AWS Kerangka Adopsi Cloud: Perspektif platform

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: AWS Kerangka Adopsi Cloud: Perspektif platform

Table of Contents

AWS Kerangka Adopsi Cloud: Perspektif platform

Amazon Web Services (kontributor)

Oktober 2023 (sejarah dokumen)

Transformasi digital adalah enabler tunggal terbesar bagi para eksekutif untuk meningkatkan pengalaman pelanggan, inovasi, dan fleksibilitas. Ini menggunakan pembelajaran mesin (ML), kecerdasan buatan (AI), data besar, dan kecepatan dan skala cloud untuk memenuhi kondisi bisnis yang berubah dan kebutuhan pelanggan yang terus berkembang.

Amazon Web Services (AWS) adalah platform cloud yang paling komprehensif dan diadopsi secara luas di dunia. Ini dapat membantu Anda mengubah organisasi Anda sekaligus mengurangi risiko bisnis, meningkatkan kinerja lingkungan, sosial, dan tata kelola (ESG), meningkatkan pendapatan, dan meningkatkan efisiensi operasional.

AWS Cloud Adoption Framework (AWS CAF) menggunakan praktik AWS terbaik untuk membantu Anda mempercepat hasil bisnis Anda. Gunakan AWS CAF untuk mengidentifikasi dan memprioritaskan peluang transformasi, mengevaluasi dan meningkatkan kesiapan cloud Anda, dan secara berulang mengembangkan peta jalan transformasi Anda.

AWS CAF mengelompokkan panduannya dalam enam perspektif: Bisnis, Orang, Tata Kelola, Platform, Keamanan, dan Operasi. Setiap perspektif tercakup dalam panduan terpisah. Panduan ini mencakup perspektif Platform, yang berfokus pada percepatan pengiriman beban kerja cloud Anda dengan lingkungan cloud hybrid kelas perusahaan, terukur, dan terukur.

Pengantar

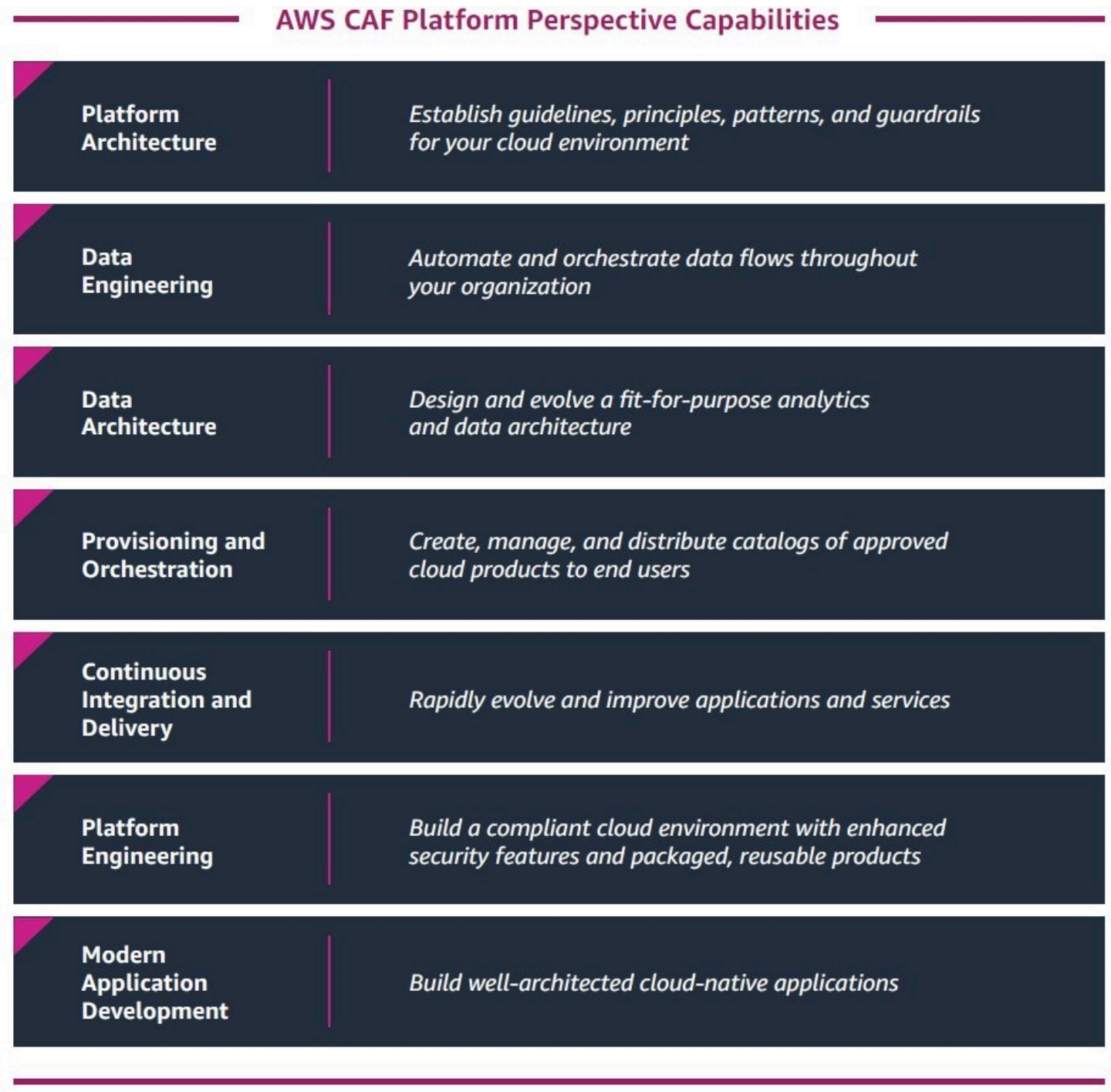
Jutaan pelanggan, termasuk startup dengan pertumbuhan tercepat, perusahaan terbesar, dan organisasi pemerintah terkemuka, menggunakannya. AWS(Lihat Kisah Sukses Pelanggan di AWS situs web.) Mereka dapat memigrasi dan memodernisasi beban kerja lama, menjadi lebih berbasis data, mengotomatiskan dan mengoptimalkan proses bisnis, dan menemukan kembali model operasi. Mereka mampu meningkatkan hasil bisnis mereka dengan mengurangi risiko bisnis, meningkatkan kinerja lingkungan, sosial, dan tata kelola (ESG), meningkatkan pendapatan, dan meningkatkan efisiensi operasional.

Kemampuan organisasi untuk secara efektif menggunakan cloud untuk mengubah secara digital (kesiapan cloud organisasi) didukung oleh serangkaian kemampuan dasar. Kemampuan adalah kemampuan organisasi untuk menggunakan proses untuk menyebarkan sumber daya (orang, teknologi, dan aset berwujud atau tidak berwujud lainnya) untuk mencapai hasil tertentu. AWS CAF mengidentifikasi kemampuan ini dan memberikan panduan preskriptif bahwa ribuan organisasi di seluruh dunia telah berhasil digunakan untuk meningkatkan kesiapan cloud mereka dan mempercepat perjalanan transformasi cloud mereka.

AWS CAF mengelompokkan kemampuannya dalam enam perspektif:

- Bisnis
- Orang
- Tata Kelola
- Platform
- Keamanan
- Operasi

Perspektif Platform berfokus pada percepatan pengiriman beban kerja cloud Anda dengan lingkungan cloud hybrid kelas perusahaan yang dapat diskalakan. Lingkungan ini terdiri dari tujuh kemampuan yang ditunjukkan pada diagram berikut. Kemampuan ini dikelola oleh para pemangku kepentingan yang secara fungsional terkait dalam perjalanan transformasi cloud mereka. Pemangku kepentingan yang khas termasuk chief technology officer (CTO), pemimpin teknologi, arsitek, dan insinyur.



Kemampuan ini dibahas secara rinci di bagian berikut dari panduan ini. Setiap bagian memberikan panduan tentang cara memulai, maju, dan akhirnya unggul dalam kemampuan tertentu.

- Arsitektur platform
- Rekayasa platform
- Arsitektur data

- Rekayasa data
- Penyediaan dan orkestrasi
- Pengembangan aplikasi modern
- Integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Perspektif Platform adalah bagian penting dari CAF. AWS Ini adalah hubungan di mana keputusan yang dibuat di semua perspektif lain bertemu untuk memberikan kelincahan dan nilai bisnis. Keputusan yang dibuat di sini membantu atau menghalangi tujuan bisnis Anda pada tingkat dasar. Perspektif Platform AWS CAF memfasilitasi penciptaan lingkungan cloud tingkat perusahaan, terukur, yang mendukung transformasi organisasi Anda. Melalui perspektif ini, AWS CAF memandu Anda dalam membangun platform yang kuat yang dapat memungkinkan perjalanan cloud Anda, yang pada akhirnya mengarah pada transformasi dan pertumbuhan bisnis yang signifikan.

