Membentuk ulang aplikasi mainframe Anda dengan menggunakan IBM Db2 bersama untuk database z/OS untuk migrasi bertahap

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Membentuk ulang aplikasi mainframe Anda dengan menggunakan IBM Db2 bersama untuk database z/OS untuk migrasi bertahap

Table of Contents

Membentuk ulang aplikasi mainframe Anda dengan menggunakan IBM Db2 bersama untuk database untuk migrasi bertahap z/OS

Luis Gustavo Dantas dan Andre Botura, Amazon Web Services ()AWS

Mei 2025 (riwayat dokumen)

Dalam lanskap teknologi perusahaan yang terus berubah, modernisasi mainframe telah menjadi persyaratan penting bagi organisasi yang perlu tetap kompetitif dan gesit. Transformasi ini bukan hanya tentang mengganti sistem lama dengan yang baru; ini adalah evolusi strategis yang menjembatani kesenjangan antara fondasi masa lalu yang kuat dan andal dan kemungkinan masa depan yang dinamis dan inovatif.

Mainframe, yang dulunya merupakan pemimpin komputasi perusahaan yang tak terbantahkan, sekarang berada pada titik balik. Kekuatan pemrosesan dan fitur keamanannya yang tiada tara telah membuatnya tetap relevan selama beberapa dekade, tetapi bisnis saat ini membutuhkan sistem yang dapat berintegrasi dengan mulus dengan layanan cloud, mendukung aplikasi seluler, dan memanfaatkan kekuatan kecerdasan buatan dan analitik data besar.

Modernisasi tidak selalu memerlukan migrasi lengkap dari mainframe. Beberapa organisasi memilih pendekatan hybrid yang memanfaatkan kekuatan lingkungan mainframe dan cloud. Strategi ini memungkinkan mereka untuk mempertahankan aplikasi warisan kritis sementara mereka secara bertahap beralih ke platform yang lebih modern. Transisi teknologi ini melibatkan lebih dari sekedar pembaruan sistem; itu membutuhkan transformasi budaya dan keterampilan organisasi. Saat perusahaan memodernisasi, mereka berinvestasi dalam teknologi baru dan tenaga kerja mereka dengan menjembatani kesenjangan generasi dan mempromosikan pembelajaran dan inovasi yang berkelanjutan.

Panduan ini membahas strategi migrasi bertahap yang menyeimbangkan manfaat sistem mainframe dengan keunggulan teknologi cloud modern. Pendekatan replatforming bertahap ini pertama-tama memigrasikan lapisan aplikasi, sambil mempertahankan konektivitas ke IBM Db2 Anda yang ada untuk z/OS database untuk merampingkan proses transisi dan meminimalkan gangguan pada operasi bisnis penting Anda saat Anda mengadopsi kemampuan baru di cloud.

Panduan ini dirancang untuk pembuat keputusan teknis dan tim implementasi yang terlibat dalam inisiatif modernisasi mainframe. Audiens utama termasuk arsitek perusahaan dan solusi, manajer

proyek teknis, dan pemimpin program modernisasi yang perlu memahami aspek strategis dan teknis dari replatforming mainframe. Konten ini sama-sama berharga untuk tim implementasi, termasuk pengembang aplikasi mainframe, AWS atau insinyur cloud, administrator database, dan DevOps insinyur, yang bertanggung jawab untuk melaksanakan implementasi modernisasi.

Hasil bisnis

Perusahaan memiliki banyak alasan kuat untuk memperbarui aplikasi lama mereka. Proses ini menciptakan rasa urgensi di seluruh industri. Ketika para ahli yang lebih tua pensiun, mereka meninggalkan kesenjangan pengetahuan yang signifikan, yang membuatnya penting untuk memodernisasi sistem sebelum keahlian ini hilang. Selain itu, perusahaan didorong oleh kebutuhan untuk mengurangi biaya, meningkatkan kelincahan, dan merespons dengan cepat kondisi pasar yang berubah dengan cepat.

Dorongan untuk transformasi digital semakin diintensifkan oleh teknologi yang muncul dan permintaan akan pengalaman pelanggan yang ditingkatkan. Faktor-faktor ini, dikombinasikan dengan risiko yang terkait dengan pemeliharaan sistem yang kompleks, mendorong organisasi untuk bertindak cepat dalam memodernisasi infrastruktur TI mereka.

Modernisasi mainframe, khususnya, menghadirkan tindakan penyeimbangan yang rumit. Perusahaan harus menjaga stabilitas dan keamanan yang dikenal mainframe, sementara mereka merangkul fleksibilitas dan skalabilitas yang ditawarkan oleh arsitektur modern. Proses ini melibatkan keputusan kompleks tentang aplikasi mana yang akan dimigrasi, mana yang akan ditulis ulang, dan mana yang harus dipertahankan pada mainframe.

Pendorong utama untuk modernisasi termasuk kelincahan dan pengurangan biaya:

- Kelincahan dan waktu untuk memasarkan. Sistem modern memungkinkan proses pengadaan yang lebih cepat dan respons yang lebih cepat terhadap perubahan permintaan pasar. Adopsi DevOps dan SysOps praktik dapat secara signifikan meningkatkan produktivitas dan kecepatan penyebaran.
- Pengurangan biaya. Modernisasi sering mengarah pada pengurangan biaya infrastruktur melalui:
  - Pay-as-you-go model, yang menyelaraskan biaya dengan penggunaan aktual.
  - Mengurangi biaya lisensi yang terkait dengan sistem warisan.
  - Peningkatan elastisitas, yang menyediakan alokasi sumber daya yang lebih baik.
  - Aktif-aktif, pengaturan ketersediaan tinggi, yang meningkatkan ketahanan sistem sambil mengoptimalkan pemanfaatan sumber daya.

Berdasarkan penggerak bisnis ini, replatforming aplikasi COBOL dianggap sebagai pendekatan strategis untuk modernisasi. Anda dapat menggunakan database bersama untuk mengikuti jalur migrasi bertahap yang menyeimbangkan kebutuhan modernisasi dengan keharusan untuk menjaga kelangsungan bisnis. Metode ini memungkinkan Anda memanfaatkan manfaat arsitektur modern sambil menjaga keandalan aplikasi COBOL Anda. Hasilnya, Anda dapat mencapai kelincahan, efisiensi biaya, dan inovasi sambil mengurangi risiko yang terkait dengan transisi skala besar dan tiba-tiba. Pendekatan database Db2 bersama yang dijelaskan dalam panduan ini menyediakan jembatan antara sistem lama dan platform modern, dan memungkinkan proses modernisasi yang lebih lancar dan lebih terkontrol.

Dalam panduan ini:

AWS Mainframe Modernization

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

AWS Mainframe Modernization Layanan ini memungkinkan Anda untuk memigrasikan aplikasi mainframe lama Anda ke lingkungan cloud-native, mempertahankan logika bisnis dan investasi yang ada, menggunakan alat otomatis dan layanan runtime terkelola, mengoptimalkan kinerja aplikasi, dan mengurangi biaya operasional. Layanan ini merampingkan proses modernisasi, sehingga Anda dapat memanfaatkan kekuatan cloud sambil mempertahankan nilai sistem mainframe inti Anda. AWS menyediakan dua pendekatan utama untuk modernisasi mainframe: replatforming dan refactoring otomatis.

Pembuatan ulang

AWS Mainframe Modernization Replatform dengan Rocket Software (sebelumnya Micro Focus) menyediakan opsi replatforming yang kuat untuk bisnis yang ingin memigrasikan aplikasi mainframe mereka ke cloud dengan gangguan minimal. Solusi ini memungkinkan Anda untuk mengkompilasi ulang dan menjalankan COBOL dan PL/I aplikasi yang ada AWS tanpa memerlukan perubahan kode yang signifikan.

Manfaat utama dari solusi AWS Replatform with Rocket Software meliputi:

- Pelestarian logika bisnis dan investasi yang ada
- Mengurangi risiko dan waktu yang lebih cepat ke pasar
- Peningkatan skalabilitas dan kinerja infrastruktur AWS
- Akses ke alat dan praktik pengembangan modern

Anda dapat menggunakan solusi ini untuk mempertahankan bahasa pemrograman mainframe Anda yang sudah dikenal sambil memanfaatkan fleksibilitas, efektivitas biaya, dan inovasi. AWS Cloud

Pemfaktoran ulang otomatis

Untuk pendekatan yang lebih transformatif daripada replatforming, Anda dapat menggunakan AWS Blu Age, yang menawarkan refactoring otomatis aplikasi mainframe ke aplikasi cloud-native berbasis Java. Solusi ini membantu Anda memodernisasi sistem warisan Anda secara lebih komprehensif, dan mengubahnya menjadi aplikasi yang dapat memanfaatkan sepenuhnya teknologi cloud-native.

