Strategi dan praktik terbaik untuk migrasi AWS besar

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Strategi dan praktik terbaik untuk migrasi AWS besar

Table of Contents

Strategi dan praktik terbaik untuk migrasi AWS besar

Amazon Web Services (kontributor)

Mei 2022 (riwayat dokumen)

Banyak AWS pelanggan ingin memigrasikan sejumlah besar server dan aplikasi ke AWS Cloud secepat mungkin dengan dampak paling kecil terhadap bisnis mereka. Organisasi Anda mungkin memulai proyek migrasi besar karena sewa pusat data mendekati pembaruan atau penghentian atau karena organisasi Anda mengambil langkah pertama dalam transformasi teknologi. Namun, skala besar tidak diukur hanya dengan jumlah server dalam ruang lingkup. Ini juga memperhitungkan tingkat transformasi organisasi yang dihasilkan dari migrasi, dengan mempertimbangkan kompleksitas seperti orang, proses, teknologi, dan prioritas.

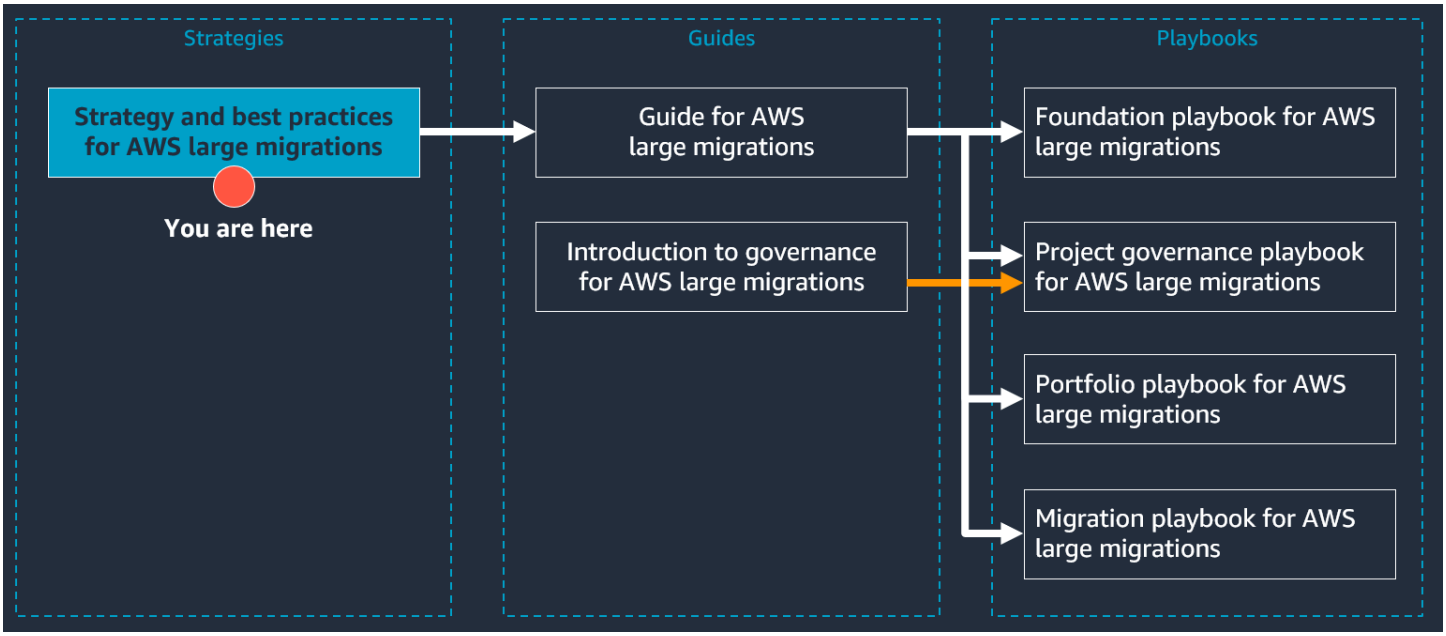
Panduan ini berfokus pada kemampuan Anda untuk bergerak dalam skala besar AWS. Anda dapat memigrasikan aplikasi yang ada dengan sedikit atau tanpa perubahan. Anda dapat menggunakan cloud sebagai titik peluncuran untuk membawa aplikasi tersebut ke teknologi cloud-native atau tanpa server, dan Anda dapat memodernisasi aplikasi untuk membuka manfaat bisnis tambahan.

Panduan ini membahas praktik terbaik untuk migrasi skala besar dan menyediakan kasus penggunaan dari pelanggan di berbagai segmen, seperti layanan keuangan, dan perawatan kesehatan. Ini juga memberikan contoh dunia nyata dari pelajaran yang dipetik selama migrasi pelanggan ke. AWS Tujuan dari panduan ini adalah untuk membantu pelanggan yang berada pada tahap awal migrasi skala besar. Namun, praktik dan strategi terbaik dalam panduan ini dapat bermanfaat pada setiap tahap perjalanan migrasi. Diasumsikan bahwa Anda sudah memiliki pengetahuan 100 tingkat Layanan AWS dan bahwa Anda mengetahui proses yang AWS disarankan untuk bermigrasi.

Panduan untuk migrasi besar

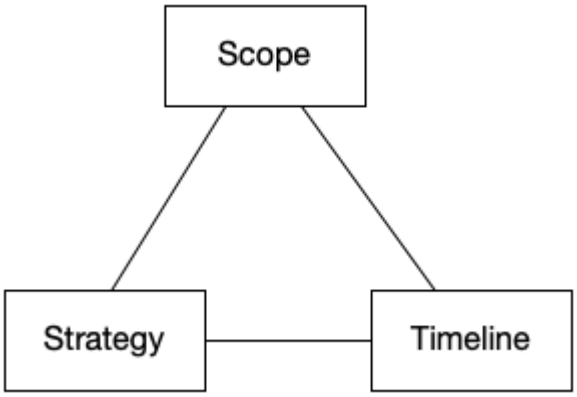
Migrasi 300 atau lebih server dianggap sebagai migrasi besar. Tantangan orang, proses, dan teknologi dari proyek migrasi besar biasanya baru bagi sebagian besar perusahaan. Dokumen ini adalah bagian dari seri Panduan AWS Preskriptif tentang migrasi besar ke. AWS Cloud Seri ini dirancang untuk membantu Anda menerapkan strategi dan praktik terbaik yang benar sejak awal, untuk merampingkan perjalanan Anda ke cloud.

Gambar berikut menunjukkan dokumen lain dalam seri ini. Tinjau strategi terlebih dahulu, lalu panduannya, lalu lanjutkan ke buku pedoman. Untuk mengakses seri lengkap, lihat Migrasi besar ke file. AWS Cloud



Ruang lingkup, strategi, dan garis waktu

Tiga elemen kunci membentuk blok bangunan dari semua program dan relevansinya dalam migrasi besar: ruang lingkup, strategi, dan garis waktu.



Untuk mengatur tahapan perjalanan migrasi Anda, elemen-elemen ini harus disejajarkan dan dipahami sejak awal program migrasi. Setiap perubahan pada salah satu elemen ini akan mempengaruhi yang lain. Penataan kembali harus diperhitungkan dalam setiap perubahan, tidak peduli seberapa dasar atau masuk akal perubahan itu.

Cakupan — Apa yang Anda migrasi?

Adalah umum untuk cakupan total program menjadi tidak terdefinisi, bahkan ketika Anda setengah jalan melalui migrasi. Ini karena berbagai faktor mungkin tidak dibongkar sampai tahap selanjutnya. Misalnya, di tengah migrasi, Anda mungkin menemukan kantong TI bayangan yang tidak direkam dalam database manajemen konfigurasi (CMDB) Anda. Atau, perencanaan mungkin berfokus pada tampilan server tanpa mempertimbangkan jaringan pendukung dan layanan keamanan yang diperlukan untuk menjalankan aplikasi tersebut (seperti koneksi VPN ke AWS Mitra, dan otoritas sertifikat untuk menandatangani sertifikat). Kami merekomendasikan menginvestasikan waktu dalam menentukan ruang lingkup, bekerja mundur dari hasil bisnis target Anda. Anda mungkin akhirnya menggunakan alat penemuan untuk mengungkap aset, praktik terbaik yang akan dibahas nanti dalam panduan ini.

Ruang lingkup akan berubah, karena migrasi besar datang dengan yang tidak diketahui. Ketidaktahuan ini bisa dalam bentuk sistem yang telah menjadi bagian dari arkeologi lingkungan dengan sedikit atau tanpa pemahaman tentang relevansinya, atau insiden produksi yang menyebabkan penundaan dan pergeseran ke rencana yang telah Anda buat. Kuncinya adalah fleksibel dan memiliki rencana darurat untuk menjaga program tetap bergerak maju.