Saat Anda bekerja melalui perspektif Platform, pertimbangkan koneksi lintas fungsi dengan para pemimpin bisnis yang perlu dikembangkan, dan nilai yang mereka bawa ke tim dan organisasi Anda. Tempatkan fokus tambahan pada perubahan model operasi dan topologi tim untuk memastikan bahwa persyaratan terpenuhi. Selain itu, cari untuk mengembangkan keterampilan yang dibutuhkan tim Anda untuk membangun platform dan mengaktifkan penggunaannya di seluruh tim aplikasi. Ingatlah orang-orang, bisnis, tata kelola, keamanan, dan tujuan operasional organisasi Anda saat Anda membuat keputusan ini–ini sangat penting dalam memastikan adopsi platform dan keberhasilan upaya Anda.

AWS dan Jaringan AWS Mitra menyediakan alat dan layanan, seperti lokakarya dan pelatihan, yang dapat membantu Anda dalam perjalanan ini untuk menerapkan dan meningkatkan postur keamanan Anda. AWS Professional Services adalah tim ahli global yang dapat membantu Anda mencapai hasil spesifik terkait transformasi cloud Anda melalui kumpulan penawaran yang AWS selaras dengan CAF.

Arsitektur platform

Tetapkan dan pertahankan pedoman, prinsip, pola, dan pagar pembatas untuk lingkungan cloud Anda.

Lingkungan cloud yang dirancang dengan baik membantu Anda mempercepat implementasi, mengurangi risiko, dan mendorong adopsi cloud. Kemampuan arsitektur platform menciptakan konsensus dalam organisasi Anda untuk standar perusahaan yang mendorong adopsi cloud. Anda menentukan cetak biru praktik terbaik dan pagar pembatas untuk memfasilitasi otentikasi, keamanan, jaringan, dan pencatatan dan pemantauan. Selain itu, Anda mempertimbangkan dan merencanakan beban kerja yang mungkin perlu Anda pertahankan di tempat karena latensi, pemrosesan data, atau persyaratan residensi data dan mengevaluasi kasus penggunaan cloud hybrid seperti cloud bursting, pencadangan dan pemulihan bencana ke cloud, pemrosesan data terdistribusi, dan komputasi tepi.

Mulai

Tentukan strategi multi-akun

Strategi multi-akun yang baik mempertimbangkan masalah skala dan efisiensi operasional. Ini berarti mengisolasi beban kerja Anda ke dalam pola logis yang paling memenuhi kebutuhan operasional Anda. Kami menyarankan Anda memulai dengan serangkaian akun dasar untuk mengakomodasi layanan terpusat dan terdesentralisasi di perusahaan Anda. Anda dapat memusatkan fungsi keamanan, keuangan, dan operasional untuk mengelola dan mengatur tim dan akun Anda yang terdistribusi dan otonom secara efektif. Anda akan ingin menyelaraskan seluruh organisasi Anda untuk memahami bagaimana platform dan beban kerja Anda akan tersegmentasi dan dikelola. Memahami struktur ini membantu Anda memastikan bahwa prinsip-prinsip keamanan ada untuk otentikasi dan otorisasi sambil menyelaraskan dengan mengembangkan kebijakan penggunaan yang dapat diterima untuk platform.

Tentukan kontrol pencegahan

Rencanakan lingkungan multi-akun yang aman dengan seperangkat kontrol default (pagar pembatas) yang disematkan. Mulailah memahami dan menggunakan mekanisme seperti kebijakan kontrol layanan (SCPs) untuk mengelola penggunaan layanan di seluruh organisasi Anda, termasuk Wilayah AWS yang tersedia untuk konsumsi dalam platform cloud Anda. Kebijakan menyediakan mekanisme terpusat untuk mengendalikan izin maksimum yang tersedia untuk semua akun dan memastikan bahwa mereka mematuhi pedoman kontrol akses organisasi.

Mendefinisikan struktur unit organisasi

Unit organisasi (OUs) berfungsi sebagai cara praktis untuk mengelola dan mengkategorikan akun berdasarkan persyaratan peraturan dan lingkungan siklus hidup pengembangan perangkat lunak (SDLC). Dengan menggunakan OUs, organisasi merampingkan proses penerapan kebijakan dan izin yang sesuai di seluruh infrastruktur cloud mereka. Beban kerja OUs dirancang khusus untuk akun yang mendukung sumber daya infrastruktur aplikasi, dan memastikan bahwa kebijakan yang tepat ditegakkan. Menggunakan OUs dan SCPs membantu meningkatkan keamanan dan kepatuhan infrastruktur cloud organisasi Anda sekaligus memastikan kelancaran pengoperasian aplikasi dan layanan Anda. Ini pada akhirnya mengarah pada proses adopsi cloud yang lebih efisien dan kuat.

Tentukan konektivitas jaringan

Konektivitas jaringan adalah aspek penting dari setiap infrastruktur cloud yang mendukung penciptaan jaringan yang aman, terukur, dan sangat tersedia untuk mendukung aplikasi dan beban kerja. Jaringan yang dirancang dengan baik memberikan kinerja tinggi secara konsisten dan memastikan operasi yang mulus di berbagai lingkungan.

Saat mendesain arsitektur jaringan, pertimbangkan apakah Anda memiliki beban kerja yang ingin Anda pertahankan di tempat karena latensi, pemrosesan data, atau persyaratan residensi data. Dengan mengevaluasi kasus penggunaan cloud hybrid seperti cloud bursting, pencadangan dan pemulihan bencana ke cloud, pemrosesan data terdistribusi, dan komputasi tepi, Anda dapat mengidentifikasi persyaratan utama untuk aspek-aspek berikut:

- Konektivitas ke dan dari internet. Aspek ini melibatkan penyediaan koneksi yang aman dan andal antara aplikasi atau beban kerja Anda dan internet. Konektivitas ini sangat penting untuk memfasilitasi akses ke sumber daya berbasis web, memungkinkan komunikasi antara pengguna dan aplikasi, dan memastikan bahwa layanan Anda dapat diakses oleh publik bila diperlukan.
- Konektivitas di seluruh lingkungan cloud Anda. Area ini berfokus pada membangun koneksi yang kuat di antara berbagai komponen dan layanan dalam infrastruktur cloud Anda. Ini memastikan bahwa data dan sumber daya mudah dibagikan dan diakses di berbagai layanan cloud, mempromosikan kolaborasi yang efisien dan operasi yang lebih lancar. Pertimbangan utama di sini adalah penggunaan virtual private cloud (VPCs). Untuk menjaga hal-hal sederhana, pertimbangkan untuk membuat standar tentang cara VPCs dibuat dan dilacak. Pertimbangkan untuk membuat standar ini secara terprogram, dan rencanakan untuk menggunakan solusi manajemen alamat IP (IPAM). Alokasikan ruang IP yang cukup untuk memungkinkan pertumbuhan, dan rancang struktur subnet untuk pemecahan masalah yang mudah saat menggunakan beberapa Availability Zone.

Pastikan untuk mengikuti praktik terbaik keamanan VPCs saat Anda merancang dan menerapkan konektivitas jaringan.

- Konektivitas antara jaringan lokal dan lingkungan cloud Anda. Aspek ini berkaitan dengan integrasi infrastruktur lokal dengan lingkungan berbasis cloud. Dengan menciptakan koneksi yang aman dan andal antara keduanya, organisasi mendapat manfaat dari keunggulan arsitektur hybrid. Misalnya, Anda dapat menggunakan sumber daya lokal dan layanan cloud secara bersamaan untuk meningkatkan kinerja, skalabilitas, dan pengoptimalan biaya.

Dengan menangani tiga bidang utama konektivitas jaringan ini, Anda dapat membangun infrastruktur cloud yang kuat yang mendukung aplikasi dan beban kerja Anda secara efektif, sehingga Anda dapat memanfaatkan manfaat adopsi cloud. Perhatikan persyaratan jaringan, dan buat desain sederhana yang memungkinkan Anda menskalakan sesuai dengan strategi multi-akun Anda.

Tentukan strategi DNS

Strategi DNS yang terencana dengan baik membantu Anda menghindari komplikasi saat lingkungan cloud Anda tumbuh. Jika Anda mempertahankan kemampuan DNS lokal, kami sarankan Anda merancang arsitektur DNS hybrid yang menggunakan infrastruktur DNS lokal bersama dengan DNS cloud untuk persyaratan DNS berbasis cloud apa pun. Integrasikan resolusi DNS dengan lingkungan DNS lokal dengan menggunakan titik akhir resolver dan aturan penerusan. Gunakan zona yang dihosting pribadi untuk menyimpan informasi tentang bagaimana Anda ingin DNS cloud merespons kueri untuk domain dan subdomainnya dalam satu atau beberapa jaringan.