Keuntungan utama dari AWS Blu Age meliputi:

- Konversi kode lama ke aplikasi Java modern yang dapat dipelihara
- Transformasi otomatis yang mengurangi upaya manual dan potensi kesalahan
- Pembuatan aplikasi cloud-native yang dioptimalkan untuk Layanan AWS
- Peningkatan kelincahan dan integrasi yang lebih mudah dengan teknologi modern

AWS Blu Age membantu Anda memigrasikan aplikasi Anda dan mempersiapkannya untuk cloud, untuk membuka kemungkinan baru untuk inovasi dan pertumbuhan. Untuk informasi lebih lanjut tentang pendekatan ini, lihat Refactoring aplikasi secara otomatis dengan AWS Blu Age dalam dokumentasi. AWS Mainframe Modernization

Manfaat replatforming

Panduan ini membahas pendekatan untuk mereplatforming aplikasi COBOL mainframe pada. AWS Pendekatan ini bertujuan untuk memodernisasi sistem warisan sementara mempertahankan IBM Db2 untuk merampingkan proses transisi. z/OS Dengan mempertahankan struktur database yang ada pada awalnya, Anda dapat mengurangi kompleksitas dan risiko selama migrasi. Pendekatan bertahap ini membantu Anda mendapatkan keuntungan dari skalabilitas dan efektivitas biaya AWS Cloud sambil menjaga integritas data penting. Keuntungan dari replatforming bertahap meliputi:

- Modernisasi yang dipercepat: Replatforming dan refactoring biasanya membutuhkan lebih sedikit waktu dan sumber daya dibandingkan dengan membayangkan ulang aplikasi lama di cloud, karena tidak melibatkan penulisan ulang seluruh aplikasi. Pendekatan ini juga mendukung transisi yang lebih bertahap yang memungkinkan organisasi untuk memodernisasi dengan kecepatan mereka sendiri sambil segera mendapatkan manfaat dari skalabilitas dan efektivitas biaya. AWS Cloud

- Mitigasi risiko: Replatforming menawarkan beberapa keuntungan dibandingkan refactoring untuk banyak organisasi. Perusahaan dapat mempertahankan COBOL dan PL/I basis kode yang ada, mempertahankan logika bisnis selama bertahun-tahun, dan meminimalkan risiko yang terkait dengan perubahan kode skala besar.
- Kontinuitas data dan migrasi bertahap: Manfaat signifikan dari replatforming adalah opsi untuk awalnya menyimpan data dalam Db2 dalam format data aslinya. z/OS Strategi ini menghindari kebutuhan akan proses migrasi data yang segera, kompleks, dan berpotensi berisiko. Dengan mempertahankan data di lingkungan aslinya selama fase awal, Anda dapat menjaga integritas data, mengurangi waktu henti, dan meminimalkan risiko kehilangan data atau korupsi selama proses modernisasi. Sebagai langkah kedua, Anda dapat merencanakan migrasi data bertahap yang terkontrol ke database cloud-native yang melibatkan pengujian dan validasi menyeluruh saat aplikasi terus berjalan di lingkungan yang direplatformed.
- Fleksibilitas dan pemeriksaan masa depan: Bagi perusahaan yang memiliki investasi signifikan dalam keterampilan dan aplikasi mainframe, replatforming memberikan jalur pragmatis menuju modernisasi yang menyeimbangkan inovasi dengan kontinuitas. Ini menawarkan fleksibilitas untuk mempertahankan struktur data penting dan metode akses pada awalnya, sementara juga menetapkan tahap untuk upaya modernisasi masa depan, termasuk migrasi data akhirnya ke solusi cloud-native sepenuhnya.

Organizations dapat mengikuti pendekatan replatforming untuk memodernisasi dengan kecepatan mereka sendiri dan memenuhi kebutuhan mendesak sambil merencanakan tujuan transformasi digital jangka panjang. Pendekatan ini juga memberi perusahaan kesempatan untuk melatih staf mereka pada layanan cloud-native.

Proses transformasi

Modernisasi mainframe adalah langkah penting bagi organisasi yang ingin memanfaatkan manfaat komputasi awan sambil melestarikan aplikasi warisan mereka yang berharga. Transformasi ini menghadirkan tantangan yang signifikan. Aplikasi mainframe biasanya sangat digabungkan dan memiliki saling ketergantungan yang rumit yang telah berkembang selama beberapa dekade operasi. Kompleksitas ini memerlukan pendekatan modernisasi yang cermat dan metodis.

Organizations perlu menavigasi fase-fase kunci berikut untuk transisi yang sukses:

- Perencanaan: Fase ini melibatkan penemuan komprehensif sistem yang ada dan memprioritaskan upaya modernisasi. Organizations menilai infrastruktur mereka saat ini, mengidentifikasi aplikasi penting, dan menentukan sistem mana yang perlu dimodernisasi terlebih dahulu.
- Membangun: Selama tahap ini, organisasi membuat proses untuk memigrasi aplikasi dan mengembangkan sistem dan infrastruktur baru. Ini melibatkan merancang dan mengimplementasikan arsitektur modern dan menyusun kode sumber.
- Running: Langkah ini terdiri dari membuat lingkungan runtime untuk meng-host aplikasi yang direplatformed. Ini melibatkan pengaturan perangkat keras, perangkat lunak, dan infrastruktur cloud yang diperlukan untuk mendukung sistem modern dan untuk memastikan bahwa mereka dapat beroperasi secara efisien di lingkungan baru.
- Pengujian: Fase ini mencakup validasi ketat dari sistem modern untuk memverifikasi bahwa semua persyaratan fungsional dan kinerja telah dipenuhi. Pengujian ekstensif dilakukan untuk memverifikasi integritas data, kompatibilitas sistem, dan kinerja keseluruhan lingkungan baru.
- Cutover: Fase terakhir berfokus pada penerapan strategi untuk transisi yang mulus dan mengendalikan pergeseran dari mainframe lama ke lingkungan modern. Ini termasuk perencanaan yang cermat dari jadwal migrasi dan rencana darurat untuk meminimalkan gangguan pada operasi bisnis.

Bagian berikut membahas fase-fase ini secara rinci:

- Perencanaan
- Membangun
- Berlari
- Pengujian
- Potongan

Perencanaan

Untuk menavigasi persyaratan aplikasi warisan mainframe secara efektif, organisasi sering memulai dengan penilaian komprehensif lingkungan mainframe mereka.

Penemuan aplikasi

Alat yang ampuh dalam fase awal ini adalah Rocket Enterprise Analyzer, yang memberikan wawasan mendalam tentang struktur, dependensi, dan kompleksitas aplikasi mainframe. Alat ini membantu Anda menentukan ruang lingkup upaya modernisasi Anda, potensi risiko, dan peluang untuk pengoptimalan.

Salah satu aspek penting untuk diungkap adalah jaringan ketergantungan data yang rumit dalam sistem mainframe. Dependensi ini sering tersembunyi di bawah lapisan kode lama dan dapat secara signifikan memengaruhi upaya modernisasi. Dengan memetakan bagaimana aplikasi dan modul yang berbeda berinteraksi dengan berbagai sumber data, Anda dapat lebih memahami efek potensial dari setiap perubahan yang Anda rencanakan untuk diterapkan.

Ketergantungan data

Penilaian menyeluruh terhadap dependensi data dapat mengungkapkan informasi penting tentang aliran data, kualitas data, dan tata kelola data dalam lingkungan mainframe Anda. Pengetahuan ini sangat berharga ketika merencanakan strategi migrasi data, memastikan integritas data selama modernisasi, dan mengidentifikasi peluang untuk optimasi data. Dengan mendapatkan gambaran yang jelas tentang data Anda, Anda dapat membuat keputusan yang lebih tepat tentang pendekatan modernisasi mana yang paling efektif dan paling tidak mengganggu operasi Anda yang ada.

Analisis top-down yang mengidentifikasi penggunaan tabel berdasarkan transaksi atau pekerjaan job control language (JCL) adalah kunci untuk menciptakan perencanaan dan prioritas gelombang. Pendekatan ini menjelaskan hubungan antara berbagai komponen sistem mainframe Anda, dan membantu Anda mengembangkan pendekatan strategis dan bertahap untuk modernisasi. Dengan mengidentifikasi tabel mana yang paling sering diakses dan proses mana, Anda dapat memprioritaskan upaya modernisasi Anda: Anda dapat fokus pada area berdampak tinggi terlebih dahulu dan memastikan transisi yang lebih lancar dengan gangguan minimal terhadap operasi bisnis yang kritis.

Selain menggunakan Rocket Enterprise Analyzer untuk menemukan dependensi data, banyak organisasi juga menggunakan solusi custom-built mereka sendiri untuk mendapatkan wawasan

yang lebih dalam tentang lingkungan mainframe mereka. Alat in-house ini sering mengeksploitasi kekayaan informasi yang tersedia dalam katalog IBM Db2 dan catatan Fasilitas Manajemen Sistem (SMF).

Tolok ukur kapasitas

Salah satu langkah dalam merencanakan proyek replatforming mainframe Anda adalah mengumpulkan informasi terperinci tentang konsumsi beban kerja Anda saat ini. Data ini akan membantu Anda memprediksi secara akurat dan menyediakan kapasitas awal yang diperlukan di lingkungan cloud target Anda. Misalnya, kami menyarankan Anda mengumpulkan data konsumsi per jam juta instruksi per detik (MIPS) untuk transaksi online dan transaksi batch dari IBM Customer Information Control System (CICS) atau Information Management System (IMS) dan job control language (JCL).