Strategi — Mengapa Anda ingin bermigrasi?

Anda mungkin berencana untuk bermigrasi ke AWS karena satu atau beberapa alasan berikut:

- Tim aplikasi Anda ingin menerapkan CI/CD pipeline baru, menyebarkan tumpukan aplikasi terbaru, atau memodernisasi platform lama yang tidak didukung.
- Tim infrastruktur Anda harus keluar dari pusat data yang sudah tua dengan cepat sebelum masa sewa berakhir dan penyedia mematikan daya.
- Dewan telah memutuskan bahwa Anda perlu pindah ke cloud sebagai arah strategis, memungkinkan langkah cepat perubahan di masa depan bisnis.

Apa pun alasannya, semua alasan ini dan lebih banyak lagi akan ada di benak bisnis dan organisasi TI Anda. Ini adalah kunci untuk memahami apa pengemudi Anda, untuk mengkomunikasikannya, dan memprioritaskan mereka. Setiap driver tambahan berpotensi menambah waktu, biaya, ruang lingkup, dan risiko untuk migrasi Anda yang sudah besar. Menyadari sepenuhnya dampak strategi terhadap garis waktu dan ruang lingkup adalah kuncinya.

Setelah Anda menentukan strategi migrasi Anda, salah satu kunci utama keberhasilan adalah penyelarasan persyaratan di berbagai pemangku kepentingan dan tim. Melakukan migrasi memerlukan tim yang berbeda di seluruh organisasi, termasuk Infrastruktur, Keamanan, Aplikasi, dan Operasi. Tim-tim ini akan memiliki prioritas individu dan proyek lain yang mungkin sudah dimulai. Jika tim-tim ini bekerja menuju jadwal dan prioritas yang berbeda, akan lebih sulit untuk menyetujui dan menerapkan rencana migrasi. Tim migrasi dan pemangku kepentingan utama harus memastikan bahwa semua tim yang terlibat bekerja menuju satu tujuan dan menyelaraskan prioritas mereka dengan satu garis waktu migrasi.

Kami merekomendasikan untuk mengeksplorasi bagaimana hasil bisnis yang diinginkan dapat diselaraskan di berbagai tim. Misalnya, bermigrasi ke AWS dan menggunakan AWS Key Management Service (AWS KMS) untuk mengenkripsi penyimpanan saat istirahat mungkin memenuhi tujuan migrasi dan keamanan.

Seringkali, bisnis ingin memodernisasi aplikasi, yang dapat menghasilkan peningkatan infrastruktur, sementara tim infrastruktur ingin hemat dan meminimalkan perubahan infrastruktur. Pola pikir untuk migrasi besar harus sedasar mungkin. Tim yang terlibat harus menghindari mencoba melakukan semuanya sekaligus.

Untuk mencapai ini, tetapkan harapan yang tepat di awal proyek. Pesan kuncinya adalah “Migrasi dulu, lalu modernisasi.” Pendekatan ini tidak hanya memungkinkan organisasi untuk mengurangi

utang teknis dan beroperasi pada skala pada akhirnya, tetapi juga membuka jalan bagi pendekatan modernisasi yang berbeda dengan menggunakan skalabilitas dan kelincahan yang dapat disediakan. AWS Cloud Berpikir jangka panjang akan membantu tim infrastruktur untuk merampingkan penyebaran dan manajemen infrastruktur. Akibatnya bisnis dapat memiliki siklus rilis fitur yang lebih cepat.

Timeline — Kapan Anda perlu menyelesaikan migrasi?

Tergantung pada kasus bisnis Anda, Anda harus memastikan bahwa Anda tidak mengambil lebih dari yang mungkin dicapai dalam waktu yang dialokasikan. Jika driver Anda untuk migrasi didasarkan pada tanggal penyelesaian yang tetap, Anda harus memilih strategi yang memenuhi persyaratan timeline tersebut. Sebagian besar migrasi besar didasarkan pada kendala berbasis waktu ini, sehingga strategi migrasi harus telah menentukan, jadwal dan hasil tetap, dengan sedikit ruang untuk ekstensi atau overrun.

Dalam jenis migrasi yang sensitif terhadap waktu ini, kami merekomendasikan pendekatan “Migrasi dulu, lalu modernisasi”. Ini membantu menetapkan harapan dan mendorong tim untuk memastikan bahwa rencana proyek dan anggaran masing-masing selaras dengan tujuan migrasi secara keseluruhan. Penting untuk mengetahui ketidaksepakatan sedini mungkin dalam proyek, gagal dengan cepat dan mengatasi ketidaksepakatan di tingkat Komite Pengarah, dan melibatkan pemangku kepentingan yang tepat untuk memastikan bahwa keselarasan sudah ada.

Sebaliknya, jika tujuan utama migrasi Anda adalah untuk mendapatkan manfaat dari modernisasi aplikasi, ini harus dipanggil di awal program. Banyak program dimulai dengan tujuan awal berdasarkan tenggat waktu yang tetap, dan mereka tidak merencanakan persyaratan dari pemangku kepentingan yang ingin menyelesaikan masalah dan masalah yang belum selesai. Dalam beberapa kasus, masalah ini telah hadir selama bertahun-tahun dalam sistem sumber, tetapi sekarang mereka menjadi penghambat buatan untuk migrasi.

Kegiatan modernisasi selama migrasi dapat memengaruhi fungsionalitas aplikasi bisnis. Bahkan apa yang dianggap sebagai peningkatan kecil, seperti perubahan versi sistem operasi, dapat memiliki efek besar pada jadwal program. Ini seharusnya tidak dianggap sepele.

Praktik terbaik untuk migrasi besar

Migrasi besar dapat menjadi tantangan, tergantung pada faktor-faktor yang mengatur bagaimana suatu organisasi berfungsi. Bagian ini mencakup beberapa faktor kunci yang dapat menyederhanakan migrasi besar jika ditangani selama fase awal upaya dan dilacak di seluruh proyek.

Praktik terbaik berikut untuk migrasi besar didasarkan pada data yang diambil dari pelanggan lain. Praktik terbaik dibagi menjadi tiga kategori:

- Orang
- Teknologi
- Proses

Perspektif orang

Bagian ini berfokus pada bidang-bidang utama berikut dari perspektif orang:

- Dukungan eksekutif — Mengidentifikasi pemimpin berulir tunggal yang diberdayakan untuk membuat keputusan
- Kolaborasi dan kepemilikan tim — Berkolaborasi di antara berbagai tim
- Pelatihan — Tim pelatihan proaktif pada berbagai perkakas

Dukungan eksekutif

Di bagian ini:

Identifikasi pemimpin berulir tunggal

Saat memulai migrasi besar, penting untuk mengidentifikasi pemimpin teknis berulir tunggal yang 100 persen berdedikasi pada proyek dan bertanggung jawab. Pemimpin itu diberdayakan untuk membuat keputusan, membantu menghindari silo, dan merampingkan aliran kerja dengan mempertahankan prioritas yang konsisten.

Pelanggan global migrasi besar dapat menskalakan dari satu server setiap minggu pada awal program ke lebih dari 80 server setiap minggu pada awal bulan kedua. Dukungan penuh CIO sebagai pemimpin single-threaded sangat penting untuk peningkatan pesat server yang dimigrasikan. CIO menghadiri panggilan cutover migrasi mingguan dengan tim migrasi untuk memastikan eskalasi real-time dan penyelesaian masalah, yang mempercepat kecepatan migrasi.

Sejajarkan tim kepemimpinan senior

Sangat penting untuk menciptakan keselarasan antara berbagai tim mengenai kriteria keberhasilan migrasi. Sementara perencanaan dan implementasi migrasi dapat dicapai oleh tim kecil yang berdedikasi, tantangan muncul ketika mendefinisikan strategi dan melakukan kegiatan periferal. Hambatan potensial ini mungkin memerlukan tindakan atau eskalasi dari berbagai bidang organisasi TI, termasuk yang berikut:

- Bisnis
- Aplikasi
- Jaringan
- Keamanan
- Infrastruktur
- Vendor pihak ketiga

Tindakan langsung dari pemilik aplikasi, kepemimpinan, penyelarasan, dan eskalasi yang jelas ke pemimpin berulir tunggal menjadi penting.