Tentukan standar penandaan

Menandai sumber daya adalah praktik penting untuk mengelola biaya secara efektif dan mengidentifikasi kepemilikan sumber daya. Pertimbangkan bagaimana organisasi Anda akan lebih lanjut memungkinkan konsumsi di cloud, termasuk penggunaan layanan tertentu dalam platform. Tentukan strategi penandaan yang melacak sumber daya mana yang digunakan oleh tim mana. Ambil masukan dari perspektif Operasi AWS CAF dan gunakan tag untuk mengotomatiskan tugas untuk infrastruktur yang Anda gunakan.

Selain itu, dengan menandai sumber daya dengan metadata yang relevan, Anda dapat mengelompokkan dan melacak pengeluaran Anda berdasarkan persyaratan organisasi yang ditentukan dalam kemampuan Cloud Financial Management (CFM) dalam perspektif Tata Kelola CAF.AWS Identifikasi mekanisme pelaporan yang mendukung praktik akuntansi dan keuangan Anda, termasuk tindakan yang harus diambil ketika kebijakan keuangan dilanggar.

Tentukan strategi observabilitas

Menetapkan strategi observabilitas adalah langkah penting untuk mengoptimalkan dan mengamankan arsitektur cloud Anda. Strategi ini berkisar pada transformasi metrik dan log yang dihasilkan oleh layanan cloud Anda menjadi wawasan yang dapat ditindaklanjuti untuk pengambilan keputusan strategis. Prioritaskan pemantauan indikator kinerja utama dan siapkan peringatan untuk mengatasi masalah potensial terlebih dahulu. Untuk mencegah proliferasi alat, mengoptimalkan biaya, dan fokus pada apa yang paling penting bagi organisasi Anda, gabungkan strategi pengamatan ini di platform dan aplikasi Anda. Untuk panduan lebih lanjut, lihat presentasi kami tentang Mengembangkan strategi observabilitas (AWS re:Invent 2022).

Maju

Tentukan kontrol proaktif dan detektif

Untuk maju, organisasi Anda harus mengidentifikasi kebutuhan akan kontrol proaktif dan detektif (pagar pembatas) dalam lingkungan. Buat kebijakan yang menentukan pagar pembatas atau batasan yang dimiliki peran dan pengguna di akun yang terletak di dalam unit organisasi (OU). Tinjau pagar pembatas detektif default untuk platform, dan pilih pagar pembatas mana yang akan diterapkan. Buat kontrol preventif dan detektif tambahan sesuai kebutuhan, dan kelompokkan mereka OUs untuk menyelaraskannya dengan strategi multi-akun Anda. Pertimbangkan alat dan mekanisme organisasi mana yang Anda perlukan untuk memeriksa sumber daya yang tidak sesuai yang diidentifikasi oleh kontrol detektif.

Tentukan standar untuk orientasi layanan

Buat standar untuk penggunaan platform yang dapat diterima dan pola yang terkait dengan konsumsi layanan dan bagaimana hal itu akan diatur. Pertimbangkan layanan awal mana yang diizinkan untuk digunakan. Buat dokumen yang menguraikan standar ini dan mempublikasikannya kepada pengguna dan operator platform. Pastikan bahwa standar-standar ini beradaptasi dari waktu ke waktu untuk memenuhi perubahan tujuan organisasi dan kemampuan komputasi awan yang berkembang.

Tentukan pola dan prinsip

Pertimbangkan pola arsitektur mana yang akan diizinkan dalam organisasi Anda dengan menggunakan input dari pemilik aplikasi, dan mulailah menentukan cetak biru untuk standardisasi. Standardisasi memungkinkan tata kelola yang lebih besar dan beban administrasi yang lebih rendah

saat Anda menskalakan di cloud. Tentukan pola yang akan menggunakan infrastruktur sebagai kode (IAc) dan rencanakan model penyebaran yang disederhanakan dengan menggunakan katalog layanan yang terintegrasi ke dalam proses kontrol perubahan dan sistem manajemen layanan TI (ITSM) Anda. Tentukan bagaimana cetak biru ini akan digunakan dan keadaan untuk mengizinkan pengecualian. Rencanakan pengecualian tersebut dan tata kelola mereka, dengan pertimbangan untuk otentikasi, pemantauan keamanan, dan pagar pembatas.

Unggul

Tentukan pola remediasi

Pertimbangkan cara membuat anotasi dan memprioritaskan temuan pagar pembatas detektif Anda sehingga mereka dapat diperbaiki sesuai dengan kerangka kerja keamanan dan kepatuhan Anda. Rencanakan untuk menggunakan otomatisasi untuk mendeteksi out-of-policy penyediaan sumber daya, termasuk yang melanggar kebijakan anggaran dan penandaan. Identifikasi kemampuan yang diperlukan untuk menetapkan dan mengukur tujuan tingkat layanan sambil memperbarui runbook dan buku pedoman Anda. Tetapkan tinjauan berkala dari praktik ini dan mekanisme umpan balik untuk menangkap data yang terkait dengan evolusi platform. Tentukan mekanisme untuk membuat dan memperbarui runbook dan buku pedoman yang sesuai.

Komunikasikan dan perbaiki kebijakan

Buat sistem manajemen konten terpusat untuk semua dokumentasi dan distribusikan ke pengguna dan operator platform. Buat mekanisme untuk menangkap umpan balik untuk pertimbangan masa depan tentang perubahan kebijakan.

Memahami kemampuan manajemen keuangan

Organizations berkembang ketika mereka mempertahankan pemahaman yang transparan dan komprehensif tentang anggaran mereka. Ini memberdayakan mereka untuk membuat keputusan yang terinformasi dengan baik, mengalokasikan sumber daya secara efisien, dan mencapai tujuan strategis mereka. Pandangan yang jelas tentang anggaran membantu organisasi unggul dengan memfasilitasi pengambilan keputusan yang terinformasi, alokasi sumber daya yang efektif, pengendalian biaya, pengukuran kinerja, dan pemeliharaan akuntabilitas dan kepatuhan. Ini pada akhirnya menghasilkan organisasi yang lebih efisien, stabil secara finansial, dan makmur. Ketika Anda memiliki strategi penandaan yang sukses, Anda dapat menggunakan filter biaya AWS Budgetsuntuk memfilter biaya berdasarkan tag sumber daya. Ini membantu Anda membuat anggaran

yang disesuaikan dengan proyek, departemen, lingkungan, atau kriteria tertentu, yang selanjutnya meningkatkan kemampuan manajemen keuangan. Anda dapat mengaitkan tag alokasi AWS biaya dan Cost Categories dengan tag untuk mendorong wawasan keuangan dan transparansi saat melaporkan biaya.

Rekayasa platform

Bangun lingkungan cloud multi-akun yang aman dan sesuai dengan produk cloud yang dikemas dan dapat digunakan kembali.

Untuk mendukung inovasi dengan memungkinkan tim pengembangan, platform perlu beradaptasi dengan cepat untuk memenuhi tuntutan bisnis. (Lihat perspektif Bisnis AWS CAF.) Ini harus dilakukan sementara cukup fleksibel untuk beradaptasi dengan tuntutan manajemen produk, cukup kaku untuk mematuhi kendala keamanan, dan cukup cepat untuk memungkinkan kebutuhan operasional. Proses ini membutuhkan pembangunan lingkungan cloud multi-akun yang sesuai dengan fitur keamanan yang disempurnakan, dan produk cloud yang dikemas dan dapat digunakan kembali.

Lingkungan cloud yang efektif memungkinkan tim Anda menyediakan akun baru dengan mudah sambil memastikan bahwa akun tersebut sesuai dengan kebijakan organisasi. Serangkaian produk cloud yang dikuratori memungkinkan Anda untuk mengkodifikasi praktik terbaik, membantu Anda dalam tata kelola, dan membantu meningkatkan kecepatan dan konsistensi penerapan cloud Anda. Terapkan cetak biru praktik terbaik Anda, dan pagar pembatas detektif dan pencegahan. Integrasikan lingkungan cloud Anda dengan lanskap yang ada untuk mengaktifkan kasus penggunaan cloud hybrid yang diinginkan.