IBM menawarkan beragam model harga untuk MIPS dalam komputasi mainframe, dan banyak dari model ini berpusat di sekitar penggunaan puncak. Di antara model berbasis puncak ini, yang paling umum adalah puncak empat jam bergulir.

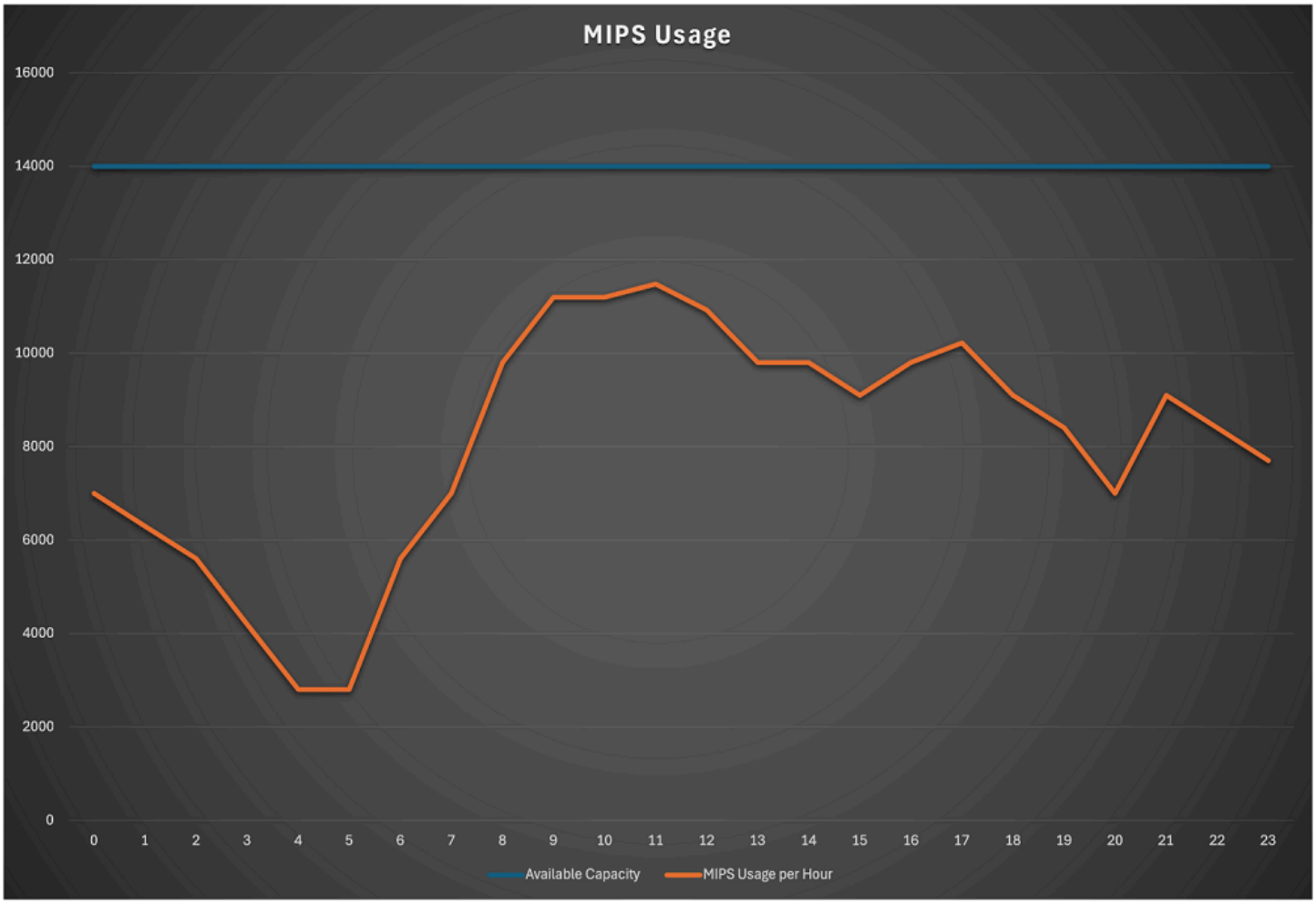
Biaya mainframe mencakup lima bidang utama yang secara signifikan memengaruhi pengeluaran keseluruhan:

- Lisensi perangkat lunak sering menjadi komponen utama. Ini mencakup sistem operasi, middleware, database, dan berbagai aplikasi, dan biaya kadang-kadang terkait dengan kapasitas atau penggunaan mesin.
- Biaya perangkat keras termasuk pembelian awal atau sewa peralatan mainframe, pemeliharaan berkelanjutan, dan peningkatan.
- Biaya penyimpanan dapat menjadi besar karena sejumlah besar data terkelola yang dikelola, dan melibatkan sistem disk, pustaka pita, dan perangkat lunak manajemen terkait.
- Biaya personel mencakup gaji untuk profesional mainframe khusus seperti programmer sistem dan administrator database.
- Pemulihan bencana dan langkah-langkah kelangsungan bisnis, termasuk sistem cadangan, perangkat keras yang berlebihan, dan fasilitas pemulihan di luar lokasi, merupakan investasi yang signifikan dalam memastikan ketersediaan tinggi dan pemulihan yang cepat.

Kelima kategori biaya ini, dikombinasikan dengan biaya berbasis MIPS, membentuk inti dari sebagian besar anggaran mainframe. Namun, proporsi relatifnya dapat sangat bervariasi tergantung pada ukuran organisasi Anda, industri, dan pola pemanfaatan mainframe tertentu.

Data MIPS per jam sangat penting untuk mendapatkan pemahaman komprehensif tentang pola dan kinerja beban kerja mainframe Anda. Tidak seperti rata-rata harian atau bulanan, data per jam memberikan wawasan terperinci yang mengungkapkan fluktuasi bernuansa dalam pemanfaatan sumber daya sistem Anda sepanjang hari. Tingkat detail ini sangat berharga untuk menilai kinerja dan kebutuhan kapasitas aplikasi Anda secara akurat di cloud.

Dengan menganalisis data MIPS per jam, Anda dapat mengidentifikasi periode penggunaan puncak, tren spot, dan menentukan potensi kemacetan yang mungkin dikaburkan dalam data agregat, seperti yang ditunjukkan pada diagram berikut. Perincian ini memungkinkan perencanaan kapasitas yang lebih tepat, membantu mengoptimalkan alokasi sumber daya, dan berpotensi mengarah pada penghematan biaya dan peningkatan efisiensi sistem.



Data MIPS per jam juga berfungsi sebagai alat tolok ukur kinerja yang penting. Ini menetapkan dasar rinci kinerja sistem Anda, yang sangat berharga ketika Anda merencanakan atau mengevaluasi perubahan sistem seperti migrasi atau peningkatan. Dengan membandingkan data MIPS per jam pra-perubahan dan pasca-perubahan, Anda dapat secara akurat mengukur dampak modifikasi ini pada

kinerja sistem Anda dan memastikan bahwa mainframe Anda terus memenuhi kebutuhan organisasi Anda.

Untuk mengumpulkan data MIPS per jam, Anda memiliki beberapa opsi. Salah satu pendekatannya adalah dengan menggunakan catatan SMF secara langsung. Catatan-catatan ini memberikan banyak informasi tentang aktivitas sistem dan penggunaan sumber daya. Atau, Anda dapat menggunakan alat khusus seperti IBM Sub-Capacity Reporting Tool (SCRT), yang dapat menyederhanakan proses pengumpulan dan analisis data MIPS.

Terlepas dari metode yang Anda pilih, penting untuk mengumpulkan data selama periode yang diperpanjang—idealnya, beberapa bulan. Periode pengumpulan yang diperpanjang ini memungkinkan Anda memperhitungkan variasi siklus dalam beban kerja Anda, seperti lonjakan end-of-month pemrosesan atau fluktuasi musiman. Dengan menangkap pola jangka panjang ini, Anda dapat mengembangkan gambaran yang lebih akurat dan komprehensif tentang karakteristik kinerja mainframe Anda, yang memungkinkan pengambilan keputusan yang lebih baik dan manajemen kapasitas yang lebih efektif.

Perencanaan gelombang

Anda dapat menggunakan informasi yang Anda kumpulkan untuk memprioritaskan inisiatif replatforming mainframe Anda secara strategis. Pendekatan yang bijaksana adalah memulai dengan beban kerja yang kurang kritis, seperti transaksi bisnis non-inti atau pekerjaan batch, untuk memungkinkan tim memperoleh pengalaman dan menyempurnakan proses dengan risiko minimal terhadap operasi penting. Selain itu, mempertimbangkan beban kerja hanya-baca sebagai kandidat awal untuk migrasi dapat menguntungkan, karena beban kerja ini biasanya melibatkan lebih sedikit kompleksitas dan risiko inkonsistensi data yang lebih rendah. Pendekatan ini memungkinkan Anda untuk membangun kepercayaan diri dan momentum dalam upaya replatforming Anda.