Kolaborasi dan kepemilikan tim

Di bagian ini:

Membuat tim pemberdayaan cloud lintas fungsi

Langkah pertama yang penting dalam proyek migrasi besar adalah memungkinkan organisasi bekerja di cloud. Untuk mencapai hal ini, kami sarankan untuk membangun Cloud Enablement

Engine (CEE). CEE adalah tim yang diberdayakan dan bertanggung jawab yang berfokus pada kesiapan operasional organisasi untuk migrasi ke. AWS CEE harus menjadi tim lintas fungsi yang mencakup representasi dari infrastruktur, aplikasi, operasi, dan keamanan. Tim ditugasi dengan tanggung jawab berikut:

- Mengembangkan kebijakan
- Mendefinisikan dan mengimplementasikan alat, proses, dan arsitektur yang akan membentuk model operasi cloud organisasi
- Terus memfasilitasi penyelarasan pemangku kepentingan di semua area yang mereka wakili

Satu pelanggan layanan kesehatan tidak memulai dengan CEE. Namun, melalui migrasi pilot awal, celah itu diidentifikasi. Menjelang tanggal pemotongan migrasi terakhir, dengan tenggat waktu yang ketat, tim menerapkan ruang perang migrasi. Di ruang perang migrasi, pemangku kepentingan dari infrastruktur, keamanan, aplikasi, dan bisnis dapat membantu menyelesaikan masalah.

Tentukan persyaratan untuk tim dan individu di luar tim migrasi inti terlebih dahulu

Identifikasi tim dan individu yang berada di luar program inti, dan tentukan keterlibatan mereka selama fase perencanaan migrasi. Untuk memfasilitasi momentum migrasi selama tahap selanjutnya, berikan perhatian khusus pada keterlibatan tim aplikasi. Pengetahuan mereka tentang aplikasi, kemampuan untuk mendiagnosis masalah, dan persyaratan untuk menandatangani cutover akan diperlukan.

Meskipun migrasi akan dipimpin oleh tim inti, tim aplikasi kemungkinan akan terlibat dalam memvalidasi rencana migrasi dan pengujian selama pemotongan. Pelanggan sering mendekati migrasi cloud sebagai proyek infrastruktur, bukan sebagai migrasi aplikasi. Hal ini dapat menyebabkan masalah selama migrasi.

Sebaiknya pertimbangkan keterlibatan tim aplikasi yang diperlukan saat memilih strategi migrasi. Misalnya, strategi rehost membutuhkan lebih sedikit keterlibatan tim aplikasi dibandingkan dengan strategi replatform atau refactor di mana lebih banyak lanskap aplikasi sedang diubah. Jika ketersediaan pemilik aplikasi terbatas, pertimbangkan untuk menggunakan rehost atau replatform sebagai lawan dari strategi refactor, relokasi, atau pembelian kembali.

Validasi bahwa tidak ada masalah lisensi saat memigrasi beban kerja

Perizinan dapat berubah saat Anda memigrasikan perusahaan secara off—produk yang tersedia ke cloud. Perjanjian lisensi Anda mungkin difokuskan pada properti lokal Anda. Misalnya, lisensi

mungkin dengan CPU atau ditautkan ke alamat MAC tertentu. Atau, perjanjian lisensi mungkin tidak mencakup hak untuk menjadi tuan rumah di lingkungan cloud publik. Namun, negosiasi ulang lisensi dengan vendor dapat mencakup waktu tunggu yang lama dan menghadirkan hard blocker untuk migrasi.

Sebaiknya berkolaborasi dengan tim manajemen sumber atau vendor Anda segera setelah ruang lingkup migrasi ditentukan. Perizinan juga dapat memengaruhi arsitektur target dan pola migrasi Anda.

Pelatihan

Di bagian ini:

- Latih tim tentang perkakas dan proses baru

Latih tim tentang perkakas dan proses baru

Setelah strategi migrasi ditentukan, investasikan waktu untuk memahami pelatihan apa yang mungkin diperlukan untuk migrasi dan untuk model operasi target Anda. Selama migrasi, Anda mungkin akan menggunakan tooling, seperti AWS Database Migration Service, yang baru untuk organisasi Anda. Tim pelatihan secara proaktif mengurangi penundaan yang dialami selama fase migrasi.

Kami merekomendasikan mencari metode transfer pengetahuan aktif yang memberikan kesempatan untuk bereksperimen dengan perkakas secara langsung. Sebagai contoh, Layanan AWS Profesional menyediakan beberapa sesi pelatihan Cloud Migration Factory untuk tiga AWS Mitra integrator sistem (SI) yang bertanggung jawab atas migrasi besar. Ini memastikan bahwa tim memiliki keakraban dasar saat pindah ke fase migrasi. Ini juga membantu mengidentifikasi ahli materi pelajaran (SMEs) yang dapat berfungsi sebagai eskalasi lini pertama dalam setiap tim AWS Mitra SI.

Perspektif teknologi

Teknologi memberikan fondasi yang bagus untuk mempercepat migrasi besar. Misalnya, solusi Cloud Migration Factory difokuskan pada cara menyediakan end-to-end otomatisasi untuk migrasi. Bagian ini mengeksplorasi beberapa praktik terbaik untuk menggunakan teknologi untuk mencapai skala dan kecepatan yang diperlukan, selaras dengan ruang lingkup, strategi, dan garis waktu.

Prinsip menyeluruh adalah melihat area otomatisasi sedapat mungkin. Jika Anda memiliki ribuan server dalam ruang lingkup, melakukan tugas secara manual dapat menjadi upaya yang mahal dan memakan waktu.

Untuk melakukan migrasi, beberapa alat biasanya digunakan, seperti berikut ini:

- Penemuan
- Implementasi migrasi
- Database manajemen konfigurasi (CMDB)
- Spreadsheet inventaris
- Manajemen proyek

Alat-alat ini digunakan pada berbagai tahap migrasi, mulai dari penilaian hingga mobilisasi hingga implementasi. Pemilihan alat-alat ini didorong oleh tujuan bisnis dan jadwal.

Setelah fase migrasi direncanakan, langkah selanjutnya adalah memastikan bahwa tim migrasi memiliki keterampilan untuk menggunakan alat yang mereka butuhkan. Jika tim tidak memiliki keterampilan atau pengalaman, rencanakan pelatihan yang ditargetkan untuk meningkatkan keahlian. Jika memungkinkan, buat acara di mana tim bisa mendapatkan pengalaman dengan alat migrasi di lingkungan yang aman. Misalnya, apakah ada server sandpit atau lab yang dapat dimigrasikan oleh tim untuk mengalami perkakas? Atau, apakah beban kerja pengembangan awal dapat diterima untuk digunakan untuk tujuan pembelajaran?

Integrasi otomatisasi, pelacakan, dan perkakas

Di bagian ini:

Otomatiskan penemuan migrasi untuk mengurangi waktu yang dibutuhkan

Sebagian besar program migrasi besar dimulai dengan memahami ruang lingkup migrasi (apa yang harus dimigrasikan) dan mengembangkan strategi (bagaimana hal itu akan dimigrasikan). Penemuan adalah aspek penting dari ini. Titik metadata yang diperlukan ditangkap untuk membentuk pohon keputusan strategi migrasi. Untuk memigrasikan beban kerja dengan cepat, Anda harus mengidentifikasi dan mengimpor metadata migrasi yang diperlukan ke dalam proses implementasi, seperti pabrik migrasi. Mekanisme yang sepenuhnya otomatis untuk mengekstrak, mengubah,

memuat (ETL) metadata migrasi sangat mengurangi waktu dan tingkat upaya yang terlibat dalam proses penemuan.

Satu pelanggan mengembangkan proses pengambilan data yang sepenuhnya otomatis untuk pabrik migrasi mereka. Rencana gelombang migrasi dengan semua metadata migrasi dihosting dan dipelihara dalam spreadsheet di Microsoft. SharePoint Ketika perubahan dilakukan pada sumber, AWS Lambda fungsi dimulai untuk memuat data ke pabrik migrasi tanpa intervensi manual. Proses pengambilan data otomatis ini membantu pelanggan mengurangi pekerjaan manual, meminimalkan kesalahan manusia, dan mempercepat kecepatan mereka. Mereka dapat memigrasikan lebih dari 1.000 server ke AWS.

Otomatiskan tugas berulang

Dalam fase implementasi migrasi, banyak proses kecil harus sering diulang. Saat menggunakan AWS Application Migration Service (MGN), misalnya, Anda harus menginstal agen di setiap server yang berada dalam lingkup migrasi.