Otomatiskan alur kerja penyediaan akun dan gunakan beberapa akun untuk mendukung tujuan keamanan dan tata kelola Anda. Siapkan konektivitas antara lingkungan lokal dan cloud serta antar akun cloud yang berbeda. Terapkan federasi antara penyedia identitas yang ada (iDP) dan lingkungan cloud Anda sehingga pengguna dapat mengautentikasi dengan menggunakan kredensi login yang ada. Pusatkan pencatatan, buat audit keamanan lintas akun, buat resolver DNS masuk dan keluar, dan dapatkan visibilitas dasbor ke akun dan pagar pembatas Anda.

Mengevaluasi dan mensertifikasi layanan cloud untuk konsumsi sesuai dengan standar perusahaan dan manajemen konfigurasi. Package dan terus meningkatkan standar perusahaan sebagai produk self-service deployable dan layanan habis pakai. Manfaatkan infrastruktur sebagai kode (IAc) untuk mendefinisikan konfigurasi dengan cara deklaratif. Buat tim pemberdayaan untuk menginjili platform kepada pengembang dan pengguna bisnis dan memungkinkan mereka untuk membangun integrasi yang mempercepat adopsi di seluruh organisasi Anda.

Menyelesaikan tugas yang dibahas di bagian berikut mengharuskan Anda untuk membangun kemampuan dan tim untuk mengembangkan organisasi Anda menuju rekayasa platform modern. Untuk detail teknis, lihat Membangun Cloud Foundation Anda di AWS whitepaper.

Mulai

Bangun landing zone dan gunakan pagar pembatas

Saat Anda memulai perjalanan menuju rekayasa platform yang matang, Anda harus terlebih dahulu menerapkan landing zone Anda dengan pagar pembatas detektif dan preventif sebagaimana didefinisikan dalam kemampuan arsitektur platform. Guardrails memastikan bahwa standar organisasi tidak dilanggar karena pemilik aplikasi menggunakan sumber daya cloud. Dengan mekanisme ini, Anda mengotomatiskan alur kerja penyediaan akun untuk menggunakan beberapa akun yang mendukung tujuan keamanan dan tata kelola Anda.

Menetapkan otentikasi

Menerapkan manajemen identitas dan kontrol akses di semua lingkungan, sistem, beban kerja, dan proses sesuai dengan standar yang ditentukan dalam perspektif Keamanan AWS CAF. Untuk identitas tenaga kerja, batasi penggunaan AWS Identity and Access Management (IAM) pengguna dan sebagai gantinya mengandalkan penyedia identitas yang memungkinkan Anda mengelola identitas di tempat terpusat. Ini membuatnya lebih mudah untuk mengelola akses di beberapa aplikasi dan layanan, karena Anda membuat, mengelola, dan mencabut akses dari satu lokasi. Gunakan proses yang ada untuk mengelola pembuatan, pembaruan, dan penghapusan akses untuk menyertakan AWS lingkungan Anda.

Menyebarkan jaringan Anda

Sesuai dengan desain arsitektur platform Anda, buat akun jaringan terpusat untuk mengontrol lalu lintas masuk dan keluar ke dan dari lingkungan Anda. Kami menyarankan Anda merancang jaringan Anda untuk konektivitas yang disediakan dengan cepat antara jaringan lokal dan AWS lingkungan Anda, ke dan dari internet, dan di seluruh lingkungan Anda. AWS Memusatkan manajemen jaringan Anda memungkinkan Anda untuk menyebarkan kontrol jaringan untuk mengisolasi jaringan dan konektivitas di seluruh lingkungan Anda dengan menggunakan kontrol preventif dan reaktif.

Kumpulkan, agregat, dan lindungi data peristiwa dan log

Gunakan observabilitas CloudWatch lintas akun Amazon. Ini menyediakan antarmuka terpadu untuk mencari, memvisualisasikan, dan menganalisis metrik, log, dan jejak di seluruh akun tertaut Anda, dan menghilangkan batasan akun.

Jika organisasi Anda memiliki persyaratan kepatuhan khusus untuk kontrol log terpusat dan keamanan, pertimbangkan untuk menyiapkan akun arsip log khusus. Ini menawarkan repositori

terpusat dan terenkripsi khusus untuk data log. Tingkatkan keamanan arsip ini dengan memutar kunci enkripsi secara teratur.

Menerapkan kebijakan yang kuat untuk melindungi data log sensitif, menggunakan teknik masking seperlunya. Gunakan agregasi log untuk kepatuhan, keamanan, dan log audit, dan pastikan penggunaan pagar pembatas dan konstruksi identitas yang ketat untuk mencegah perubahan yang tidak sah pada konfigurasi log.

Tetapkan kontrol

Sesuai dengan definisi dari perspektif Keamanan AWS CAF, gunakan kemampuan keamanan dasar yang memenuhi persyaratan bisnis Anda. Gunakan kontrol pencegahan dan detektif tambahan, dan sediakan kontrol tersebut secara terprogram dan konsisten di semua akun Anda jika diperlukan. Integrasikan kontrol detektif ke dalam perkakas operasional sebagaimana didefinisikan oleh kemampuan arsitektur platform sehingga sumber daya yang tidak sesuai dapat ditinjau oleh mekanisme operasional.

Menerapkan manajemen keuangan cloud

Sesuai dengan perspektif Tata Kelola AWS CAF, terapkan tag alokasi biaya, dan Cost Categories AWS yang menyelaraskan strategi penandaan organisasi Anda dengan akuntabilitas keuangan untuk konsumsi cloud. AWS Cost Categories memungkinkan Anda menagih atau menampilkan biaya cloud kembali ke pusat biaya internal dengan menggunakan alat seperti AWS Cost Explorerdan data penagihan yang dipublikasikan di AWS Cost and Usage Report.

Maju

Bangun otomatisasi infrastruktur

Sebelum Anda melanjutkan, evaluasi dan sertifikasi layanan cloud untuk konsumsi sesuai dengan arsitektur platform Anda. Kemudian, paket dan terus meningkatkan standar perusahaan sebagai produk yang dapat digunakan dan layanan habis pakai, dan gunakan infrastruktur sebagai kode (IAc) untuk mendefinisikan konfigurasi dengan cara deklaratif. Otomatisasi infrastruktur meniru siklus pengembangan perangkat lunak dengan memungkinkan akses ke layanan tertentu di setiap akun dengan kontrol akses berbasis peran (RBAC) atau kontrol akses berbasis atribut (ABAC). Terapkan metode untuk menyediakan akun baru dengan cepat dan menyelaraskannya dengan kemampuan layanan dan manajemen insiden Anda dengan menggunakan APIs, atau mengembangkan kemampuan swalayan. Otomatiskan integrasi jaringan dan alokasi IP saat akun

dibuat untuk memastikan kepatuhan dan keamanan jaringan. Integrasikan akun baru dengan solusi manajemen layanan TI (ITSM) Anda dengan menggunakan konektor asli yang dikonfigurasikan untuk dioperasikan. AWS Perbarui buku pedoman dan runbook Anda sebagaimana mestinya.

Menyediakan layanan observabilitas terpusat

Untuk mencapai observabilitas cloud yang efektif, platform Anda harus mendukung pencarian dan analisis real-time dari data log lokal dan terpusat. Saat skala operasi Anda, kemampuan platform Anda untuk mengindeks, memvisualisasikan, dan menafsirkan log, metrik, dan jejak adalah kunci untuk mengubah data mentah menjadi wawasan yang dapat ditindaklanjuti.

Dengan mengkorelasikan log, metrik, dan jejak, Anda dapat mengekstrak kesimpulan yang dapat ditindaklanjuti dan mengembangkan tanggapan yang ditargetkan dan terinformasi. Tetapkan aturan yang memungkinkan respons proaktif terhadap peristiwa atau pola keamanan yang diidentifikasi dalam log, metrik, atau jejak Anda. Saat AWS solusi Anda berkembang, pastikan bahwa strategi pemantauan Anda berskala bersama-sama untuk mempertahankan dan meningkatkan kemampuan observabilitas Anda.

Menerapkan manajemen sistem dan tata kelola AMI

Organizations yang menggunakan instans Amazon Elastic Compute Cloud (Amazon EC2) secara ekstensif memerlukan perkakas operasional untuk mengelola instans dalam skala besar. Manajemen aset perangkat lunak, deteksi dan respons titik akhir, manajemen inventaris, manajemen kerentanan, dan manajemen akses adalah kemampuan dasar bagi banyak organisasi. Kemampuan ini sering disampaikan melalui agen perangkat lunak yang diinstal pada instance. Kembangkan kemampuan untuk mengemas agen dan konfigurasi khusus lainnya ke Amazon Machine Images (AMI), dan buat AMI ini tersedia bagi konsumen platform cloud. Gunakan kontrol pencegahan dan detektif yang mengatur penggunaan AMI ini. AMI harus berisi perkakas yang memungkinkan pengelolaan instans EC2 yang berjalan lama dalam skala besar, terutama untuk beban kerja Amazon EC2 yang dapat berubah yang tidak mengkonsumsi yang baru secara teratur. AMIs Anda dapat menggunakan AWS Systems Managerskala besar untuk mengotomatiskan peningkatan agen, mengumpulkan inventaris sistem, mengakses instans EC2 dari jarak jauh, dan menambal kerentanan sistem operasi.