Selain itu, pengelompokan beban kerja yang berbagi tabel Db2 untuk operasi tulis atau pembaruan dapat merampingkan proses migrasi. Dengan mengidentifikasi beban kerja yang saling berhubungan ini, Anda dapat merencanakan gelombang migrasi kohesif yang menjaga integritas data dan meminimalkan kebutuhan akan solusi sementara yang kompleks. Strategi ini tidak hanya mengurangi risiko konflik data tetapi juga mengoptimalkan garis waktu replatforming secara keseluruhan dengan menangani komponen terkait secara bersamaan. Pada akhirnya, pendekatan prioritas berbasis data ini memastikan pertimbangan yang seimbang tentang kekritisan, kompleksitas, dan saling ketergantungan, dan mengarah pada proses modernisasi mainframe yang lebih efisien dan sukses.

Membangun

Menggunakan database Db2 bersama memungkinkan eksekusi bersamaan dari aplikasi identik atau konsisten di lingkungan mainframe dan cloud. Pendekatan ini menawarkan beberapa keuntungan ketika Anda mempertahankan versi aplikasi yang sama di kedua platform, dan memberikan peningkatan fleksibilitas dan keandalan dalam operasi Anda.

Salah satu keuntungan utama dari strategi ini adalah kemampuan untuk menerapkan rencana rollback yang efektif. Jika masalah muncul selama migrasi atau penerapan, memiliki versi aplikasi yang sama memungkinkan pengembalian yang mulus ke status sebelumnya, dan meminimalkan waktu henti dan potensi inkonsistensi data.

Konsistensi aplikasi

Mencerminkan komponen aplikasi dari manajer kontrol sumber terdistribusi ke mainframe adalah pendekatan strategis selama proses replatforming. Metode ini mendukung penggunaan alat manajemen kode sumber modern sambil mempertahankan sinkronisasi dengan lingkungan mainframe. Proses mirroring ini bersifat sementara, dan hanya berlangsung sampai beban kerja berfungsi penuh dalam produksi pada platform terdistribusi.

Dengan memigrasikan kode sumber aplikasi yang di-replatformed ke alat manajemen perubahan terdistribusi, Anda dapat memanfaatkan beberapa manfaat yang ditawarkan oleh pengelola kode sumber modern. Ini termasuk:

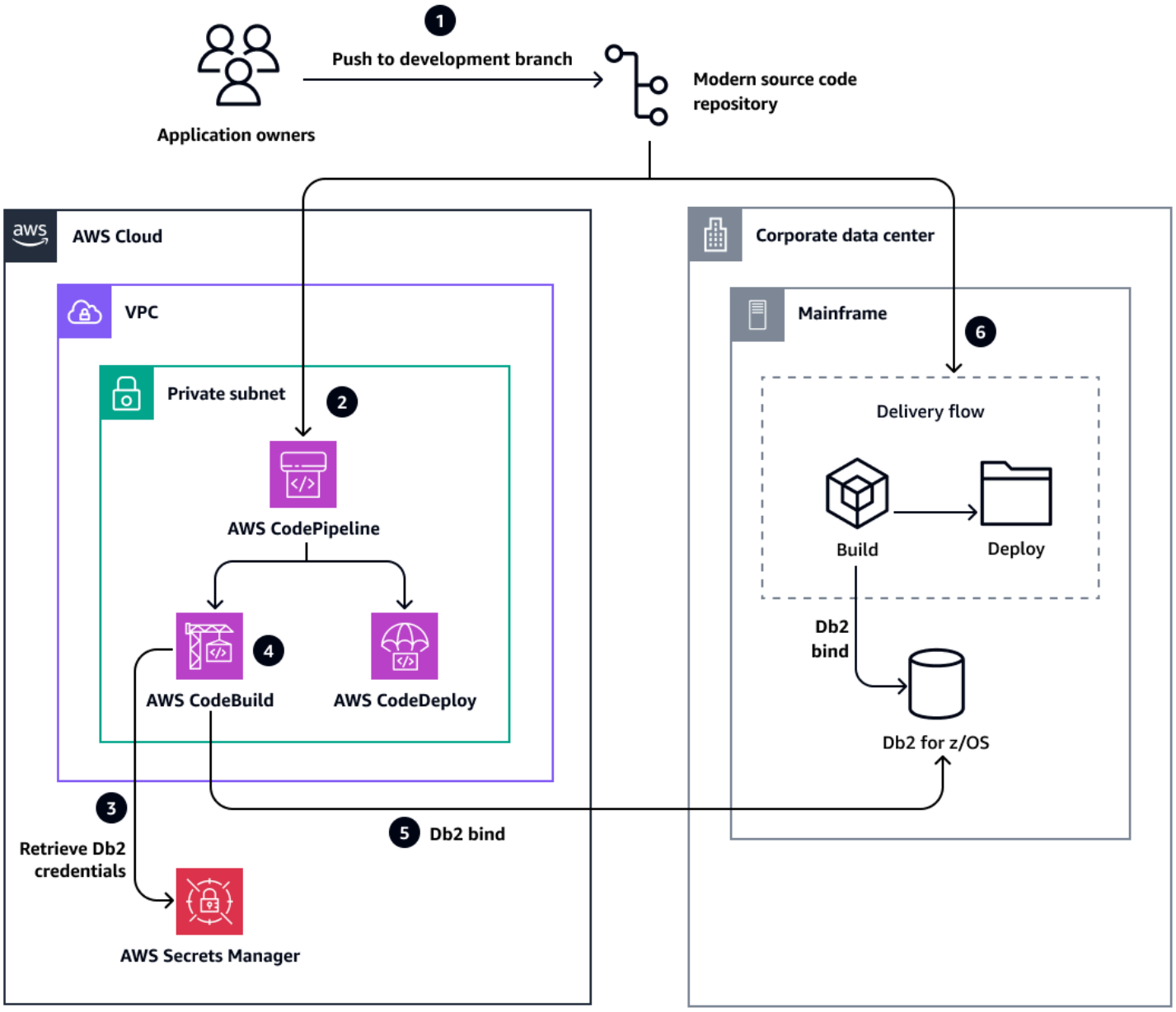
- Kolaborasi yang disempurnakan: Alat terdistribusi sering memberikan dukungan yang lebih baik untuk kolaborasi tim dengan menyertakan fitur seperti permintaan tarik, tinjauan kode, dan strategi percabangan.
- Kontrol versi yang ditingkatkan: Sistem modern menawarkan kontrol versi yang lebih terperinci, dan membuatnya lebih mudah untuk melacak perubahan dan mengelola berbagai versi kode.
- Integrasi dengan CI/CD jaringan pipa: Banyak alat terdistribusi yang terintegrasi secara mulus dengan pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD), yang merampingkan proses pengembangan.
- Visibilitas dan keterlacakan yang lebih baik: Alat-alat ini sering memberikan dasbor dan kemampuan pelaporan yang unggul, dan menawarkan wawasan yang lebih besar tentang proses pengembangan.
- Support for Modern Development Practices: Sistem terdistribusi biasanya lebih cocok untuk metodologi dan DevOps praktik agile.

Proses mirroring melibatkan sinkronisasi kode dari manajer kontrol sumber terdistribusi kembali ke mainframe. Ini memastikan bahwa kedua lingkungan tetap konsisten selama periode transisi. Namun, Anda harus menerapkan mirroring sebagai sinkronisasi satu arah, di mana pembaruan mengalir dari sistem terdistribusi ke mainframe, bukan dua arah. Pendekatan ini menjaga konsistensi dan mencegah potensi konflik yang dapat timbul dari pembaruan simultan di kedua lingkungan.

Dengan mengadopsi strategi mirroring ini, Anda dapat secara bertahap mengalihkan upaya pengembangan Anda ke platform terdistribusi sambil memastikan bahwa lingkungan mainframe tetap ada. up-to-date Ini memberikan transisi yang lebih mulus dan jaring pengaman selama proses replatforming. Ketika beban kerja berfungsi penuh dan stabil di lingkungan produksi terdistribusi, Anda dapat menghentikan proses pencerminan secara bertahap dan menyelesaikan migrasi ke sistem manajemen kode sumber modern.

Arsitektur

Diagram berikut menunjukkan bagaimana sistem manajemen kode sumber terdistribusi dapat mencerminkan komponen aplikasi dan mempertahankan sinkronisasi antara lingkungan AWS Cloud dan mainframe. AWS Cloud Lingkungan menggunakan CI/CD layanan seperti AWS CodeBuild, AWS CodePipeline, dan AWS CodeDeployuntuk membangun dan menyebarkan aplikasi.



Dalam alur kerja ini:

1. Pemilik aplikasi mengirimkan rilis aplikasi baru ke cabang pengembangan repositori kode sumber.
2. Pemicu AWS CodePipeline rilis baru.
3. AWS CodeBuild mengambil kredensi Db2 dari. AWS Secrets Manager
4. CodeBuild mengkompilasi aplikasi.
5. CodeBuild menggunakan Db2 untuk z/OS untuk mengikat aplikasi.
6. Alur pengiriman mainframe membangun dan menyebarkan aplikasi juga.