Membangun pabrik migrasi yang sesuai dengan kebutuhan bisnis dan teknis spesifik Anda adalah cara paling efektif untuk mencapai efisiensi dan kecepatan yang diperlukan untuk menghasilkan migrasi besar yang sukses. Pabrik migrasi menyediakan kerangka kerja integrasi dan orkestrasi yang menggunakan kumpulan data standar untuk mempercepat migrasi. Setelah semua tugas diidentifikasi, habiskan waktu untuk mengotomatiskan semua tugas manual yang dapat diotomatisasi bersama runbook preskriptif.

Solusi Cloud Migration Factory adalah contohnya. Cloud Migration Factory dirancang untuk menyediakan fondasi otomatisasi migrasi tempat Anda dapat mengotomatiskan aspek yang spesifik untuk organisasi Anda. Misalnya, Anda mungkin ingin memperbarui tanda di CMDB untuk menyoroti bahwa server lokal sekarang dapat dinonaktifkan. Dalam skenario ini, Anda dapat membuat otomatisasi yang melakukan tugas ini di akhir gelombang migrasi. Cloud Migration Factory memiliki penyimpanan metadata terpusat dengan semua gelombang, aplikasi, dan metadata server. Skrip otomatisasi dapat terhubung ke Cloud Migration Factory untuk mendapatkan daftar server dalam gelombang itu dan melakukan tindakan apa pun yang sesuai. Cloud Migration Factory mendukung AWS Application Migration Service.

Otomatiskan pelacakan dan pelaporan untuk mempercepat pengambilan keputusan

Sebaiknya buat dasbor pelaporan migrasi otomatis untuk melacak dan melaporkan data langsung, termasuk indikator kinerja utama (KPIs) untuk program. Proyek migrasi melibatkan pemangku kepentingan dari seluruh organisasi, termasuk yang berikut:

- Tim aplikasi
- Penguji
- Tim penonaktifan
- Arsitek
- Tim infrastruktur
- Kepemimpinan

Untuk menjalankan peran mereka, para pemangku kepentingan ini membutuhkan data langsung. Misalnya, tim jaringan harus mengetahui gelombang migrasi yang akan datang untuk memahami dampak pada koneksi bersama antara sumber daya lokal dan AWS. Tim kepemimpinan ingin memahami seberapa banyak migrasi selesai. Memiliki umpan data langsung yang dapat diandalkan dan otomatis mencegah miskomunikasi dan memberikan dasar di mana keputusan dapat dibuat.

Seorang pelanggan layanan kesehatan besar sedang bekerja menuju pintu keluar pusat data dengan tenggat waktu yang akan datang. Mengingat skala dan kompleksitasnya, sejumlah besar waktu awalnya dihabiskan untuk melacak dan mengkomunikasikan status migrasi antar pemangku kepentingan. Tim migrasi kemudian menggunakan Amazon Quick Sight untuk membangun dasbor otomatis yang memvisualisasikan data, secara signifikan menyederhanakan pelacakan dan komunikasi sambil meningkatkan kecepatan migrasi.

Jelajahi perkakas yang dapat memfasilitasi migrasi Anda

Memilih alat yang tepat untuk migrasi Anda tidaklah mudah, terutama jika tidak ada seorang pun di organisasi Anda yang pernah mengelola migrasi besar sebelumnya.

Sebaiknya luangkan waktu untuk memilih alat yang sesuai untuk mendukung migrasi. Eksplorasi ini mungkin melibatkan biaya lisensi, tetapi dapat memberikan manfaat biaya ketika Anda mempertimbangkan inisiatif yang lebih luas. Atau, Anda mungkin menemukan bahwa perkakas yang disematkan di organisasi Anda dapat memberikan hasil yang serupa. Misalnya, Anda mungkin sudah memiliki alat pemantauan kinerja aplikasi yang digunakan di seluruh perkebunan Anda, yang dapat memberikan informasi penemuan yang kaya.

Seorang pelanggan teknologi awalnya enggan untuk menjalankan alat penemuan otomatis selama migrasi mereka karena kurangnya keakraban. Akibatnya, AWS Mitra SI harus menjalankan 510 jam rapat per aplikasi untuk menemukan estate secara manual, termasuk nama server, versi sistem operasi, dan dependensi. Diperkirakan bahwa jika alat penemuan telah digunakan, upaya penemuan bisa dikurangi lebih dari 1.000 jam.

Prasyarat dan validasi pasca migrasi

Di bagian ini:

Bangun landing zone selama fase pra-migrasi

Kami merekomendasikan untuk membangun lingkungan AWS target, atau landing zone, sebelumnya, daripada membangun target virtual private cloud (VPCs) dan subnet selama gelombang migrasi. Membangun landing zone yang dirancang dengan baik merupakan prasyarat untuk migrasi. Landing zone harus mencakup pemantauan, tata kelola, operasional, dan kontrol keamanan.

Membangun dan memvalidasi landing zone sebelum migrasi meminimalkan ketidakpastian yang datang dengan menjalankan beban kerja Anda di lingkungan baru. Dengan adanya landing zone, para pemangku kepentingan dapat fokus pada migrasi beban kerja tanpa mengkhawatirkan aspek yang dikelola di tingkat akun atau VPC.

Garis besar kegiatan prasyarat

Di samping landing zone, penting untuk menyelaraskan prasyarat teknis lainnya sebelum migrasi, terutama proses dengan lead time yang panjang. Misalnya, buat perubahan firewall yang diperlukan untuk memungkinkan data direplikasi dari tempat ke AWS lokasi. Mengkomunikasikan prasyarat teknis sejak dini membantu mempersiapkan dan mengalokasikan sumber daya yang dibutuhkan. Migrasi sering terhenti karena prasyarat belum terpenuhi. Hal ini tidak hanya berdampak pada gelombang migrasi yang sedang berlangsung, ini mungkin mendorong kembali tanggal semua migrasi masa depan saat masalah sedang diperbaiki.

Sebuah perusahaan jasa keuangan dimaksudkan untuk melakukan migrasi massal ke AWS, dengan tujuan mengosongkan beberapa pusat data. Namun, bandwidth mereka tersedia antara lokal dan tidak AWS cukup untuk kecepatan yang mereka inginkan. Sayangnya, meningkatkan bandwidth membutuhkan koneksi baru dan memiliki lead time tiga bulan. Ini berarti bahwa kecepatan migrasi dibatasi selama tiga bulan pertama.

Menerapkan pemeriksaan pasca-migrasi untuk perbaikan berkelanjutan

Terakhir, ingatlah untuk menerapkan validasi pasca-migrasi seperti integrasi operasi, optimalisasi biaya, dan pemeriksaan tata kelola dan kepatuhan. Validasi pasca-migrasi mencakup penilaian beban kerja yang dimigrasi sebelumnya untuk mengungkap pelajaran teknis yang harus diterapkan pada gelombang masa depan.

Selanjutnya, ini adalah peluang besar untuk menerapkan operasi pengendalian biaya. Misalnya, selama migrasi, Anda mungkin memutuskan untuk mencocokkan ukuran AWS instans dengan estat lokal untuk mengurangi kebutuhan pengujian kinerja. Sekarang pengujian tidak lagi berada di jalur kritis penutupan pusat data, Anda dapat menggunakan Amazon CloudWatch untuk menilai pemanfaatan instance dan menentukan apakah instance berukuran lebih kecil akan cocok.

Untuk menggambarkan pentingnya fase ini, pelanggan teknologi besar melakukan migrasi besar tetapi awalnya tidak menyertakan validasi pasca-migrasi. Setelah memigrasikan lebih dari 100 server, mereka mengidentifikasi bahwa AWS Systems Manager Agen (Agen SSM) tidak dikonfigurasi dengan benar. Semua server yang sebelumnya dimigrasi harus diperbaiki, dan migrasi terhenti. Pelanggan juga mengidentifikasi bahwa instans sebesar lima kali perkiraan awal, sehingga mereka menerapkan pos pemeriksaan biaya pada akhir setiap gelombang migrasi.

Perspektif proses

Proses membawa konsistensi tetapi mereka juga berkembang dan rentan terhadap perubahan karena setiap proyek unik. Saat Anda menjalankan proses berulang kali, Anda akan mengidentifikasi celah dan ruang untuk perbaikan yang dapat menambah manfaat besar saat Anda gagal, belajar, mengadopsi, dan mengulangi. Perubahan ini dapat mengarah pada ide atau inovasi baru yang dapat dimanfaatkan oleh proyek dan bisnis di masa depan, yang memberikan katalis untuk pertumbuhan yang membawa kualitas dan kepercayaan tim.

Proses dalam migrasi dapat menjadi kompleks karena melintasi teknologi dan batas yang mungkin tidak terkait sebelumnya. Perspektif ini memberikan proses dan panduan tentang persyaratan khusus untuk migrasi besar.