Mengelola penggunaan kredensi

Sesuai dengan perspektif Keamanan AWS CAF, implementasikan peran dan kredensi sementara. Gunakan perkakas untuk mengelola akses jarak jauh ke instans atau sistem lokal dengan menggunakan agen pra-instal tanpa menyimpan rahasia. Kurangi ketergantungan pada kredensil

jangka panjang, dan pindai kredensi hardcode di templat IAc Anda. Jika Anda tidak dapat menggunakan kredensil sementara, gunakan alat terprogram seperti token aplikasi dan kata sandi basis data untuk mengotomatiskan rotasi dan manajemen kredensi. Kodifikasi pengguna, grup, dan peran dengan menggunakan prinsip hak istimewa paling rendah dengan IAc, dan mencegah pembuatan akun identitas secara manual dengan menggunakan pagar pembatas.

Membangun perkakas keamanan

Alat pemantauan keamanan harus mendukung pemantauan keamanan terperinci di seluruh infrastruktur, aplikasi, dan beban kerja dan memberikan tampilan agregat untuk analisis pola. Seperti semua alat manajemen keamanan lainnya, Anda harus memperluas alat deteksi dan respons (XDR) yang diperluas untuk menyediakan fungsi untuk menilai, mendeteksi, merespons, dan memulihkan keamanan aplikasi, sumber daya, dan lingkungan Anda sesuai dengan persyaratan yang ditentukan dalam perspektif Keamanan AWS CAF. AWS

Unggul

Sumber dan distribusikan konstruksi identitas dengan otomatisasi

Kodifikasi dan konstruksi identitas versi seperti peran, kebijakan, dan templat dengan alat IAc. Gunakan alat validasi kebijakan untuk memeriksa peringatan keamanan, kesalahan, peringatan umum, perubahan yang disarankan pada kebijakan IAM Anda, dan temuan lainnya. Jika perlu, terapkan dan hapus konstruksi identitas yang menyediakan akses sementara ke lingkungan secara otomatis, dan melarang penerapan oleh individu yang menggunakan konsol.

Tambahkan deteksi dan peringatan untuk pola anomali di seluruh lingkungan

Secara proaktif menilai lingkungan untuk kerentanan yang diketahui dan menambahkan deteksi untuk peristiwa dan pola aktivitas yang tidak biasa. Tinjau temuan dan buat rekomendasi kepada tim arsitektur platform untuk perubahan yang mendorong efisiensi dan inovasi lebih lanjut.

Menganalisis dan memodelkan ancaman

Menerapkan pemantauan dan pengukuran berkelanjutan terhadap tolok ukur industri dan keamanan sesuai dengan persyaratan dari perspektif Keamanan AWS CAF. Saat Anda menerapkan pendekatan instrumentasi Anda, tentukan jenis data dan informasi peristiwa mana yang paling baik

menginformasikan fungsi manajemen keamanan Anda. Pemantauan ini mencakup beberapa vektor serangan, termasuk penggunaan layanan. Fondasi keamanan Anda harus mencakup kemampuan komprehensif untuk pencatatan dan analitik yang aman di seluruh lingkungan multi-akun Anda yang mencakup kemampuan untuk mengkorelasikan peristiwa dari berbagai sumber. Cegah perubahan pada konfigurasi ini dengan kontrol dan pagar pembatas tertentu.

Terus mengumpulkan, meninjau, dan mengisi kembali izin

Rekam perubahan pada peran dan izin identitas dan terapkan peringatan saat pagar pembatas detektif mendeteksi penyimpangan dari status konfigurasi yang diharapkan. Gunakan alat pengenal gabungan dan pola untuk meninjau koleksi acara terpusat Anda dan menentukan izin sesuai kebutuhan.

Pilih, ukur, dan terus tingkatkan metrik platform Anda

Untuk memungkinkan operasi platform yang sukses, buat dan tinjau metrik komprehensif secara rutin. Pastikan bahwa mereka selaras dengan tujuan organisasi dan kebutuhan pemangku kepentingan. Lacak kinerja platform dan metrik peningkatan, dan gabungkan parameter operasional seperti tambalan, cadangan, dan kepatuhan dengan menggunakan pemberdayaan tim dan indikator adopsi alat.

Gunakan observabilitas CloudWatch lintas akun untuk manajemen metrik yang efisien. Layanan ini merampingkan agregasi dan visualisasi data untuk memungkinkan keputusan berdasarkan informasi dan peningkatan yang ditargetkan. Gunakan metrik ini sebagai indikator keberhasilan dan pendorong perubahan untuk menumbuhkan lingkungan perbaikan berkelanjutan.

Arsitektur data

Merancang dan mengembangkan arsitektur fit-for-purpose data dan analitik.

Arsitektur data dan analitik yang dirancang dengan baik sangat penting untuk mendapatkan wawasan yang dapat ditindaklanjuti. Dengan merancang dan mengembangkan arsitektur fit-for-purpose data dan analitik, organisasi mengurangi kompleksitas, biaya, dan utang teknis sambil membuka wawasan berharga dari volume data mereka yang terus berkembang. Dengan menyelaraskan dengan prinsip-prinsip AWS CAF, bisnis dapat membuat arsitektur data yang terintegrasi secara mulus dengan platform yang ada. Penyelarasan ini memposisikan organisasi untuk memanfaatkan keuntungan yang ditawarkan oleh teknologi pemrosesan data dan analitik modern.

Arsitektur data dan analitik adalah cetak biru kemampuan organisasi untuk memperoleh nilai dari data. Ini membantu organisasi mendapatkan wawasan bisnis baru dan merupakan katalisator untuk pertumbuhan bisnis. Untuk mendukung kebutuhan bisnis, arsitektur data modern harus selaras dengan tujuan bisnis jangka pendek dan jangka panjang dan unik untuk persyaratan budaya dan kontekstual organisasi. Di dunia sekarang ini, keberhasilan implementasi dan adopsi arsitektur data dan analitik didasarkan pada prinsip memungkinkan data yang tepat pada waktu yang tepat untuk konsumen yang tepat.

Hal ini dicapai dengan merencanakan dan mengatur bagaimana aset data organisasi dimodelkan, secara fisik atau logis, bagaimana data diamankan, dan bagaimana model data ini berinteraksi satu sama lain untuk mengatasi masalah bisnis dan untuk mendapatkan pola yang tidak diketahui dan menghasilkan wawasan.

Mulai

Tentukan kemampuan menyeluruh

Dalam lingkungan bisnis saat ini, sangat penting bagi platform analitik data modern untuk memperoleh nilai dari data untuk mendukung berbagai domain dalam organisasi. Alih-alih mengadopsi pendekatan arsitektur data tunggal, arsitektur data modern harus menyertakan toolset dan pola yang dibuat khusus dan dioptimalkan untuk kasus penggunaan tertentu. Arsitektur harus dapat berkembang dan mencakup blok bangunan dasar, seperti data lake yang dapat diskalakan, layanan analitik yang dibangun khusus, akses data terpadu, dan tata kelola terpadu.

Mengatur zona data

Bagaimana data diatur dan disimpan untuk akses cepat dan mudah adalah aspek penting dari arsitektur data. Ini dapat dicapai dengan menyiapkan zona data khusus dalam danau data. Zona data dikategorikan sebagai berikut:

- Data mentah yang dikumpulkan dari sumber heterogen
- Data yang dikuratori dan diubah untuk mendukung kebutuhan analitis setiap domain
- Gunakan case atau data mart berbasis produk untuk kebutuhan pelaporan
- Data yang terpapar secara eksternal dengan kontrol keamanan dan kepatuhan

Merencanakan kelincahan dan demokratisasi data

Efektivitas platform analitik tergantung pada kecepatan penyediaan data serta demokratisasi data yang disediakan untuk konsumsi. Kelincahan penyediaan data dicapai dengan kemampuan arsitektur data untuk mendapatkan dan memproses data dalam berbagai cara–seperti real-time, near-real time, batch, micro-batch, atau hybrid–berdasarkan kasus penggunaan. Demokratisasi data dicapai dengan mendefinisikan berbagi data dan alur kerja kontrol akses yang dipantau oleh pengelola data. Menerapkan pasar data adalah salah satu enabler untuk mendemokratisasi data.