Berjalan

Untuk memastikan performa optimal dan latensi rendah antara aplikasi berbasis Internet dan database lokal, sebaiknya Anda menerapkannya. AWS Direct Connect Layanan ini menyediakan koneksi jaringan khusus antara AWS dan pusat data organisasi Anda, dan menawarkan kinerja jaringan yang lebih konsisten dan mengurangi latensi dibandingkan dengan koneksi berbasis internet. Hal ini sangat penting untuk operasi database yang membutuhkan waktu respon cepat.

Untuk mencapai ketersediaan tinggi (HA) dan elastisitas untuk aplikasi yang sedang berjalan AWS, Anda dapat menerapkan arsitektur yang kuat dengan menggunakan komponen berikut:

- Elastic Load Balancing (ELB): Anda dapat menerapkan penyeimbang beban untuk mendistribusikan lalu lintas masuk di beberapa instans Amazon Elastic Compute Cloud (Amazon EC2) yang dijalankan aplikasi Anda. Ini memastikan distribusi beban kerja yang merata dan menyediakan titik masuk tunggal untuk permintaan klien.
- Grup Auto Scaling: Instans EC2 yang menghosting aplikasi dapat diatur ke dalam grup Auto Scaling. Hal ini memungkinkan infrastruktur untuk secara otomatis menyesuaikan jumlah instance berdasarkan metrik yang telah ditentukan seperti pemanfaatan CPU atau lalu lintas jaringan. Selama waktu puncak, instans tambahan dapat diluncurkan untuk menangani peningkatan beban, sedangkan selama periode yang lebih tenang, instans yang tidak perlu dapat dihentikan untuk mengoptimalkan biaya.
- Instans EC2: Aplikasi dapat digunakan pada instans EC2 dalam grup Auto Scaling. Instans ini harus didistribusikan di beberapa Availability Zone untuk meningkatkan toleransi kesalahan dan memastikan ketersediaan yang tinggi.
- Penerapan multi-AZ: Dengan menyebarkan instance aplikasi di beberapa Availability Zone, sistem dapat menahan kegagalan satu Availability Zone tanpa berdampak signifikan pada ketersediaan secara keseluruhan.

Arsitektur ini memungkinkan aplikasi untuk skala mulus berdasarkan permintaan sambil mempertahankan ketersediaan tinggi. Penyeimbang beban memastikan bahwa lalu lintas didistribusikan secara merata di seluruh instans yang sehat, dan grup Auto Scaling mengelola jumlah instans berdasarkan beban kerja aktual.

Untuk lebih meningkatkan keandalan, Anda dapat menerapkan sistem pemantauan dan peringatan yang kuat dengan menggunakan Amazon CloudWatch untuk membantu mendeteksi dan menanggapi masalah atau kegagalan kinerja apa pun dengan segera. Selain itu, pengujian reguler

terhadap kemampuan penskalaan otomatis dan skenario failover akan memastikan bahwa sistem berperilaku seperti yang diharapkan selama berbagai kondisi beban dan potensi kegagalan.

Dengan mengadopsi pendekatan ini, Anda bisa mendapatkan keuntungan dari skalabilitas dan fleksibilitas AWS Cloud sambil mempertahankan koneksi aman ke database Db2 lokal Anda. Pengaturan hybrid ini berfungsi sebagai jalur yang sangat baik menuju migrasi cloud penuh, dan memberikan transisi bertahap dan mitigasi risiko selama proses berlangsung.

Komit dua fase (2PC)

AWS Mainframe Modernization Replatform dengan Rocket Software menawarkan dukungan untuk transaksi dua fase commit (2PC) melalui implementasi Extended Architecture (XA). Kemampuan ini sangat penting untuk menjaga integritas data di seluruh sistem terdistribusi, terutama di lingkungan mainframe di mana transaksi kompleks sering menjangkau banyak sumber daya.

Arsitektur XA, yang terintegrasi ke dalam AWS Replatform dengan Rocket Software, memungkinkan koordinasi transaksi di berbagai sumber daya seperti database dan antrian pesan. Integrasi ini memastikan bahwa semua bagian dari transaksi terdistribusi baik melakukan komit atau memutar kembali secara serempak, untuk menjaga konsistensi di seluruh sistem.

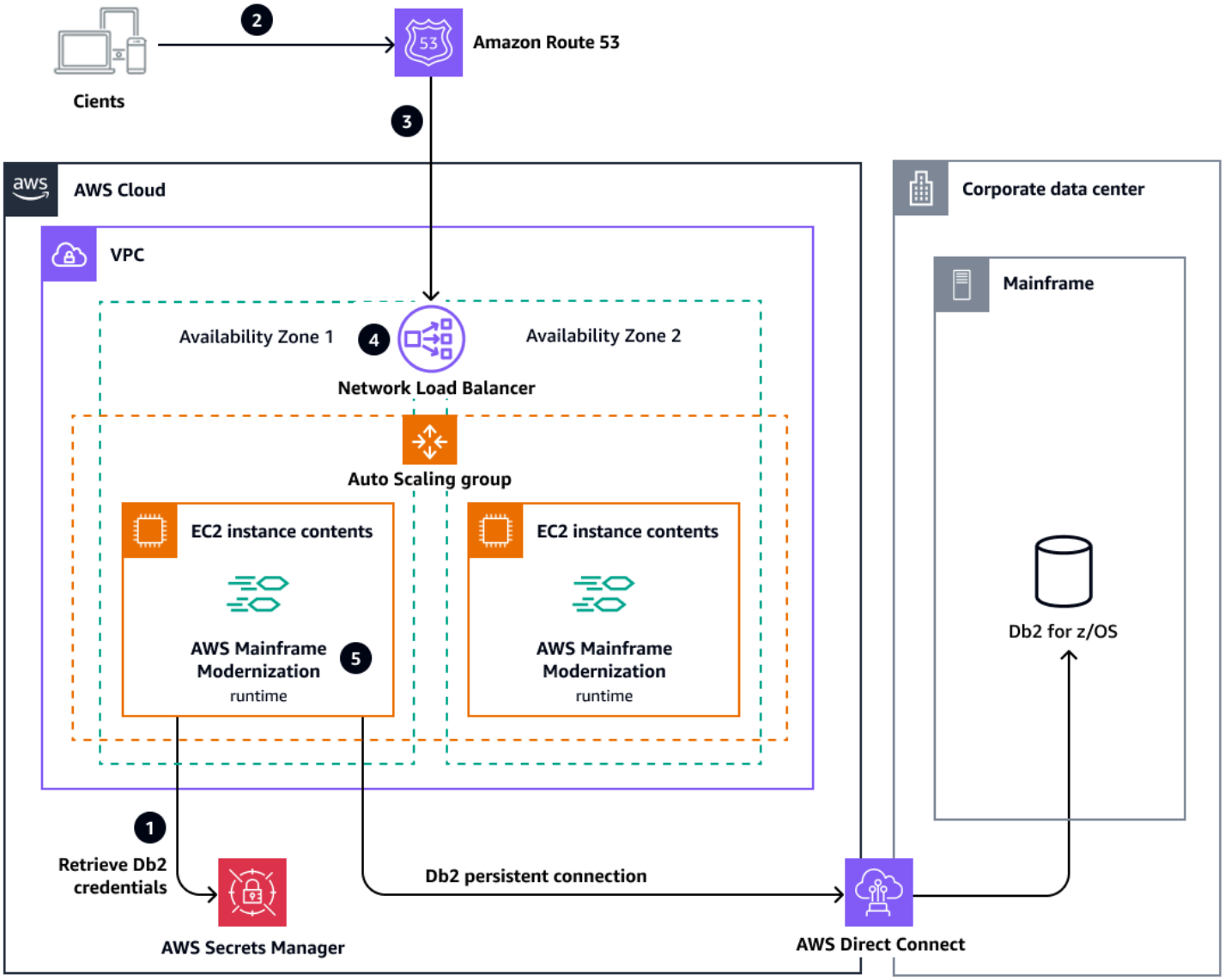
Proses 2PC terdiri dari dua fase:

- Mempersiapkan fase: Manajer transaksi menanyakan semua manajer sumber daya yang terlibat dalam transaksi untuk memastikan bahwa mereka siap untuk berkomitmen.
- Fase komit: Jika semua manajer sumber daya merespons secara positif, manajer transaksi menginstruksikan mereka untuk melakukan perubahan. Jika salah satu manajer sumber daya tidak dapat berkomitmen, semua manajer diinstruksikan untuk mengembalikan perubahan.

Dengan menggunakan XA, AWS Replatform with Rocket Software menyediakan solusi yang andal dan terukur untuk mengelola transaksi yang kompleks dan terdistribusi di lingkungan mainframe modern. Fitur ini sangat penting bagi organisasi yang ingin memigrasikan aplikasi mainframe mereka ke cloud tanpa mengorbankan integritas atau kinerja transaksional.

Infrastruktur runtime

Diagram berikut menunjukkan lingkungan yang sangat tersedia dan elastis dalam AWS Cloud yang mencakup dua Availability Zone, instans EC2 dalam grup Auto Scaling, Network Load Balancer, dan koneksi khusus antara AWS lingkungan mainframe dan melalui. AWS Direct Connect



Dalam arsitektur ini:

1. Ketika AWS Mainframe Modernization runtime dimulai, ia mengambil kredensi Db2 dari AWS Secrets Managerdan membuka koneksi persisten dengan Db2 untuk z/OS.

