untuk membuat pekerjaan pencegahan yang tidak terlihat terlihat terlihat, terukur, dan dihargai. Langkah-langkah potensial meliputi:

- Dokumentasikan dan simulasikan apa yang bisa terjadi tanpa tindakan pencegahan.
- Bagikan cerita tentang peristiwa di mana tindakan pencegahan mencegah potensi bencana.
- Tunjuk ke organisasi sebaya yang tidak mempersiapkan dan akibatnya menderita konsekuensi.
- Menyajikan biaya pencegahan dalam konteks dampak potensial yang mereka cegah.
- Memecah upaya pencegahan menjadi tonggak dan pencapaian yang terlihat.
- Bangun memori institusional tentang mengapa tindakan pencegahan ada dan kepentingan historisnya.

- Secara teratur mendidik para pemangku kepentingan tentang nilai ketahanan dan praktik rekayasa kekacauan.

Kesimpulan

Rekayasa kekacauan adalah keharusan strategis bagi organisasi. Meskipun perjalanan adopsi Anda mungkin menghadapi tantangan seperti kesalahpahaman, resistensi budaya, dan kendala sumber daya, menetapkan tujuan yang jelas dan didorong oleh kepemimpinan dapat mengkatalisasi proses tersebut. Ketika praktik matang, ukur laba atas investasi melalui pendekatan holistik yang menangkap peningkatan operasional kuantitatif dan manfaat organisasi kualitatif. Pendekatan holistik sangat penting selama tekanan ekonomi.

Untuk mengubah kebutuhan strategis ini menjadi kenyataan, mulailah dengan menilai tingkat kematangan organisasi Anda saat ini. Apakah organisasi Anda pada tahap eksperimen akar rumput, fase yang didorong oleh tujuan, atau di suatu tempat di antaranya? Berdasarkan penilaian ini, buat peta jalan yang disesuaikan untuk mencapai hal-hal berikut:

- Menetapkan tata kelola rekayasa kekacauan (misalnya, menunjuk Chief Resilience Officer).
- Integrasikan praktik kekacauan ke dalam alur kerja pengembangan.
- Melaksanakan program pelatihan reguler.
- Kembangkan metrik ketahanan yang komprehensif.

Transformasi ini tidak akan terjadi dalam semalam. Namun, mengambil langkah-langkah konkret ini, sambil mengamankan dukungan eksekutif yang berkelanjutan, akan membantu meningkatkan rekayasa kekacauan ke tingkat strategis yang sama dengan keamanan siber. Mirip dengan cybersecurity, chaos engineering dapat menjadi bagian integral dari DNA dan proses operasional organisasi Anda.

Sumber daya

- ITIC 2021 Perangkat Keras Server Global, Hasil Survei Keandalan OS Server
- Kasus Bisnis untuk Rekayasa Kekacauan
- Keamanan Siber Sebagai Investasi Strategis: Bagaimana Optimalisasi ROI Dapat Mengarah Ke Masa Depan yang Lebih Aman
- Panduan Pemimpin I&O untuk Rekayasa Kekacauan
- Cara menggunakan skor AWS Resilience Hub
- Menerapkan eksperimen yang direkomendasikan menggunakan konsol AWS Resilience Hub

Lampiran A - Jenis tujuan untuk rekayasa kekacauan

Deskripsi jenis tujuan berikut mencakup contoh dunia nyata tentang bagaimana Amazon dan organisasi lain telah merancang tujuan untuk rekayasa kekacauan.

Tujuan arsitektur tangguh

Salah satu pendorong awal untuk mengadopsi chaos engineering adalah mengidentifikasi dan mengurangi titik kegagalan tunggal (SPOF) di seluruh sistem dan infrastruktur. Tujuan ditetapkan untuk memvalidasi ketahanan sistem dan arsitektur kritis, terutama untuk layanan atau aplikasi baru.

Tujuan arsitektur tangguh melibatkan menjalankan eksperimen chaos yang mensimulasikan kegagalan dalam dependensi layanan. Eksperimen mengkonfirmasi apakah batas waktu, percobaan ulang, perilaku caching, dan konfigurasi pemutus sirkuit berfungsi dengan benar. Eksperimen ini membantu mengungkap masalah untuk remediasi, mencegah insiden yang berdampak pada pelanggan. Sebagai contoh, lihat Membangun layanan tangguh di Prime Video dengan rekayasa kekacauan.