Tentukan pengiriman data yang aman

Arsitektur data modern adalah benteng bagi dunia luar dalam keamanan tetapi memungkinkan akses mudah ke karyawan atau pengguna data, sebagaimana didefinisikan oleh fungsi pekerjaan mereka, dan mematuhi pembatasan kepatuhan seperti Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA), informasi yang dapat diidentifikasi secara pribadi (PII), Peraturan Perlindungan Data Umum (GDPR), dan sebagainya. Ini dicapai dengan metode kontrol akses berbasis peran (RBAC) dan kontrol akses berbasis tag (TBAC). Pada AWS, tag digunakan untuk mengontrol akses ke data untuk menyederhanakan manajemen kontrol akses. Lakukan ini sejalan dengan prinsip-prinsip yang diuraikan dalam perspektif Keamanan AWS CAF.

Rencanakan efektivitas biaya

Gudang data tradisional menyediakan komputasi dan penyimpanan yang digabungkan erat dengan biaya pemanfaatan sumber daya yang tinggi. Arsitektur modern memisahkan komputasi dan penyimpanan, dan mengimplementasikan penyimpanan berjenjang berdasarkan siklus hidup data.

Misalnya, aktif AWS, Anda dapat menggunakan Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) untuk mengontrol biaya dan memisahkan penyimpanan data dari komputasi. Kelas penyimpanan Amazon S3 dibuat khusus untuk menyediakan penyimpanan biaya terendah untuk pola akses yang berbeda. Selain itu, alat AWS komputasi (seperti Amazon Athena AWS Glue, Amazon Redshift, dan SageMaker Amazon Runtime) tidak memiliki server, jadi Anda tidak perlu mengelola infrastruktur, dan Anda hanya membayar untuk apa yang Anda gunakan.

Maju

Arsitektur data modern dapat ditingkatkan lebih lanjut untuk meningkatkan luasnya penggunaan data — dari analitik standar yang mendukung fungsi bisnis dan operasional hingga kemampuan yang lebih kompleks yang mendukung prediksi dan wawasan — dan membantu mendukung pengambilan keputusan yang lebih cepat. Untuk mencapai hal ini, arsitektur mendukung kemampuan yang dijelaskan di bagian berikut.

Memahami rekayasa fitur

Rekayasa fitur menggunakan pembelajaran mesin dan melibatkan pengaturan toko fitur atau marts fitur. Tim ilmu data membuat fitur baru (atribut turunan) untuk model pembelajaran yang diawasi dan tidak diawasi dan menyimpannya di marts fitur untuk transformasi yang disederhanakan dan akurasi data yang ditingkatkan. Perusahaan dapat menggunakan kembali fitur di beberapa model analitik, yang meningkatkan kecepatan ke pasar.

Rencanakan untuk mendenormalisasi kumpulan data

Membangun kumpulan data denormalisasi atau data mart dapat secara signifikan menyederhanakan kumpulan data untuk pengguna bisnis dengan membuat data yang diperlukan tersedia di satu lokasi dan meningkatkan kecepatan analitik. Jika dirancang dengan hati-hati, satu catatan dapat mendukung beberapa model penggunaan dan mengurangi siklus hidup pengembangan secara keseluruhan. Tata kelola yang efektif dari kumpulan data denormalisasi juga signifikan karena dua alasan. Menerapkan data denormalisasi dapat membuat sejumlah besar kumpulan data redundan, yang bisa menjadi tantangan untuk dikelola dalam skala besar. Selain itu, kumpulan data ini bisa semakin sulit untuk digunakan kembali jika tidak dimodelkan dengan benar.

Desain portabilitas dan skalabilitas

Organisasi besar jarang memiliki semua aplikasi dan pengguna mereka pada satu platform data. Aplikasi dan penyimpanan data mereka biasanya didistribusikan di seluruh platform lokal dan

cloud lama, sehingga sulit bagi tim analitik untuk mencampur dan menggabungkan data. Kami menyarankan Anda menyimpan data berdasarkan karakteristik seperti domain, geografi, kasus penggunaan bisnis, dan sebagainya. Kontainerisasi ini meningkatkan portabilitas antara berbagai platform dan aplikasi dan mendukung konsumsi yang lebih efektif. Mensegmentasi data ke dalam wadah dan mengeksposnya APIs membantu Anda menskalakan arsitektur data dengan lebih mudah. Ini memungkinkan hybrid, aliran end-to-end data dan membantu aplikasi lokal dan berbasis cloud bekerja dengan mulus.

Unggul

Ketika arsitektur analitik modern berkembang dalam suatu organisasi, penting untuk mengelola perubahan itu dengan memperkenalkan konsep yang dapat digunakan kembali. Konsep-konsep ini meningkatkan daya tahan dan adopsi sambil menjaga biaya tetap terkendali. Beberapa konsep yang perlu dipertimbangkan dibahas di bagian berikut.

Rancang kerangka kerja yang dapat dikonfigurasi

Organizations sering membuat beberapa model yang kompleks untuk memenuhi kebutuhan bisnis mereka yang unik. Model-model ini memerlukan pembuatan beberapa pipa data dan fitur rekayasa. Seiring waktu, ini menciptakan redundansi yang signifikan dan meningkatkan biaya operasi. Membuat kerangka kerja yang menggabungkan satu set model dasar yang digerakkan oleh parameter dan dapat dikonfigurasi mengurangi waktu pengembangan dan biaya pengoperasian. Mesin analitik dapat mengimplementasikan model yang dapat dikonfigurasi ini untuk memberikan output yang diinginkan.

Berencana untuk membangun mesin analitik terpadu

Masalah bisnis unik dan sering membutuhkan teknologi khusus untuk memenuhi persyaratan, menghasilkan beberapa mesin analitik dalam suatu organisasi. Merancang dan mengembangkan antarmuka mesin analitik berbasis AI terpadu yang dapat mendukung berbagai paradigma pemrograman menyederhanakan penggunaan dan mengurangi biaya.

Mendefinisikan DataOps

Sebagian besar profesional data menghabiskan banyak waktu untuk melakukan operasi data seperti menemukan data yang tepat, mengubah, memodelkan, dan sebagainya. Memiliki operasi data tangkas (DataOps) dapat sangat meningkatkan arsitektur data dengan memecah silo insinyur data, ilmuwan data, pemilik data, dan analis. DataOps memungkinkan komunikasi yang lebih baik antar

tim, mengurangi waktu siklus, dan memastikan kualitas data yang tinggi. Arsitektur data dan analitik telah mengalami banyak transformasi dari waktu ke waktu karena perubahan kebutuhan bisnis dan kemajuan teknologi. Sebuah organisasi harus berusaha untuk mengembangkan, menerapkan, dan memelihara arsitektur data dan analitik yang berkembang dari waktu ke waktu dan mendukung bisnisnya.

Rekayasa data

Mengotomatiskan dan mengatur aliran data di seluruh organisasi Anda.

Gunakan metadata untuk mengotomatiskan pipeline yang memproses data mentah dan menghasilkan output yang dioptimalkan. Manfaatkan pagar pembatas arsitektur dan kontrol keamanan yang ada seperti yang didefinisikan di seluruh arsitektur platform AWS CAF dan kemampuan rekayasa platform, serta perspektif Operasi. Bekerja dengan tim pemberdayaan rekayasa platform untuk mengembangkan cetak biru yang dapat digunakan kembali untuk pola umum yang menyederhanakan penyebaran pipa.

Mulai

Menyebarkan danau data

Menetapkan kemampuan penyimpanan data dasar dengan menggunakan solusi penyimpanan yang sesuai untuk data terstruktur dan tidak terstruktur. Ini memungkinkan Anda untuk mengumpulkan dan menyimpan data dari berbagai sumber, dan membuat data dapat diakses untuk diproses dan dianalisis lebih lanjut. Penyimpanan data adalah komponen penting dari strategi rekayasa data. Arsitektur penyimpanan data yang dirancang dengan baik memungkinkan organisasi untuk menyimpan, mengelola, dan mengakses data mereka secara efisien dan hemat biaya. AWS menawarkan berbagai layanan penyimpanan data untuk memenuhi kebutuhan bisnis tertentu.

Misalnya, Anda dapat menetapkan kemampuan penyimpanan data dasar dengan menggunakan Amazon Simple Storage Service (Amazon S3) untuk penyimpanan objek, Amazon Relational Database Service (Amazon RDS)untuk database relasional, dan Amazon Redshift untuk pergudangan data. Layanan ini membantu Anda menyimpan data dengan aman dan hemat biaya, dan membuat data mudah diakses untuk diproses dan dianalisis lebih lanjut. Kami menyarankan Anda juga menerapkan praktik terbaik penyimpanan data, seperti partisi dan kompresi data, untuk meningkatkan kinerja dan mengurangi biaya.