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan

yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

2. Klien mengikat alamat Network Load Balancer di Amazon Route 53.
3. Route 53 mengalihkan transaksi ke Network Load Balancer.
4. Network Load Balancer mendistribusikan transaksi di beberapa instans EC2.
5. Beban kerja yang berjalan pada AWS Mainframe Modernization berinteraksi dengan Db2 untuk dengan menggunakan koneksi z/OS persisten melalui. AWS Direct Connect

Pengujian

Saat Anda memplatform ulang aplikasi COBOL sambil mempertahankan Db2 z/OS sebagai database bersama, penting untuk memastikan bahwa sistem baru berfungsi setara dengan aslinya. Lingkungan hibrida ini menghadirkan tantangan dan peluang unik untuk pengujian. Strategi berikut menguraikan pendekatan komprehensif untuk pengujian kesetaraan fungsional dan dirancang untuk memvalidasi kinerja aplikasi yang direplatformed, integritas data, dan integrasi tanpa batas dengan database Db2 untuk z/OS yang ada.

Mulailah dengan mengidentifikasi proses bisnis penting dan transaksi yang perlu dibandingkan antar sistem. Kemudian, buat rencana pengujian terperinci dengan skenario spesifik yang secara efektif akan mengevaluasi kesetaraan fungsional dari transaksi ini. Akhirnya, kembangkan kumpulan data pengujian komprehensif yang mencakup semua skenario yang diidentifikasi, dan pastikan keduanya identik untuk kedua sistem untuk memungkinkan perbandingan yang akurat.

Lingkungan sumber

- Cuplikan awal (snapshot pertama):
  - Pastikan tabel data tidak digunakan oleh aplikasi lain selama pengujian, karena ini dapat mempengaruhi uji ekivalensi.
  - Ambil snapshot dari Db2 untuk z/OS tabel yang digunakan oleh transaksi sebelum menjalankan tes apa pun.
- Pengujian sistem sumber:
  - Jalankan rangkaian tes lengkap pada aplikasi COBOL asli.
  - Catat semua transaksi, input, dan output.
  - Memantau kinerja sistem dan pemanfaatan sumber daya.

- Snapshot pengujian pasca-sumber (snapshot kedua):
  - Ambil snapshot lain dari Db2 untuk z/OS database setelah Anda menyelesaikan tes sistem sumber.

Lingkungan target

- Setel ulang basis data:
  - Kembalikan database ke keadaan awal dengan menggunakan snapshot pertama.
- Pengujian sistem target (lingkungan replatformed):
  - Jalankan rangkaian pengujian yang sama pada aplikasi yang direplatformed.
  - Pastikan bahwa semua pengujian sistem target menggunakan input yang sama dengan pengujian sistem sumber.
  - Memantau kinerja sistem dan pemanfaatan sumber daya.
- Snapshot pengujian pasca-target (snapshot ketiga):
  - Ambil snapshot akhir dari Db2 untuk z/OS database setelah Anda menyelesaikan tes sistem target.

Analisis

- Perbandingan dan analisis:
  - Bandingkan snapshot kedua dan ketiga untuk mengidentifikasi perbedaan dalam data.
  - Menganalisis hasil tes, dan membandingkan output dari sumber dan sistem target.
  - Mengevaluasi metrik kinerja antara dua lingkungan.
- Pengujian integrasi:
  - Lakukan tes yang melibatkan aplikasi replatformed dan komponen COBOL yang tersisa.
  - Verifikasi interaksi yang mulus antara kedua lingkungan.
- Pengujian failover dan pemulihan:
  - Uji skenario di mana satu lingkungan gagal dan lingkungan lainnya mengambil alih.
  - Pastikan konsistensi dan integritas data selama situasi failover
- Pengujian beban dan stress:
  - Lakukan pengujian dengan beban yang bervariasi untuk menilai kinerja sistem hybrid di bawah tekanan.

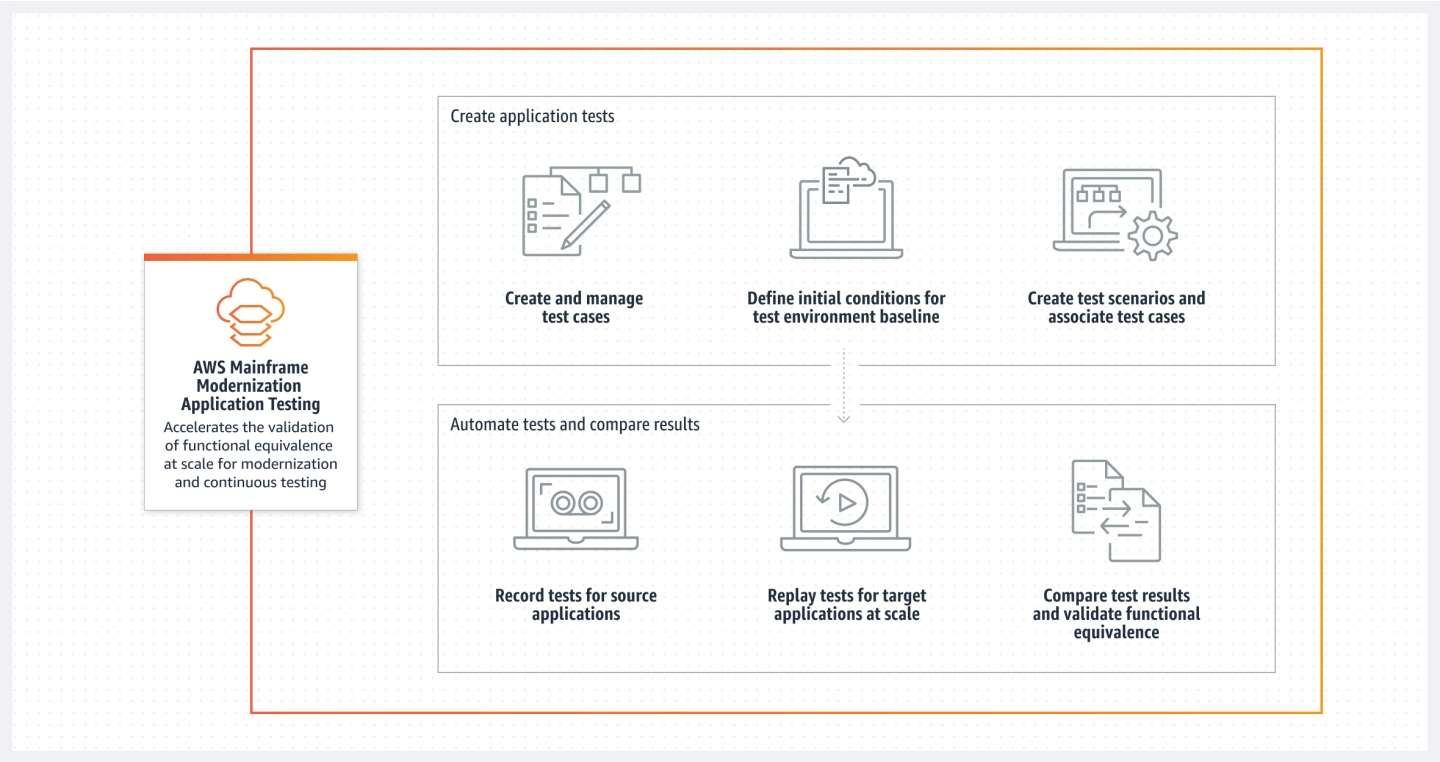
  - Identifikasi kemacetan atau masalah kinerja di kedua lingkungan.
- Dokumentasi dan pelaporan:
  - Dokumentasikan semua hasil pengujian, perbedaan, dan metrik kinerja.
  - Siapkan laporan komprehensif yang membandingkan sumber dan sistem target.

Menguji aplikasi Anda di AWS Mainframe Modernization

AWS Pengujian Aplikasi Mainframe ModernizationLayanan ini mengotomatiskan pelaksanaan tes aplikasi dalam skala besar. AWS Pengujian Aplikasi membantu mengoptimalkan dan mengurangi modernisasi aplikasi mainframe dan menguji biaya proyek.

> **Note**
>
> AWS Mainframe Modernization Layanan (Managed Runtime Environment experience) tidak lagi terbuka untuk pelanggan baru. Untuk kemampuan yang mirip dengan AWS Mainframe Modernization Service (Managed Runtime Environment experience) jelajahi AWS Mainframe Modernization Service (Self-Managed Experience). Pelanggan yang sudah ada dapat terus menggunakan layanan ini seperti biasa. Untuk informasi selengkapnya, lihat perubahan AWS Mainframe Modernization ketersediaan.

Diagram berikut menunjukkan cara AWS Pengujian Aplikasi kerja pada tingkat tinggi.