Tujuan pemulihan layanan

Tujuan pemulihan layanan berfokus pada peningkatan kemampuan untuk pulih dari gangguan operasional atau kegagalan infrastruktur. Misalnya, organisasi Anda mungkin bertujuan untuk mencapai tujuan waktu pemulihan tertentu (RTO) untuk layanan inti Anda jika terjadi pemadaman. Tim dapat merancang eksperimen chaos untuk memvalidasi dan mengoptimalkan strategi evakuasi, mekanisme failover, dan proses pemulihan otomatis. Pengoptimalan pada akhirnya mengurangi waktu yang diperlukan untuk restorasi layanan. Sebagai contoh, lihat AWS Lambda: Ketahanan under-the-hood.

Tujuan pengalaman pengguna

Mempertahankan pengalaman pengguna yang konsisten dan andal adalah yang terpenting, terutama selama periode lalu lintas tinggi atau peristiwa kritis. Dalam kasus seperti itu, tetapkan tujuan yang berpusat di sekitar memenuhi tujuan tingkat layanan tertentu (). SLOs Pendekatan yang berpusat pada pelanggan ini memastikan bahwa upaya ketahanan secara langsung selaras dengan memberikan pengalaman pengguna yang unggul, bahkan dalam menghadapi kegagalan atau kondisi

yang terdegradasi. Sebagai contoh, lihat Ketahanan Rekayasa: Pelajaran dari Perjalanan Rekayasa Kekacauan Amazon Search.

Tujuan berbasis metrik

Anda dapat menetapkan sasaran berdasarkan metrik kuantitatif, seperti skor ketahanan yang dihitung dengan memberikan poin ke layanan yang mengadopsi praktik terbaik ketahanan yang terbukti. Anda kemudian dapat menggunakan eksperimen kekacauan tertentu untuk menentukan skor ketahanan. Skor ini dapat berfungsi sebagai ukuran bagi tim untuk melacak kemajuan mereka dalam mengurangi risiko ketersediaan yang diketahui dan menerapkan langkah-langkah ketahanan yang direkomendasikan. Namun, sangat penting untuk menafsirkan skor tersebut dengan hati-hati dan menghindari terlalu menekankan satu metrik dengan mengorbankan tujuan ketahanan yang lebih luas. Sebagai contoh, lihat Memahami skor ketahanan.

Tujuan kepatuhan regulasi

Industri jasa keuangan telah muncul sebagai pelari terdepan dalam merangkul rekayasa kekacauan, terutama didorong oleh persyaratan peraturan yang ketat yang mengamanatkan kemampuan ketahanan yang kuat. Peraturan akan menuntut agar lembaga keuangan secara proaktif mengidentifikasi, menguji, dan memulihkan kerentanan dalam sistem dan proses kritis mereka. Peraturan ini meliputi:

- Makalah Antar Lembaga tentang Praktik Suara untuk Memperkuat Ketahanan Operasional yang dikeluarkan oleh lembaga federal AS
- Pedoman Bank Sentral Eropa tentang ketahanan operasional
- Proposal Komisi Eropa untuk Undang-Undang Ketahanan Operasional Digital (DORA)

Jika organisasi Anda adalah lembaga keuangan, patuhi peraturan ini dengan menetapkan tujuan eksplisit untuk menunjukkan ketahanan operasional melalui strategi pengujian dan validasi yang komprehensif. Sebagai contoh, lihat London Stock Exchange Group menggunakan chaos engineering AWS untuk meningkatkan ketahanan.

Lampiran B - Ukuran kuantitatif dan kualitatif

Bagian ini menguraikan metrik kuantitatif untuk melacak peningkatan operasional dan langkah-langkah kualitatif untuk menilai hasil organisasi yang lebih luas dari praktik rekayasa kekacauan.

Ukuran kuantitatif

Langkah-langkah kuantitatif berikut menyediakan kerangka kerja untuk melacak metrik utama yang dapat menunjukkan insiden langsung dan peningkatan operasional yang dicapai melalui praktik rekayasa kekacauan:

- Insiden:
  - Frekuensi insiden - Lacak jumlah insiden dalam kerangka klasifikasi insiden dan klasifikasikan berdasarkan kekritisan mereka (kritis, besar, kecil) selama periode waktu tertentu. Untuk informasi lebih lanjut tentang kerangka klasifikasi insiden, lihat Lampiran C.
  - Waktu henti dan degradasi - Ukur total durasi waktu henti atau degradasi layanan untuk setiap klasifikasi insiden.
  - Metrik respons insiden - Untuk memahami insiden, mengukur Waktu untuk Mendeteksi, Waktu untuk Identifikasi, Waktu untuk Mengurangi, Waktu untuk Memulihkan, Waktu untuk Eskalasi, dan metrik terkait lainnya untuk setiap klasifikasi insiden.
  - Insiden yang berdampak pada pelanggan - Melacak jumlah insiden yang berdampak pada pelanggan atau persentase insiden yang terkandung sebelum dampak pelanggan.
  - Perubahan buku runbook - Lacak jumlah pembaruan atau revisi runbook yang dihasilkan dari wawasan yang diperoleh melalui eksperimen chaos. Runbook memberikan instruksi terperinci untuk melakukan operasi atau prosedur tertentu untuk pulih dari jenis insiden tertentu.
- Biaya:
  - Biaya infrastruktur - Mengumpulkan data tentang biaya infrastruktur, termasuk sumber daya komputasi awan dan langkah-langkah redundansi yang diperlukan oleh tindakan yang diambil untuk meningkatkan ketahanan.
  - Dampak pelanggan — Mengukur dampak terhadap pengalaman pelanggan, churn rate, dan kerugian pendapatan yang terkait dengan kegagalan sistem atau downtime.
  - Produktivitas staf - Lacak waktu yang dihabiskan oleh tim teknik dan operasi pada respons insiden, pemadaman kebakaran, menulis postmortem, dan tugas reaktif lainnya yang terkait dengan kegagalan sistem.

- Perbaikan sistem berkelanjutan — Hitung jumlah perbaikan proses, perubahan arsitektur, atau mekanisme pemulihan otomatis yang diterapkan sebagai akibat langsung dari wawasan dari eksperimen chaos.
- Kepatuhan - Melacak biaya dan bekerja untuk memenuhi persyaratan peraturan atau standar industri yang terkait dengan ketahanan operasional.
- Adopsi - Lacak tingkat adopsi praktik kekacauan di seluruh organisasi.
- Kepuasan pelanggan — Mengukur perubahan dalam metrik kepuasan pelanggan untuk mengukur bagaimana peningkatan keandalan sistem mempengaruhi bisnis.

Langkah-langkah kualitatif

Langkah-langkah kualitatif berikut menyediakan kerangka kerja untuk melacak hasil organisasi yang lebih luas yang dicapai melalui praktik rekayasa kekacauan:

- Keyakinan dan kesiapan karyawan:
  - Tim survei secara berkala untuk mengukur tingkat kepercayaan mereka dalam menangani insiden dunia nyata dan kesiapan mereka yang dirasakan untuk rotasi on-call.
  - Lacak persentase insinyur panggilan yang telah berpartisipasi dalam eksperimen kekacauan sebagai bagian dari pelatihan mereka.
- Pergeseran budaya:
  - Menilai sejauh mana pola pikir ketahanan telah meresap ke organisasi melalui survei, sesi umpan balik, atau audit.
  - Lacak jumlah tim yang secara aktif memperjuangkan dan mengadvokasi praktik rekayasa kekacauan.
- Kolaborasi lintas fungsi dan berbagi pengetahuan:
  - Lacak frekuensi dan kehadiran sesi berbagi pengetahuan lintas tim atau lokakarya yang terkait dengan pembelajaran teknik kekacauan.
  - Lacak jumlah inisiatif rekayasa kekacauan bersama yang melibatkan banyak tim atau departemen.
- Efektivitas pelatihan:
  - Mengevaluasi efektivitas program pelatihan rekayasa kekacauan dengan melakukan survei atau penilaian pasca-pelatihan.
  - Lacak jumlah insinyur yang berpartisipasi dalam program pelatihan rekayasa kekacauan dan baca postmortem.

- Daya tarik dan retensi bakat:
  - Mengevaluasi apakah program rekayasa kekacauan membantu menarik dan mempertahankan bakat teknik terbaik dengan mengurangi waktu dan upaya yang dihabiskan untuk memperbaiki pemadaman.
- Reputasi merek:
  - Lacak setiap perubahan dalam persepsi merek atau reputasi yang terkait dengan komitmen organisasi yang ditunjukkan terhadap ketahanan operasional.
- Keunggulan kompetitif:
  - Lacak keunggulan kompetitif dibandingkan rekan-rekan industri dalam hal ketersediaan sistem.