Kembangkan pola konsumsi data

Untuk mengotomatiskan dan mengatur aliran data, buat proses konsumsi data untuk mengumpulkan data dari beragam sumber, termasuk database, file, dan. APIs Proses penyerapan data Anda harus mendukung kelincahan bisnis dan mempertimbangkan kontrol tata kelola.

Orkestrator harus mampu menjalankan layanan berbasis cloud dan menyediakan mekanisme penjadwalan otomatis. Ini harus menawarkan opsi untuk tautan bersyarat dan dependensi di antara tugas, bersama dengan kemampuan polling dan penanganan kesalahan. Selain itu, harus terintegrasi secara mulus dengan sistem peringatan dan pemantauan untuk memastikan bahwa jaringan pipa berjalan dengan lancar.

Beberapa mekanisme orkestrasi populer meliputi:

- Orkestrasi berbasis waktu memulai alur kerja pada interval rekursif dan pada frekuensi yang ditentukan.
- Orkestrasi berbasis peristiwa memulai alur kerja berdasarkan terjadinya peristiwa seperti pembuatan file atau permintaan API.
- Polling mengimplementasikan mekanisme di mana tugas atau alur kerja memanggil layanan (misalnya, melalui API) dan menunggu respons yang ditentukan sebelum melanjutkan ke langkah berikutnya.

Desain arsitektur modern menekankan memanfaatkan layanan terkelola yang menyederhanakan manajemen infrastruktur di cloud dan mengurangi beban pengembang dan tim infrastruktur. Pendekatan ini juga berlaku untuk rekayasa data. Kami menyarankan Anda menggunakan layanan terkelola jika berlaku untuk membangun jaringan pipa konsumsi data untuk mempercepat proses rekayasa data Anda. Dua contoh dari jenis layanan ini adalah Amazon Managed Workflows for Apache Airflow (Amazon MWAA) dan: AWS Step Functions

- Apache Airflow adalah alat orkestrasi populer untuk menulis, menjadwalkan, dan memantau alur kerja secara terprogram. AWS menawarkan Amazon Managed Workflows for Apache Airflow (Amazon MWAA) sebagai layanan terkelola yang memungkinkan pengembang untuk fokus membangun daripada mengelola infrastruktur untuk alat orkestrasi. Amazon MWAA memudahkan pembuatan alur kerja dengan menggunakan skrip Python. Grafik asiklik terarah (DAG) mewakili alur kerja sebagai kumpulan tugas dengan cara yang menunjukkan hubungan dan dependensi setiap tugas. Anda dapat memiliki DAGs sebanyak yang Anda inginkan, dan Apache Airflow akan menjalankannya sesuai dengan hubungan dan dependensi masing-masing tugas.
- AWS Step Functionsmembantu pengembang membangun alur kerja visual kode rendah untuk mengotomatiskan TI dan proses bisnis. Alur kerja yang Anda buat dengan Step Functions disebut mesin status, dan setiap langkah alur kerja Anda disebut status. Anda dapat menggunakan Step Functions untuk membuat alur kerja untuk penanganan kesalahan bawaan, penerusan parameter, pengaturan keamanan yang direkomendasikan, dan manajemen status. Ini mengurangi jumlah

kode yang harus Anda tulis dan pertahankan. Tugas melakukan pekerjaan dengan berkoordinasi dengan AWS layanan lain atau aplikasi yang Anda host baik di tempat atau di lingkungan cloud.

Mempercepat pemrosesan data

Pemrosesan data adalah langkah penting dalam memahami sejumlah besar data yang dikumpulkan oleh organisasi modern. Untuk memulai pemrosesan data, AWS menawarkan layanan terkelola seperti AWS Glue, yang menyediakan kemampuan ekstrak, transformasi, dan pemuatan (ETL) yang kuat. Organizations dapat menggunakan layanan ini untuk mulai memproses dan mengubah data mentah, termasuk pembersihan, normalisasi, dan agregasi data untuk mempersiapkannya untuk analisis.

Pemrosesan data dimulai dengan teknik sederhana seperti agregasi dan penyaringan untuk melakukan transformasi data awal. Seiring berkembangnya kebutuhan pemrosesan data, Anda dapat menerapkan proses ETL yang lebih canggih yang memungkinkan Anda mengekstrak data dari berbagai sumber, mengubahnya agar sesuai dengan kebutuhan spesifik Anda, dan memuatnya ke gudang data terpusat atau database untuk analisis terpadu. Pendekatan ini memastikan bahwa data akurat, lengkap, dan tersedia untuk analisis tepat waktu.

Dengan menggunakan layanan AWS terkelola untuk pemrosesan data, organisasi dapat memperoleh manfaat dari tingkat otomatisasi, skalabilitas, dan efektivitas biaya yang lebih tinggi. Layanan ini mengotomatiskan banyak tugas pemrosesan data rutin, seperti penemuan skema, pembuatan profil data, dan transformasi data, dan membebaskan sumber daya berharga untuk kegiatan yang lebih strategis. Selain itu, layanan ini menskalakan secara otomatis untuk mendukung volume data yang terus bertambah.

Menyediakan layanan visualisasi data

Temukan cara untuk membuat data tersedia bagi pengambil keputusan yang menggunakan visualisasi data untuk menafsirkan data secara bermakna dan cepat. Melalui visualisasi, Anda dapat menafsirkan pola dan meningkatkan keterlibatan di berbagai pemangku kepentingan, terlepas dari keterampilan teknis mereka. Platform yang baik memungkinkan tim rekayasa data untuk menyediakan sumber daya yang menyediakan visualisasi data dengan cepat dan dengan sedikit overhead. Anda juga dapat memberikan kemampuan swalayan dengan menggunakan alat yang dapat dengan mudah menanyakan penyimpanan data tanpa perlu keahlian teknik. Pertimbangkan untuk menggunakan perkakas bawaan yang dapat memberikan kecerdasan bisnis tanpa server melalui visual data dan dasbor interaktif, dan yang dapat menggunakan bahasa alami untuk menanyakan data back-end.

Maju

Menerapkan pemrosesan data mendekati waktu nyata

Pemrosesan data adalah komponen penting dari setiap pipa rekayasa data, yang memungkinkan organisasi untuk mengubah data mentah menjadi wawasan yang bermakna. Selain pemrosesan batch tradisional, pemrosesan data real-time telah menjadi semakin penting dalam lingkungan bisnis yang serba cepat saat ini. Pemrosesan data real-time memungkinkan organisasi untuk merespons peristiwa saat terjadi, dan meningkatkan pengambilan keputusan dan efisiensi operasional.

Validasi kualitas data

Kualitas data secara langsung berdampak pada keakuratan dan keandalan wawasan dan keputusan yang berasal dari data. Menerapkan validasi data dan proses pembersihan sangat penting untuk memastikan bahwa Anda menggunakan data berkualitas tinggi dan dapat dipercaya untuk analisis.

Validasi data melibatkan verifikasi keakuratan, kelengkapan, dan konsistensi data dengan memeriksanya terhadap aturan dan kriteria yang telah ditentukan. Ini membantu mengidentifikasi setiap perbedaan atau kesalahan dalam data, dan memastikan bahwa itu sesuai untuk tujuan. Pembersihan data melibatkan identifikasi dan koreksi ketidakakuratan, ketidakkonsistenan, atau duplikasi dalam data.

Dengan menerapkan proses dan alat kualitas data, organisasi dapat meningkatkan akurasi dan keandalan wawasan yang berasal dari data, menghasilkan pengambilan keputusan dan efisiensi operasional yang lebih baik. Ini tidak hanya meningkatkan kinerja organisasi tetapi juga meningkatkan kepercayaan dan kepercayaan pemangku kepentingan dalam data dan analisis yang dihasilkan.

Buktikan layanan transformasi data

Transformasi data menyiapkan data untuk analitik tingkat lanjut dan model pembelajaran mesin. Ini melibatkan penggunaan teknik seperti normalisasi data, pengayaan, dan deduplikasi untuk memastikan bahwa data bersih, konsisten, dan siap untuk analisis.

- Normalisasi data melibatkan pengorganisasian data ke dalam format standar, menghilangkan redundansi, dan memastikan bahwa data konsisten di berbagai sumber. Hal ini memudahkan untuk menganalisis dan membandingkan data dari berbagai sumber dan memungkinkan organisasi untuk mendapatkan pemahaman yang lebih komprehensif tentang operasi mereka.