Prosesnya terdiri dari langkah-langkah ini:

1. Buat dan kelola kasus uji, yang merupakan unit tindakan pengujian terkecil. Identifikasi tipe data yang paling mewakili kesetaraan fungsional antara sistem sumber dan target.
2. Tentukan konfigurasi lingkungan pengujian dengan menentukan CloudFormation template dan atribut tambahan.
3. Buat rangkaian pengujian, yang merupakan kumpulan kasus uji.
4. Unggah dan putar ulang kumpulan data: Tangkap set data input dan output pada mainframe, unggah ke AWS, lalu putar ulang skenario pengujian pada sistem target.
5. Bandingkan kumpulan data sumber dan target. AWS Pengujian Aplikasi secara otomatis membandingkan set data output dari sumber dan sistem target. Tinjau dan evaluasi ini untuk mengidentifikasi perbedaan.

Untuk informasi lebih lanjut, lihat AWS Mainframe Modernizationdokumentasi.

Potongan

Dalam modernisasi mainframe, salah satu tantangan paling kritis adalah meminimalkan downtime dan risiko selama transisi ke platform baru. Strategi blue/green penyebaran menawarkan pendekatan yang kuat dan fleksibel untuk migrasi sistem.

Penerapan biru/hijau adalah teknik yang mengurangi waktu henti dan risiko dengan menjalankan dua lingkungan produksi identik yang disebut biru dan hijau. Begini cara kerjanya dalam konteks modernisasi mainframe:

- Lingkungan biru: Ini adalah sistem mainframe Anda saat ini yang menangani semua lalu lintas produksi.
- Lingkungan hijau: Ini adalah platform baru Anda yang dimodernisasi AWS yang siap untuk mengambil alih.

Strategi blue/green cutover mencakup langkah-langkah ini: penyediaan, siaran langsung, putar kembali jika masalah muncul, dan menyimpulkan.

Ketentuan

Pada tahap ini, Anda menyediakan lingkungan baru (hijau) AWS dengan mengikuti langkah-langkah berikut:

1. Memplatform ulang lingkungan: Zona yang dihosting Route 53 harus berisi catatan DNS yang menunjuk ke lingkungan mainframe (biru).
2. Verifikasi konektivitas: Pastikan koneksi yang tepat antara manajer transaksi Anda Akun AWS dan lokal dan Db2 untuk database. z/OS
3. Jalankan tes asap: Gunakan alamat penyeimbang AWS beban untuk mengakses lingkungan yang direplatformed dan lakukan tes asap komprehensif untuk memverifikasi hal-hal berikut:
   - Semua beban kerja yang diharapkan tersedia.
   - 3270 transaksi diproses dengan benar.
   - Interaksi data dengan Db2 untuk berfungsi seperti z/OS yang diharapkan.

Pergi langsung

Pada tahap ini, Anda mengalihkan lalu lintas ke lingkungan hijau dan memantau perubahannya.

1. Gunakan kebijakan perutean lalu lintas di Route 53 untuk mengalihkan lalu lintas:
   - Opsi A: Anda dapat mengalihkan lalu lintas sekaligus.
   - Opsi B: Atau, Anda dapat menggunakan distribusi tertimbang bertahap.
2. Memantau dan memvalidasi:
   - Perhatikan AWS lingkungan dengan cermat saat lalu lintas bergeser.
   - Periksa 3270 pemrosesan transaksi.
   - Verifikasi Db2 untuk z/OS komunikasi.
   - Memantau masalah kinerja.
   - Mintalah pengguna memvalidasi hasil transaksi.

Rollback

Jika masalah muncul, Anda dapat dengan cepat memperbarui Route 53 untuk mengarahkan lalu lintas kembali ke lingkungan mainframe (biru) lokal.

Anda harus menyelidiki dan menyelesaikan masalah sebelum Anda mencoba pemotongan lain.

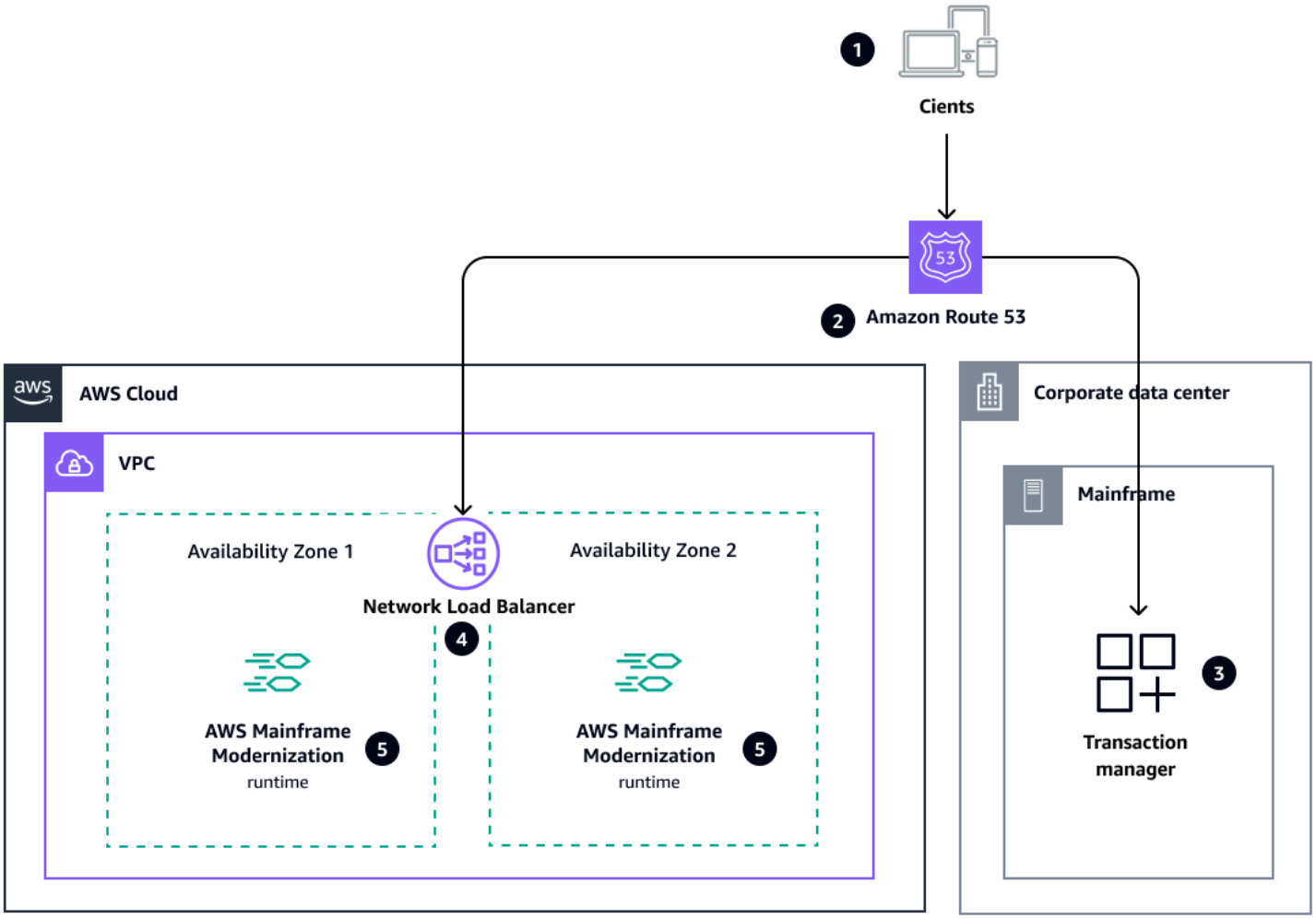
Menyimpulkan

Setelah Anda memantau lalu lintas dan memvalidasi bahwa lingkungan hijau Anda berfungsi dengan benar, Anda dapat secara bertahap meningkatkan lalu lintas aplikasi. AWS

Setelah periode stabil, Anda dapat menonaktifkan lingkungan transaksi mainframe (biru), dan menyimpan Db2 untuk z/OS database di tempat.

Arsitektur

Diagram berikut menggambarkan aliran cutover.



Proses cutover terdiri dari yang berikut:

1. Aplikasi klien, frontend, dan backend untuk frontend (BFFs) mengirim transaksi ke nama domain Route 53.
2. Route 53 merutekan koneksi ke manajer transaksi mainframe atau ke Network Load Balancer, tergantung pada kebijakan perutean yang ditentukan.
3. Manajer transaksi memproses transaksi yang dikirim ke mainframe.
4. Network Load Balancer mendistribusikan transaksi ke lingkungan replatform yang tersedia untuk diproses.
5. Lingkungan AWS Mainframe Modernization replatform memproses permintaan.

Praktik terbaik

Bagian ini menguraikan serangkaian praktik terbaik untuk mengatasi tantangan utama dalam memplatform ulang beban kerja mainframe ke lingkungan cloud sambil menjaga database di Db2 untuk z/OS.