Lampiran C - Klasifikasi insiden

Melacak insiden dalam kerangka klasifikasi sangat penting karena kerangka kerja memberikan pandangan holistik tentang jenis kegagalan dan masalah yang berdampak pada sistem. Jika organisasi Anda melacak insiden hanya dalam satu kelas, seperti kesalahan infrastruktur, Anda mungkin kehilangan wawasan dan peluang untuk perbaikan di bidang lain. Dengan melacak insiden di berbagai kelas, Anda mendapatkan pemahaman yang lebih baik tentang beragam eksperimen chaos yang harus dilakukan. Perspektif ini membantu mengidentifikasi titik buta potensial dan mendukung perluasan ruang lingkup teknik, yang mengarah pada sistem yang lebih tangguh dan toleran terhadap kesalahan.

Kerangka klasifikasi insiden yang disarankan dirancang untuk membantu mengkategorikan insiden berdasarkan sifat dan potensi dampaknya. Ini menggunakan klasifikasi tingkat tinggi yang mengelompokkan insiden ke dalam delapan kategori utama:

- Masalah penerapan:
  - Penerapan gagal
  - Kegagalan rollback
  - Masalah konfigurasi selama penerapan
- Bug dan regresi perangkat lunak:
  - Bug fungsional
  - Masalah integrasi
  - Masalah kinerja
  - Masalah kuota
  - Masalah mekanisme ketahanan (percobaan ulang, batas waktu)
  - Masalah integritas data
- Masalah pengujian:
  - Tes yang hilang
  - Tes tidak efektif
  - Tes bersisik
- Kesalahan infrastruktur:
  - Kegagalan perangkat keras (server, perangkat jaringan, penyimpanan)
  - Masalah penskalaan

    - Kegagalan ketergantungan (layanan pihak ketiga, APIs)
    - Masalah konektivitas jaringan
- Masalah operasional:
    - Kesalahan manusia (salah konfigurasi, perubahan tidak disengaja)
    - Memantau dan mengingatkan kegagalan
    - Masalah perencanaan kapasitas
    - Kegagalan Backup dan Restore
- Insiden keamanan:
    - Upaya akses tidak sah
    - Pelanggaran data
    - Serangan Denial of Service (DoS)
- Pemadaman layanan pihak ketiga:
    - Pemadaman penyedia cloud
    - Kegagalan DNS
    - Gangguan API dan layanan eksternal
- Faktor lingkungan:
    - Bencana alam (gempa bumi, kebakaran, banjir, pemadaman listrik)
    - Masalah terkait cuaca

Ini adalah kerangka klasifikasi contoh yang tidak meyakinkan yang dapat Anda sesuaikan agar sesuai dengan kebutuhan dan organisasi spesifik Anda. Kami merekomendasikan untuk meninjau dan memperbarui kerangka klasifikasi secara berkala saat sistem Anda berkembang atau jenis insiden baru muncul.

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	Januari 28, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

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

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

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

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat database manajemen konfigurasi.

repositori kode

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

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

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

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

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

CV

Lihat visi komputer.

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat bahasa definisi database.

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

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

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

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

pertukaran data elektronik (EDI)

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

enkripsi

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

kunci enkripsi

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

endianness

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

titik akhir

Lihat titik akhir layanan.

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

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

epik

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

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

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

FM

Lihat model pondasi.

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

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

strategi greenfield

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

pagar pembatas

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

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

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

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

data penahanan

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

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

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

aplikasi idle

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

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretabilitas

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

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

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

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

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

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

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

cabang utama

Lihat cabang.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

mekanisme

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

akun anggota

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

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat pembelajaran mesin.

modernisasi

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

penilaian kesiapan modernisasi

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

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

aplikasi monolitik (monolit)

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

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

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

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

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

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

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

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

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

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

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

ketekunan poliglot

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

penilaian portofolio

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

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona yang dihosting pribadi

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

kontrol proaktif

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

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

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

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat Retrieval Augmented Generation.

ransomware

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

Matriks RASCI

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

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

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

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat 7 Rs.

Wilayah

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

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Tipe dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

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

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

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

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan

detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans EC2 Amazon, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

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

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

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

SPOF

Lihat satu titik kegagalan.

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

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

variabel target

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

daftar tugas

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

lingkungan uji

Lihat lingkungan.

pelatihan

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

gerbang transit

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

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

tugas yang tidak terdiferensiasi

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

lingkungan atas

Lihat lingkungan.

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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