Mempersiapkan migrasi besar Anda

Bagian berikut menguraikan prinsip-prinsip inti yang diperlukan untuk memastikan bahwa Anda memulai perjalanan migrasi Anda dengan arah yang jelas dan dukungan dari para pemangku kepentingan yang akan sangat penting untuk keberhasilannya.

Di bagian ini:

Tentukan driver bisnis dan komunikasikan garis waktu, ruang lingkup, dan strategi

Saat mendekati migrasi besar ke AWS, Anda akan segera menemukan bahwa ada banyak cara untuk memigrasikan server Anda. Sebagai contoh, Anda dapat melakukan hal berikut:

- Rehost beban kerja menggunakan. AWS Application Migration Service
- Kontainerisasi aplikasi Anda dan host di Amazon Elastic Container Service (Amazon ECS) atau platform container terkelola Amazon Elastic Kubernetes Service (Amazon EKS).
- Desain ulang beban kerja Anda menjadi aplikasi tanpa server sepenuhnya.

Untuk menentukan jalur migrasi yang benar, penting untuk bekerja mundur dari driver bisnis Anda. Jika tujuan akhir Anda adalah untuk meningkatkan kelincahan bisnis, Anda mungkin menyukai dua pola kedua, yang melibatkan lebih banyak tingkat transformasi. Jika tujuan Anda adalah mengosongkan pusat data pada akhir tahun, Anda dapat memilih untuk menghosting ulang beban kerja karena kecepatan yang disediakan rehosting.

Migrasi besar biasanya melibatkan berbagai pemangku kepentingan, termasuk yang berikut:

- Pemilik aplikasi
- Tim jaringan
- Administrator basis data
- Sponsor eksekutif

Ini adalah kunci untuk mengidentifikasi driver bisnis migrasi dan memasukkan daftar itu dalam dokumen, seperti piagam proyek yang dapat diakses oleh anggota program migrasi. Selanjutnya, buat indikator kinerja utama (KPIs) yang selaras dengan hasil bisnis target Anda.

Misalnya, satu pelanggan ingin memigrasikan 2.000 server dalam waktu 12 bulan untuk mencapai hasil bisnis target mereka dari mengosongkan pusat data mereka. Namun, tim keamanan mereka tidak selaras dengan tujuan ini. Hasilnya adalah beberapa bulan perdebatan teknis tentang apakah akan melewatkan tanggal penutupan pusat data tetapi memodernisasi aplikasi lebih lanjut atau untuk meng-host ulang pada awalnya untuk memungkinkan penutupan pusat data yang tepat waktu dan kemudian memodernisasi aplikasi. AWS

Tentukan jalur eskalasi yang jelas untuk membantu menghapus pemblokir

Program migrasi cloud besar biasanya melibatkan berbagai pemangku kepentingan. Lagi pula, Anda berpotensi mengubah aplikasi yang telah di-host di tempat selama beberapa dekade. Adalah umum bagi setiap pemangku kepentingan untuk memiliki prioritas yang saling bertentangan.

Sementara semua prioritas mungkin mendorong nilai, program kemungkinan akan memiliki jumlah anggaran terbatas dan hasil target yang ditentukan. Mengelola berbagai pemangku kepentingan dan berfokus pada target hasil bisnis dapat menjadi tantangan. Tantangan ini diperparah ketika Anda mengalikannya dengan ratusan atau ribuan aplikasi yang berada dalam lingkup migrasi. Selanjutnya, para pemangku kepentingan kemungkinan melaporkan ke dalam tim kepemimpinan yang berbeda, yang memiliki prioritas lain. Dengan pemikiran ini, di samping mendokumentasikan dengan jelas hasil bisnis target, penting untuk menentukan matriks eskalasi yang jelas untuk membantu menghilangkan pemblokir. Ini dapat menghemat banyak waktu dan membantu menyelaraskan berbagai tim menuju tujuan bersama.

Salah satu contoh yang menunjukkan hal ini adalah perusahaan jasa keuangan yang tujuannya adalah untuk mengosongkan pusat data utama mereka dalam waktu 12 bulan. Tidak ada mandat atau jalur eskalasi yang jelas, yang mengakibatkan para pemangku kepentingan menyusun jalur migrasi yang mereka inginkan, terlepas dari kendala waktu dan anggaran. Setelah eskalasi ke CIO, mandat yang jelas ditetapkan dan mekanisme disediakan untuk meminta keputusan yang diperlukan.

Minimalkan perubahan yang tidak perlu

Perubahan itu baik tetapi lebih banyak perubahan berarti lebih banyak risiko. Ketika kasus bisnis untuk migrasi besar disetujui, kemungkinan besar ada hasil bisnis target yang mendorong inisiatif ini, seperti mengosongkan pusat data pada tanggal tertentu. Meskipun umum bagi para teknolog untuk ingin menulis ulang semuanya untuk memanfaatkan AWS layanan sepenuhnya, ini mungkin bukan tujuan bisnis Anda.

Satu pelanggan fokus pada migrasi dua tahun dari seluruh infrastruktur skala web perusahaan ke. AWS Mereka menciptakan aturan dua minggu sebagai mekanisme untuk mencegah tim aplikasi

menghabiskan waktu berbulan-bulan menulis ulang aplikasi mereka. Dengan menggunakan aturan dua minggu, pelanggan dapat mempertahankan migrasi jangka panjang dengan irama yang konsisten ketika ratusan aplikasi harus dipindahkan selama periode beberapa tahun. Untuk informasi selengkapnya, lihat posting blog Aturan Dua Minggu: Memfaktorkan Ulang Aplikasi Anda untuk Cloud dalam 10 Hari.

Kami merekomendasikan meminimalkan perubahan apa pun yang tidak sesuai dengan hasil bisnis. Sebaliknya, bangun mekanisme untuk mengelola perubahan tambahan ini dalam proyek masa depan.

Dokumentasikan end-to-end proses lebih awal

Dokumentasikan proses migrasi lengkap dan penugasan kepemilikan pada tahap awal program migrasi besar. Dokumentasi ini penting dalam mendidik semua pemangku kepentingan tentang bagaimana migrasi akan berjalan dan peran serta tanggung jawab mereka. Dokumentasi juga akan membantu Anda memahami di mana masalah mungkin terjadi dan untuk memberikan pembaruan dan iterasi proses saat Anda maju melalui migrasi.

Selama pengembangan proyek migrasi, pastikan bahwa setiap proses yang ada dipahami dan bahwa titik integrasi dan dependensi didokumentasikan dengan jelas. Sertakan tempat-tempat di mana keterlibatan dengan pemilik proses eksternal akan diperlukan, termasuk permintaan perubahan, permintaan layanan, dukungan vendor, dan dukungan jaringan dan firewall. Setelah proses dipahami, kami merekomendasikan untuk mendokumentasikan kepemilikan dalam matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI) untuk melacak aktivitas yang berbeda. Untuk menyelesaikan proses, buat rencana hitung mundur dengan mengidentifikasi garis waktu yang terlibat dalam setiap langkah migrasi. Rencana hitung mundur umumnya bekerja mundur dari tanggal dan waktu pemotongan migrasi beban kerja.

Pendekatan dokumentasi ini bekerja dengan baik untuk perusahaan peralatan rumah tangga multinasional yang bermigrasi dengan AWS sukses dalam waktu kurang dari setahun dan keluar dari empat pusat data. Mereka memiliki enam tim organisasi yang berbeda dan beberapa pihak ketiga yang terlibat, yang memperkenalkan overhead manajemen yang mengakibatkan back-and-forth keputusan dan keterlambatan implementasi. Tim Layanan AWS Profesional, bersama dengan pelanggan dan pihak ketiga mereka, mengidentifikasi proses utama untuk kegiatan migrasi dan mendokumentasikannya dengan pemilik masing-masing. Matriks RACI yang dihasilkan dibagikan dan disepakati oleh semua tim yang terlibat. Menggunakan matriks RACI dan matriks eskalasi, pelanggan mengurangi pemblokir dan masalah yang menciptakan penundaan. Mereka kemudian dapat keluar dari pusat data lebih cepat dari jadwal.

Dalam contoh lain menggunakan RACI dan matriks eskalasi, perusahaan asuransi dapat keluar dari pusat data dalam waktu kurang dari 4 bulan. Pelanggan memahami dan menerapkan model tanggung jawab bersama, dan matriks RACI terperinci diikuti untuk melacak kemajuan setiap proses dan aktivitas selama migrasi. Akibatnya, pelanggan dapat bermigrasi lebih dari 350 server dalam 12 minggu pertama implementasi.