Latensi jaringan

Untuk memprediksi secara akurat dampak latensi pemisahan aplikasi dari database Db2 selama upaya pelaporan ulang, kami menyarankan Anda melakukan evaluasi menyeluruh terhadap jumlah panggilan Db2 untuk transaksi dan proses batch. Evaluasi ini harus dilakukan dengan menggunakan data jejak dan harus mencakup langkah-langkah berikut:

- Kumpulkan data jejak: Kumpulkan jejak detail transaksi representatif dan pekerjaan batch, dan pastikan bahwa jejak menangkap semua interaksi Db2, termasuk entri dan keluar.
- Analisis data jejak: Hitung jumlah entri dan keluar Db2 untuk setiap transaksi dan pekerjaan batch, dan hitung jumlah rata-rata interaksi Db2 per transaksi dan proses batch.
- Ukur waktu respons saat ini: Periksa apakah akses Db2 selaras dengan perjanjian tingkat layanan (SLA) aplikasi Anda.
- Perkirakan latensi jaringan: Tentukan latensi jaringan yang diharapkan antara aplikasi replatformed dan database Db2. Pertimbangkan faktor-faktor seperti jarak fisik, infrastruktur jaringan, dan potensi kemacetan.
- Hitung dampak potensial: Untuk setiap transaksi dan proses batch, kalikan jumlah entri dan keluar Db2 dengan estimasi latensi jaringan. Tambahkan waktu yang dihitung ini ke waktu respons saat ini untuk memprediksi total waktu pemrosesan yang baru.
- Nilai hasil: Evaluasi apakah peningkatan latensi yang diprediksi dapat diterima untuk kebutuhan bisnis Anda dan identifikasi transaksi atau proses apa pun yang mungkin memerlukan pengoptimalan atau desain ulang untuk mengurangi masalah latensi.
- Pertimbangkan strategi mitigasi: Jelajahi opsi seperti pengumpulan koneksi, caching, atau pengambilan data batch untuk mengurangi jumlah interaksi Db2 individu. Evaluasi kemungkinan memindahkan data yang sering diakses lebih dekat ke tingkat aplikasi.

Dengan mengikuti langkah-langkah ini, Anda akan dapat membuat keputusan berdasarkan data tentang kelayakan strategi replatforming Anda dan mengidentifikasi potensi masalah kinerja sebelum

berdampak pada lingkungan produksi Anda. Pendekatan ini akan membantu memastikan transisi yang mulus sambil mempertahankan tingkat kinerja yang dapat diterima untuk aplikasi yang bergantung pada database Anda.

Keamanan

- Amankan pembuatan aplikasi Anda: Gunakan subnet pribadi di virtual private cloud (VPC) untuk AWS CodeBuild dijalankan guna membantu memastikan isolasi dan keamanan yang ditingkatkan. Menerapkan konteks tepercaya Db2 dari CodeBuild subnet CIDR untuk akses database yang aman selama proses pembuatan.
- Amankan lingkungan runtime Anda: Gunakan konteks tepercaya Db2 dari CIDR subnet runtime untuk koneksi database yang aman.
- Kelola kredenal database dengan aman: Menerapkan jadwal rotasi kredenal reguler untuk meminimalkan risiko akses yang tidak sah. Simpan kredensil Db2 dengan aman di. AWS Secrets Manager
- Menetapkan keamanan jaringan: Menerapkan segmentasi jaringan yang kuat dan aturan firewall untuk melindungi lingkungan build dan runtime. Gunakan kombinasi yang benar AWS Direct Connect dan AWS Site-to-Site VPN untuk mencapai tingkat keamanan yang diperlukan.
- Menegakkan enkripsi: Menerapkan enkripsi untuk data dalam perjalanan antara aplikasi Anda dan Db2 untuk z/OS.

Tata kelola aplikasi

- Tetapkan sumber kebenaran: Tetapkan manajemen konfigurasi perangkat lunak baru (SCM) —misalnya, GitHub —sebagai sumber kebenaran tunggal untuk kode aplikasi yang dimigrasi. Ini memastikan konsistensi dan menghilangkan perbedaan versi antara lingkungan cloud dan mainframe selama periode transisi.
- Perbarui proses manajemen perubahan: Perbarui proses manajemen perubahan untuk mempertimbangkan modifikasi kode dalam paradigma lingkungan ganda yang baru ini. Proses ini harus mencakup:
  - Hapus alur kerja persetujuan untuk perubahan kode.
  - Ulasan kode wajib sebelum menggabungkan kode ke cabang utama.
  - Prosedur penyebaran yang disinkronkan untuk memastikan bahwa kedua lingkungan menerima pembaruan secara bersamaan.

- Mekanisme rollback jika terjadi masalah di kedua lingkungan.

Elastisitas

Elastisitas komputasi awan memperkenalkan pergeseran paradigma yang secara signifikan mengubah struktur biaya mainframe dan manajemen sumber daya. Tidak seperti lingkungan mainframe tradisional, yang memiliki kapasitas tetap dan model harga berbasis puncak, platform cloud menawarkan skalabilitas dinamis dan pay-as-you-go pendekatan yang berpotensi mengarah pada penghematan biaya yang besar dan peningkatan efisiensi operasional.

Dalam lingkungan cloud, organisasi dapat meningkatkan atau menurunkan sumber daya komputasi mereka secara real time berdasarkan permintaan aktual, yang menghilangkan kebutuhan akan penyediaan berlebihan untuk mengakomodasi beban puncak. Elastisitas ini memungkinkan bisnis membayar hanya untuk sumber daya yang mereka konsumsi alih-alih berinvestasi dalam lisensi perangkat keras dan perangkat lunak yang mahal untuk menangani lonjakan penggunaan sesekali.

Untuk detail tentang cara kerja penetapan harga AWS, lihat AWS Harga.

Langkah berikutnya

Modernisasi mainframe adalah inisiatif yang kompleks dan kritis yang menuntut pengetahuan khusus dan solusi canggih. Anda dapat mempercepat proses modernisasi Anda dan mencapai hasil bisnis yang lebih cepat melalui kemitraan strategis yang akan membantu Anda dengan tugas-tugas berikut:

- Evaluasi dan prioritaskan: Tinjau aplikasi mainframe Anda dan identifikasi mana yang cocok untuk replatforming sambil menjaga database di Db2 untuk z/OS. Pertimbangkan faktor-faktor seperti kompleksitas, kekritisan bisnis, dan potensi laba atas investasi (ROI).
- Kembangkan strategi migrasi: Buat rencana terperinci untuk membuat ulang aplikasi yang Anda pilih, termasuk jadwal, alokasi sumber daya, dan strategi mitigasi risiko.
- Menilai alat dan teknologi: Meneliti dan memilih alat dan teknologi yang tepat untuk memfasilitasi proses replatforming, seperti platform modernisasi aplikasi atau alat konversi kode.
- Terlibat dengan para ahli: Pertimbangkan untuk bermitra dengan spesialis modernisasi mainframe atau perusahaan konsultan yang memiliki pengalaman dalam mereplatforming proyek.
- Bukti konsep: Mulailah dengan bukti konsep skala kecil untuk memvalidasi pendekatan Anda dan mengidentifikasi tantangan potensial sebelum meningkatkan ke aplikasi yang lebih besar.
- Pengujian dan validasi: Kembangkan strategi pengujian komprehensif untuk memastikan bahwa aplikasi yang di-replatformed berfungsi dengan benar dan menjaga integritas data dengan Db2 yang ada untuk database. z/OS
- Pelatihan dan transfer pengetahuan: Persiapkan tim Anda untuk lingkungan baru dengan memberikan pelatihan tentang aplikasi yang direplatformed dan alat atau teknologi baru apa pun yang diperkenalkan.
- Implementasi bertahap: Pertimbangkan pendekatan bertahap untuk replatforming, di mana Anda secara bertahap memigrasikan aplikasi sambil memantau kinerja dan mengatasi masalah apa pun yang muncul.
- Optimalisasi berkelanjutan: Setelah melakukan replatforming, terus memantau dan mengoptimalkan kinerja aplikasi Anda dan interaksinya dengan Db2 untuk z/OS database guna memastikan kesuksesan jangka panjang.
- Modernisasi sesuai kecepatan Anda: Sekarang beban kerja berjalan AWS dan sudah memanfaatkan cloud, mulailah merencanakan fase bayangkan ulang modernisasi Anda.

Sumber daya

Untuk informasi selengkapnya tentang migrasi dan modernisasi mainframe, lihat sumber daya berikut.

AWS dokumentasi

- Mengonfigurasi Amazon Route 53 sebagai layanan DNS Anda
- Merutekan lalu lintas ke penyeimbang beban ELB
- Perutean tertimbang
- Membentuk ulang aplikasi dengan Perangkat Lunak Roket

Referensi Perangkat Lunak Roket

- Antarmuka Panggilan Eksternal Fokus Mikro (ECI)
- Layanan Web CICS

Referensi IBM

- Konteks tepercaya (IBM Db2 untuk dokumentasi) z/OS

Alat

- Server Perusahaan Roket

AWS Pola dan panduan panduan preskriptif

- Membangun program COBOL Db2 dengan menggunakan dan AWS Mainframe ModernizationAWS CodeBuild
- DevOps untuk AWS Mainframe Modernization
- Modernisasi mainframe: Pola decoupling untuk memigrasi kode aplikasi

- Mengamankan dan merampingkan akses pengguna dalam database federasi Db2 AWS dengan menggunakan konteks tepercaya

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	7 Mei 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: 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 AWS Region 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 Region, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat Paket kesesuaian dalam dokumentasi. AWS Config

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD 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, AWS Region, 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 pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

mekanisme

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

akun anggota

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

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke. 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 AWS Region 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.