Dokumen pola migrasi standar dan artefak

Anggap ini sebagai membuat pemotong kue untuk implementasi. Referensi, dokumentasi, runbook, dan pola yang dapat digunakan kembali adalah kunci untuk skala. Jurnal ini adalah pengalaman, pembelajaran, jebakan, masalah, dan solusi yang dapat digunakan kembali dan dihindari oleh proyek migrasi masa depan, yang secara signifikan mempercepat migrasi. Pola dan artefak juga merupakan investasi yang akan membantu meningkatkan proses dan memandu proyek masa depan.

Misalnya, satu pelanggan melakukan migrasi selama setahun di mana aplikasi dimigrasikan oleh tiga Mitra SI AWS yang berbeda. Pada tahap awal, setiap AWS Mitra menggunakan standar, runbook, dan artefak mereka sendiri. Ini menempatkan banyak tekanan pada tim pelanggan, karena informasi yang sama dapat disajikan kepada mereka dengan cara yang berbeda. Setelah sakit awal ini, pelanggan menetapkan kepemilikan pusat atas semua dokumentasi dan artefak untuk digunakan dalam migrasi, dengan proses untuk mengirimkan perubahan yang direkomendasikan. Aset-aset ini meliputi:

- Proses migrasi standar dan daftar periksa
- Gaya diagram jaringan dan standar format
- Aplikasi standar arsitektur dan keamanan berdasarkan kekritisan bisnis

Selain itu, perubahan pada salah satu dokumen dan standar ini dikirim ke semua tim dengan irama mingguan, dan setiap mitra diminta untuk mengkonfirmasi penerimaan dan kepatuhan terhadap perubahan apa pun. Ini sangat meningkatkan komunikasi dan konsistensi untuk proyek migrasi, dan ketika upaya migrasi besar yang terpisah di unit bisnis lain dimulai, tim itu dapat mengadopsi proses dan dokumen yang ada, sangat mempercepat keberhasilan mereka.

Menetapkan satu sumber kebenaran untuk metadata migrasi dan status

Ketika datang untuk merencanakan migrasi besar, membangun sumber kebenaran penting untuk menjaga berbagai tim selaras dan memungkinkan keputusan berbasis data. Ketika Anda memulai perjalanan ini, Anda mungkin menemukan banyak sumber data yang dapat Anda gunakan, seperti

database manajemen konfigurasi (CMDB), alat pemantauan kinerja aplikasi, daftar inventaris, dan sebagainya.

Atau, Anda mungkin menemukan bahwa ada beberapa sumber data dan Anda harus membuat mekanisme untuk menangkap data yang dibutuhkan. Misalnya, Anda mungkin perlu menggunakan alat penemuan untuk mengungkap informasi teknis, dan untuk mensurvei para pemimpin TI untuk mendapatkan informasi bisnis.

Penting untuk menggabungkan berbagai sumber data ke dalam satu kumpulan data yang dapat Anda gunakan untuk migrasi. Anda kemudian dapat menggunakan satu sumber kebenaran untuk melacak migrasi selama implementasi. Misalnya, Anda dapat melacak server mana yang telah dimigrasi.

Pelanggan jasa keuangan yang ingin memigrasikan semua beban kerja untuk AWS fokus pada perencanaan migrasi dengan kumpulan data yang telah disediakan. Dataset ini memiliki kesenjangan kunci, seperti kekritisan bisnis dan informasi ketergantungan, sehingga program memulai latihan penemuan.

Dalam contoh lain, sebuah perusahaan di industri yang sama pindah ke implementasi gelombang migrasi berdasarkan out-of-date pemahaman tentang inventaris infrastruktur server mereka. Mereka dengan cepat mulai melihat jumlah migrasi berkurang karena datanya salah. Dalam hal ini, pemilik aplikasi tidak dipahami, yang berarti bahwa mereka tidak dapat menemukan penguji tepat waktu. Selain itu, data tidak selaras dengan penonaktifan yang telah diselesaikan oleh tim aplikasi mereka, sehingga server berjalan tanpa digunakan untuk tujuan bisnis.

Menjalankan migrasi besar Anda

Setelah Anda menetapkan hasil bisnis Anda dan mengkomunikasikan strategi kepada para pemangku kepentingan, Anda dapat beralih ke perencanaan bagaimana Anda mengukir ruang lingkup migrasi besar ke dalam peristiwa atau gelombang migrasi berkelanjutan. Contoh berikut memberikan panduan utama untuk membuat rencana gelombang.

Di bagian ini:

- Rencanakan gelombang migrasi sebelumnya untuk memastikan aliran yang stabil
- Pertahankan implementasi gelombang dan perencanaan gelombang sebagai proses dan tim yang terpisah
- Mulai dari yang kecil untuk hasil yang bagus
- Minimalkan jumlah jendela cutover

- Gagal cepat, terapkan pengalaman, dan iterasi
- Jangan lupa retrospektif

Rencanakan gelombang migrasi sebelumnya untuk memastikan aliran yang stabil

Merencanakan migrasi Anda adalah salah satu fase terpenting dari program ini. Ini berlaku dengan pepatah "jika Anda gagal merencanakan, Anda berencana untuk gagal." Merencanakan gelombang migrasi sebelumnya memungkinkan proyek mengalir dengan cepat karena tim menjadi lebih proaktif terhadap situasi migrasi. Ini membantu skala proyek dengan lebih mudah, dan meningkatkan pengambilan keputusan dan peramalan karena tuntutan proyek meningkat dan menjadi kompleks. Perencanaan ke depan juga meningkatkan kemampuan tim untuk beradaptasi dengan perubahan.

Misalnya, pelanggan layanan keuangan besar sedang mengerjakan program keluar pusat data. Awalnya, pelanggan merencanakan gelombang migrasi secara berurutan, menyelesaikan satu gelombang sebelum mulai merencanakan gelombang berikutnya. Pendekatan ini menghasilkan lebih sedikit waktu untuk mempersiapkan. Ketika para pemangku kepentingan diberitahu bahwa aplikasi mereka sedang dimigrasikan AWS, mereka masih memiliki beberapa langkah yang harus dilakukan sebelum memulai migrasi. Ini menambah penundaan yang signifikan pada program. Setelah pelanggan menyadari hal ini, mereka menerapkan aliran perencanaan migrasi yang holistik dan berfokus pada masa depan di mana gelombang migrasi direncanakan beberapa bulan sebelumnya. Ini memberikan pemberitahuan yang cukup bagi tim aplikasi untuk melakukan aktivitas pra-migrasi mereka seperti memberi tahu AWS Mitra, analisis lisensi, dan sebagainya. Mereka kemudian dapat menghapus tugas-tugas itu dari jalur kritis program.

Pertahankan implementasi gelombang dan perencanaan gelombang sebagai proses dan tim yang terpisah

Ketika tim perencanaan gelombang dan implementasi gelombang terpisah, kedua proses dapat bekerja secara paralel. Dengan komunikasi dan koordinasi, ini menghindari perlambatan migrasi karena tidak cukup server atau aplikasi yang siap untuk mencapai kecepatan yang diharapkan. Misalnya, tim migrasi mungkin perlu memigrasi 30 server setiap minggu, tetapi hanya 10 server yang siap dalam gelombang saat ini. Tantangan ini sering disebabkan oleh hal-hal berikut:

- Tim implementasi migrasi tidak terlibat dalam perencanaan gelombang, dan data yang dikumpulkan dalam fase perencanaan gelombang tidak lengkap. Tim implementasi migrasi harus mengumpulkan lebih banyak data server sebelum memulai gelombang.

- Implementasi migrasi dijadwalkan dimulai tepat setelah perencanaan gelombang, tanpa buffer di antaranya.

Sangat penting untuk merencanakan gelombang sebelumnya, dan untuk membuat penyangga antara persiapan dan dimulainya implementasi gelombang. Penting juga untuk memastikan bahwa tim perencanaan gelombang dan tim migrasi bekerja sama untuk mengumpulkan data yang tepat dan menghindari pengerjaan ulang.

Mulai dari yang kecil untuk hasil yang bagus

Rencanakan untuk memulai dari yang kecil dan meningkatkan kecepatan migrasi dengan setiap gelombang berikutnya. Gelombang awal harus berupa aplikasi tunggal, kecil, kurang dari 10 server. Tambahkan aplikasi dan server tambahan di gelombang berikutnya, bangun hingga kecepatan migrasi penuh Anda. Memprioritaskan aplikasi yang kurang kompleks atau berisiko, dan meningkatkan kecepatan pada jadwal, memberi tim waktu untuk menyesuaikan diri untuk bekerja sama dan mempelajari prosesnya. Selain itu, tim dapat mengidentifikasi dan menerapkan perbaikan proses dengan setiap gelombang, yang secara substansional dapat meningkatkan kecepatan gelombang selanjutnya.

Satu pelanggan bermigrasi lebih dari 1.300 server dalam setahun. Dengan memulai dengan migrasi pilot dan beberapa gelombang yang lebih kecil, tim migrasi dapat mengidentifikasi berbagai cara untuk meningkatkan migrasi di kemudian hari. Misalnya, mereka mengidentifikasi segmen jaringan pusat data baru sebelumnya. Mereka bekerja dengan tim firewall mereka di awal proses untuk menerapkan aturan firewall yang memungkinkan komunikasi dengan alat migrasi. Ini membantu mencegah penundaan yang tidak perlu dalam gelombang future. Selain itu, tim dapat mengembangkan skrip untuk membantu mengotomatiskan lebih banyak penemuan dan proses pemotongan mereka dengan setiap gelombang. Memulai dari yang kecil membantu tim fokus pada perbaikan proses awal, dan sangat meningkatkan kepercayaan diri mereka.

Minimalkan jumlah jendela cutover

Migrasi massal membutuhkan pendekatan disiplin untuk mendorong skala. Menjadi terlalu fleksibel di beberapa daerah adalah anti-pola untuk migrasi besar. Dengan membatasi jumlah jendela cutover mingguan, waktu yang dihabiskan untuk aktivitas cutover memiliki nilai yang lebih tinggi.

Misalnya, jika jendela cutover terlalu fleksibel, Anda bisa berakhir dengan 20 cutover dengan lima server masing-masing. Sebagai gantinya, Anda dapat memiliki dua cutover dengan masing-masing 50 server. Karena waktu dan upaya untuk setiap cutover serupa, memiliki pemotongan yang lebih

sedikit dan lebih besar mengurangi beban operasional penjadwalan, dan membatasi penundaan yang tidak perlu.

Sebuah perusahaan teknologi besar mencoba untuk bermigrasi dari beberapa pusat data yang disewa sebelum kontrak berakhir. Kehilangan kedaluwarsa akan menghasilkan persyaratan pembaruan jangka pendek yang mahal. Sebelumnya dalam migrasi, tim aplikasi diizinkan untuk mendikte jadwal migrasi hingga menit terakhir, termasuk memilih keluar dari migrasi karena alasan apa pun hanya beberapa hari sebelum pemotongan. Hal ini menyebabkan banyak penundaan pada tahap awal proyek. Seringkali, pelanggan harus bernegosiasi dengan tim aplikasi lain pada menit terakhir untuk mengisi. Pelanggan akhirnya meningkatkan disiplin perencanaan mereka, tetapi kesalahan awal ini menyebabkan stres terus-menerus bagi tim migrasi. Penundaan jadwal keseluruhan mengakibatkan beberapa aplikasi tidak berhasil keluar dari pusat data tepat waktu.

Gagal cepat, terapkan pengalaman, dan iterasi

Setiap migrasi pada awalnya memiliki jebakan. Gagal lebih awal membantu tim belajar, memahami kemacetan, dan menerapkan pelajaran yang dipetik ke gelombang yang lebih besar. Diharapkan bahwa beberapa gelombang pertama dalam migrasi akan lambat karena alasan berikut:

- Anggota tim menyesuaikan satu sama lain dan prosesnya.
- Migrasi besar biasanya melibatkan banyak alat dan orang yang berbeda.
- Butuh waktu untuk mengintegrasikan, menguji, gagal, belajar, dan terus meningkatkan end-to-end proses.

Masalah umum dan diharapkan selama beberapa gelombang pertama. Penting untuk memahami dan mengomunikasikan hal ini kepada seluruh organisasi, karena beberapa tim mungkin tidak suka mencoba hal-hal baru dan gagal. Kegagalan dapat mengecilkan hati tim dan menjadi pemblokir untuk migrasi masa depan. Memastikan semua orang memahami bahwa masalah awal adalah bagian dari pekerjaan dan mendorong semua orang untuk mencoba dan gagal adalah kunci keberhasilan migrasi.

Satu perusahaan berencana untuk bermigrasi lebih dari 10.000 server dalam 24-36 bulan. Untuk mencapai tujuan itu, mereka perlu memigrasi hampir 300 server sebulan. Namun, itu tidak berarti mereka memigrasikan 300 server sejak hari pertama. Gelombang pasangan pertama adalah gelombang belajar sehingga tim dapat memahami bagaimana segala sesuatunya bekerja dan siapa yang memiliki izin untuk melakukan apa. Mereka juga mengidentifikasi integrasi yang akan meningkatkan proses, seperti mengintegrasikan dengan CMDB dan. CyberArk Mereka menggunakan

gelombang pembelajaran untuk gagal, meningkatkan, dan gagal lagi, menyempurnakan proses dan otomatisasi. Setelah 6 bulan, mereka dapat bermigrasi lebih dari 120 server setiap minggu.

Jangan lupa retrospektif

Ini adalah bagian penting dari proses gesit. Di sinilah tim berkomunikasi, menyesuaikan, belajar, setuju, dan bergerak maju. Retrospektif pada tingkat paling dasar adalah melihat ke belakang, mendiskusikan apa yang terjadi, menentukan apa yang berjalan dengan baik dan apa yang perlu ditingkatkan. Perbaikan kemudian dapat dibangun berdasarkan diskusi tersebut. Retrospektif membungkus beberapa formalitas atau proses di sekitar gagasan pelajaran yang dipelajari. Retrospektif penting karena untuk mencapai skala dan kecepatan agar migrasi besar berhasil, proses, alat, dan tim harus terus berkembang dan meningkat. Retrospektif dapat memainkan peran penting dalam hal itu.

Sesi pelajaran tradisional tidak terjadi sampai akhir program, sehingga seringkali pelajaran ini tidak ditinjau pada awal gelombang migrasi berikutnya. Dengan migrasi besar, pelajaran yang dipetik harus diterapkan pada gelombang berikutnya dan harus menjadi bagian penting dari proses perencanaan gelombang.

Untuk satu pelanggan, retrospektif mingguan diadakan untuk mendiskusikan dan mendokumentasikan pelajaran yang dipetik dari pemotongan. Dalam sesi ini, mereka menemukan area di mana ada ruang untuk merampingkan dari sudut pandang proses atau otomatisasi. Hal ini mengakibatkan implementasi jadwal hitung mundur dengan aktivitas tertentu, pemilik, dan skrip otomatisasi untuk meminimalkan tugas manual, termasuk validasi alat pihak ketiga dan instalasi CloudWatch agen Amazon, selama pemotongan.

Di perusahaan teknologi besar lainnya, retrospektif reguler diadakan dengan tim untuk mengidentifikasi masalah dengan gelombang migrasi sebelumnya. Ini menghasilkan peningkatan proses, skrip, dan otomatisasi yang mendorong waktu migrasi rata-rata turun sebesar 40 persen selama program berlangsung.

Pertimbangan tambahan

Banyak daerah harus diperhitungkan dalam program migrasi besar. Bagian berikut memberikan pemikiran tentang item lain yang harus dipertimbangkan.

Di bagian ini:

- Bersihkan saat Anda pergi
- Menerapkan beberapa fase untuk setiap transformasi tambahan

Bersihkan saat Anda pergi

Migrasi tidak dianggap berhasil jika biayanya 10 kali lipat dari yang Anda harapkan, dan proyek tidak selesai sampai sumber daya yang digunakan untuk migrasi ditutup dan dibersihkan. Pembersihan ini harus menjadi bagian dari kegiatan pasca-migrasi. Ini memastikan bahwa Anda tidak akan meninggalkan sumber daya dan layanan yang tidak terpakai di lingkungan Anda yang akan menambah biaya. Pembersihan pasca-migrasi juga merupakan praktik keamanan yang baik untuk mencegah ancaman dan kerentanan yang mengekspos lingkungan Anda.

Dua hasil utama dari pindah ke AWS Cloud adalah penghematan biaya dan keamanan. Meninggalkan sumber daya yang tidak terpakai dapat mengalahkan tujuan bisnis pindah ke cloud. Sumber daya paling umum yang tidak dibersihkan meliputi:

- Data uji
- Database uji
- Akun uji, termasuk aturan firewall, grup keamanan, dan alamat IP daftar kontrol akses jaringan (ACL jaringan)
- Port disediakan untuk pengujian
- Volume Amazon Elastic Block Store (Amazon EBS)
- Snapshot
- Replikasi (seperti menghentikan replikasi data dari lokal ke lokasi) AWS
- File yang menggunakan ruang (seperti backup database sementara yang digunakan untuk bermigrasi)
- Contoh yang menghosting alat migrasi

Dalam salah satu contoh praktik pembersihan yang buruk, AWS Mitra SI tidak menghapus agen replikasi setelah migrasi berhasil. AWS Audit menemukan bahwa server replikasi dan volume EBS yang telah dimigrasi menelan biaya $20.000 (USD) setiap bulan. Untuk mengurangi masalah ini, Layanan AWS Profesional membuat proses audit otomatis yang memberi tahu AWS Mitra SI ketika server basi masih direplikasi. AWS Mitra SI kemudian dapat mengambil tindakan pada contoh yang tidak digunakan dan basi.

Untuk migrasi masa depan, sebuah proses diadopsi untuk menentukan periode hypercare pasca-migrasi 48 jam untuk memastikan adopsi platform yang lancar. Tim infrastruktur pelanggan kemudian mengajukan permintaan penonaktifan untuk server lokal. Disarankan bahwa setelah persetujuan

permintaan penonaktifan, server dari gelombang masing-masing akan dihapus dari konsol layanan migrasi aplikasi.

Menerapkan beberapa fase untuk setiap transformasi tambahan

Saat melakukan migrasi besar, penting untuk tetap fokus pada tujuan inti Anda, seperti penutupan pusat data atau transformasi infrastruktur. Dalam migrasi yang lebih kecil, scope creep mungkin memiliki dampak minimal. Namun, beberapa hari upaya tambahan dikalikan dengan potensi ribuan server dapat menambah sejumlah besar waktu untuk program. Selain itu, perubahan tambahan mungkin juga memerlukan pembaruan dokumentasi, proses, dan pelatihan untuk tim dukungan.

Untuk mengatasi potensi cakupan creep, Anda dapat menerapkan pendekatan multi-fase untuk migrasi Anda. Misalnya, jika tujuan Anda adalah untuk mengosongkan pusat data, fase 1 mungkin hanya mencakup rehosting beban kerja AWS secepat mungkin. Setelah beban kerja di-rehost, fase 2 dapat mengimplementasikan aktivitas transformasional tanpa mempertaruhkan hasil bisnis target.

Misalnya, satu pelanggan berencana untuk keluar dari pusat data mereka dalam 12 bulan. Namun, migrasi mereka mencakup aktivitas transformasi lainnya, seperti meluncurkan alat pemantauan kinerja aplikasi baru dan meningkatkan sistem operasi. Lebih dari 1.000 server berada dalam lingkup migrasi, sehingga aktivitas ini menambahkan penundaan signifikan pada migrasi. Selanjutnya, pendekatan ini membutuhkan pelatihan dalam penggunaan perkakas baru. Pelanggan kemudian memutuskan untuk menerapkan pendekatan multi-fase dengan fokus awal pada rehost. Ini meningkatkan kecepatan migrasi mereka dan mengurangi risiko tidak memenuhi tanggal penutupan pusat data.

Kesimpulan

Migrasi besar menghadirkan tantangan yang berbeda jika dibandingkan dengan migrasi yang lebih kecil. Ini sebagian besar disebabkan oleh kompleksitas yang diperkenalkan oleh skala. Misalnya, menginstal agen ke satu server cukup mudah dan akan memakan waktu sekitar 5 menit. Namun, jika Anda memiliki 5.000 server dalam ruang lingkup untuk migrasi Anda, ini akan memakan waktu sekitar 416 jam dan akan menghadirkan tantangan berikut:

- Ada kemungkinan bahwa ada beberapa sistem operasi yang membutuhkan proses yang berbeda.
- Mungkin ada domain Microsoft Active Directory terpisah untuk dikelola karena merger dan akuisisi sebelumnya.
- Proses dan alat yang efektif diperlukan untuk mengatur instalasi agen untuk setiap gelombang dan kemudian melacak dan melaporkan kemajuannya.

Strategi ini menguraikan praktik terbaik migrasi besar berdasarkan pengalaman Layanan AWS Profesional yang membantu berbagai pelanggan. Ini termasuk perspektif orang, proses, dan teknologi. Jika Anda ingin memulai atau sedang dalam proses migrasi ke AWS, konsultan di Layanan AWS Profesional akan dengan senang hati membantu Anda. Hubungi AWS perwakilan Anda untuk memulai percakapan.

Untuk langkah selanjutnya, kami sarankan Anda meninjau seri Panduan AWS Preskriptif yang dirancang untuk membantu Anda merencanakan dan menyelesaikan migrasi besar ke. AWS Cloud Untuk seri lengkapnya, lihat Migrasi besar ke. AWS Cloud

Sumber daya

AWS migrasi besar

Untuk mengakses seri Panduan AWS Preskriptif lengkap untuk migrasi besar, lihat Migrasi besar ke. AWS Cloud

Sumber AWS Bimbingan Preskriptif Terkait

- Mengotomatiskan migrasi server skala besar dengan Cloud Migration Factory
- Praktik terbaik untuk menilai aplikasi yang akan pensiun selama migrasi ke AWS Cloud
- Menyiapkan lingkungan AWS multi-akun yang aman dan dapat diskalakan
- Mengevaluasi kesiapan migrasi
- Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar

Referensi tambahan

- AWS Solusi Pabrik Migrasi Cloud
- Layanan migrasi cloud gratis di AWS
- AWS Database Migration Service
- Migrasi dengan AWS

Video

- Menjalankan migrasi skala besar ke AWS (AWS re:invent 2020)
- CloudEndure Praktik terbaik Pabrik Migrasi (AWS Re: Invent 2020)

Kontributor

Strategi ini ditulis oleh tim harimau Migrasi Besar global dalam Layanan AWS Profesional. Tim telah berhasil memigrasikan ribuan server ke AWS atas nama AWS pelanggan. Para kontributor untuk dokumen ini antara lain:

- Chris Baker, Insinyur Produk Utama
- Dwayne Bordelon, Arsitek Aplikasi Cloud Senior
- Rodolfo Jr. Cerrada, Arsitek Aplikasi Senior
- Pratik Chunawala, Arsitek Awan Utama
- Bill David, Manajer Solusi Pelanggan Utama
- Dev Kar, Konsultan Senior
- Wally Lu, Konsultan Utama
- Jon Madison, Arsitek Awan Utama
- Abhishek Naik, Arsitek Solusi Senior
- Damien Renner, Spesialis Migrasi Senior
- Amit Rudraraju, Arsitek Cloud Senior

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
Layanan CloudEndure Migrasi Dihapus	Kami menghapus referensi ke layanan CloudEndure Migrasi. AWS Application Migration Service adalah layanan migrasi utama yang direkomendasikan untuk lift-and-shift migrasi ke. AWS Cloud	Mei 11, 2022
Nama AWS solusi yang diperbarui	Kami memperbarui nama AWS solusi yang direferen sikan dari Pabrik CloudEndu re Migrasi ke Pabrik Migrasi Cloud.	2 Mei 2022
Sumber daya yang diperbarui	Kami memperbarui bagian Pendahuluan dan Sumber Daya dengan dokumen terbaru dalam seri migrasi besar.	8 Maret 2022
Publikasi awal	—	September 16, 2021

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: Migrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di. AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

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

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

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

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau menyebabkan kerugian bagi individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

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

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat database manajemen konfigurasi.

repositori kode

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

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

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu

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

CV

Lihat visi komputer.

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat bahasa definisi database.

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi basis data.

desain berbasis domain

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

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

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

pertukaran data elektronik (EDI)

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

enkripsi

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

kunci enkripsi

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

endianness

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

titik akhir

Lihat titik akhir layanan.

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

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

epik

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

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

Menyediakan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. 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 CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

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

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

data penahanan

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

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

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

aplikasi idle

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

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

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

cabang utama

Lihat cabang.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

mekanisme

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

akun anggota

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

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

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

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat pembelajaran mesin.

modernisasi

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

penilaian kesiapan modernisasi

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

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

aplikasi monolitik (monolit)

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

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

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

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

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

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

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

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

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

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakan berbasis sumber daya), atau menentukan izin maksimum untuk semua akun 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.

penilaian portofolio

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

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona 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 pengidentifikasi pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat Retrieval Augmented Generation.

ransomware

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

Matriks RASCI

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

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

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

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat 7 Rs.

Region

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

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). 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 Konsol Manajemen AWS atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

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

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

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

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

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

SPOF

Lihat satu titik kegagalan.

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai 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 terkini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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