- Pengayaan data melibatkan peningkatan data yang ada dengan informasi tambahan dari sumber eksternal seperti data demografis atau tren pasar. Ini memberikan wawasan berharga tentang perilaku pelanggan atau tren industri yang mungkin tidak terlihat dari sumber data internal saja.
- Deduplikasi melibatkan identifikasi dan penghapusan entri data duplikat, dan memastikan bahwa data akurat dan bebas dari kesalahan. Hal ini sangat penting ketika berhadapan dengan dataset besar, di mana bahkan sebagian kecil duplikasi mungkin condong hasil analisis.

Dengan menggunakan teknik transformasi data tingkat lanjut, organisasi memastikan bahwa data mereka berkualitas tinggi, akurat, dan siap untuk analisis yang lebih kompleks. Ini mengarah pada pengambilan keputusan yang lebih baik, peningkatan efisiensi operasional, dan keunggulan kompetitif di pasar.

Aktifkan demokratisasi data

Mempromosikan budaya demokratisasi data dengan membuat data dapat diakses, dimengerti, dan dapat digunakan untuk semua karyawan. Demokratisasi data membantu karyawan membuat keputusan berbasis data dan berkontribusi pada budaya berbasis data organisasi. Ini berarti memecah silo dan menciptakan budaya di mana data dibagikan dan digunakan oleh semua karyawan untuk mendorong pengambilan keputusan.

Secara keseluruhan, demokratisasi data adalah tentang menciptakan budaya di mana data dihargai, dapat diakses, dan dimengerti oleh semua orang di organisasi. Dengan memungkinkan demokratisasi data, organisasi menumbuhkan budaya berbasis data yang mendorong inovasi, meningkatkan pengambilan keputusan, dan pada akhirnya mengarah pada kesuksesan bisnis.

Unggul

Menyediakan orkestrasi berbasis UI

Untuk membangun organisasi yang gesit dan menggunakan pendekatan yang efektif, penting untuk merencanakan platform orkestrasi modern yang digunakan oleh sumber daya pengembangan dan operasi di seluruh lini bisnis. Tujuannya adalah untuk mengembangkan, menyebarkan, dan berbagi jalur data dan alur kerja tanpa bergantung pada satu tim, teknologi, atau model dukungan. Ini dicapai melalui kemampuan seperti orkestrasi berbasis UI. Fitur seperti drag-and-drop interaksi memungkinkan pengguna yang memiliki sedikit keahlian teknis untuk membangun DAGs dan menyatakan aliran data mesin. Komponen-komponen ini kemudian dapat menghasilkan kode yang dapat dieksekusi yang mengatur pipeline data.

DataOps membantu mengatasi kompleksitas manajemen data dan memastikan aliran data yang mulus di seluruh organisasi. Pendekatan berbasis metadata memastikan kualitas dan kepatuhan data sesuai dengan mandat organisasi Anda. Investasi dalam perangkat seperti layanan mikro, kontainerisasi, dan fungsi tanpa server meningkatkan skalabilitas dan kelincahan.

Mengandalkan tim rekayasa data untuk menghasilkan nilai dari data dan menyerahkan tugas day-to-day infrastruktur ke otomatisasi memungkinkan organisasi mencapai keunggulan dalam otomatisasi dan orkestrasi. Pemantauan dan pencatatan tugas manajemen aliran data yang mendekati waktu nyata mendukung tindakan remediasi segera dan meningkatkan kinerja dan keamanan pipa aliran data. Prinsip-prinsip ini membantu mencapai skalabilitas dan kinerja sambil memastikan model berbagi data yang aman, dan mengatur organisasi untuk sukses di masa depan.

Integrasikan DataOps

DataOps adalah pendekatan modern untuk rekayasa data yang menekankan integrasi proses pengembangan dan operasi untuk merampingkan pembuatan, pengujian, dan penyebaran pipa data. Untuk menerapkan praktik DataOps terbaik, organisasi menggunakan infrastruktur sebagai kode (IAc) dan alat integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD). Alat-alat ini mendukung pembuatan, pengujian, dan penerapan pipa otomatis, yang secara signifikan meningkatkan efisiensi dan mengurangi kesalahan. DataOps Tim bekerja dengan tim pemberdayaan rekayasa platform untuk membangun otomatisasi ini, sehingga setiap tim dapat fokus pada apa yang mereka lakukan terbaik.

Menerapkan DataOps metodologi membantu menumbuhkan lingkungan kolaboratif untuk insinyur data, ilmuwan data, dan pengguna bisnis, dan memungkinkan pengembangan, penyebaran, dan pemantauan jalur data dan solusi analitik yang cepat. Pendekatan ini memberikan komunikasi dan kolaborasi yang lebih mulus di seluruh tim, yang mengarah pada inovasi yang lebih cepat dan hasil yang lebih baik.

Untuk memanfaatkan sepenuhnya manfaat DataOps, penting untuk merampingkan proses rekayasa data. Ini dicapai dengan menggunakan praktik terbaik dari tim rekayasa platform, termasuk tinjauan kode, integrasi berkelanjutan, dan pengujian otomatis. Dengan menerapkan praktik-praktik ini, organisasi memastikan bahwa jaringan data dapat diandalkan, terukur, dan aman, dan memenuhi kebutuhan pemangku kepentingan bisnis dan teknis.

Penyediaan dan orkestrasi

Membuat, mengelola, dan mendistribusikan katalog produk cloud yang disetujui kepada pengguna.

Penyediaan infrastruktur secara konsisten, terukur, dan berulang menjadi lebih menantang seiring pertumbuhan organisasi Anda. Penyediaan dan orkestrasi yang efisien membantu Anda mencapai tata kelola yang konsisten dan memenuhi persyaratan kepatuhan Anda sekaligus memungkinkan pengguna untuk menerapkan hanya produk cloud yang disetujui.

Menggunakan kembali produk yang telah disetujui sebelumnya di organisasi Anda memungkinkan pengembang Anda untuk membangun aplikasi lebih cepat dan lebih konsisten sambil memenuhi persyaratan keamanan dan tata kelola organisasi Anda.

Mulai

Menyebarkan model hub-and-spoke katalog

Aset perangkat lunak yang dikelola dalam katalog layanan sebagai portofolio dibagikan dengan pengguna dalam satu atau beberapa akun dalam suatu hub-and-spoke pola. Anda dapat menggunakan pasar pribadi dan penyedia layanan pribadi untuk mengkurasi bermacam-macam solusi pihak ketiga dan mendistribusikannya dengan templat infrastruktur sebagai kode (IAc) Anda.

Untuk memungkinkan pembangun Anda mengonsumsi produk yang telah disetujui sebelumnya, tentukan proses untuk meninjau, menyetujui, dan mempublikasikan produk ini kepada pengguna Anda. Mulailah dengan merancang dan menerapkan repositori yang dikelola secara terpusat yang berisi produk yang telah disetujui sebelumnya ini. Rancang sistem yang memberikan akses ke lisensi dan produk di repositori ini ketika pengguna di organisasi Anda perlu mengkonsumsi setiap produk.

Izinkan pembangun di organisasi Anda untuk mengirimkan produk untuk disetujui ke mekanisme penerbitan, sehingga produk ini tersedia untuk semua pengguna di organisasi Anda setelah disetujui.

Templat kurasi untuk digunakan kembali

Ketika Anda telah mengkodifikasi template IAc untuk solusi Anda dan mendefinisikan hub-and-spoke model Anda, Anda harus menentukan dua kategori template untuk setiap akun spoke: disediakan/diberlakukan dan tersedia untuk dikonsumsi. Template yang disediakan/diberlakukan disediakan langsung dari akun manajemen ke setiap akun anggota sebagai kemampuan dasar. Templat yang

tersedia untuk dikonsumsi tersedia bagi pembangun untuk ditelusuri dan disediakan dengan cara swalayan.

Terapkan parameter default untuk digunakan kembali

Terapkan templat IAC yang menyertakan parameter default yang dapat dipilih oleh pembangun Anda. Hal ini memungkinkan pembangun untuk menyelaraskan dengan tata kelola tanpa harus mengevaluasi rincian setiap parameter, dan mencegah mereka membuat pilihan yang salah. Pendekatan ini hanya mengekspos apa yang diperlukan untuk pengaturan. Misalnya, AWS Service Catalogmenerapkan pendekatan ini dengan kemampuan kendala yang mengontrol aturan yang diterapkan pada produk dalam portofolio tertentu. Kustomisasi ini telah dikonfigurasi sebelumnya saat tim pembuat menggunakan penyediaan templat swalayan.

Menetapkan proses persetujuan

Pengguna harus dapat mengirimkan permintaan untuk mengakses produk yang tidak disetujui jika mereka memiliki pembenaran bisnis untuk menggunakan produk tersebut. Buat sistem notifikasi yang memberi tahu pengguna saat pembaruan untuk produk yang mereka gunakan tersedia, sehingga mereka dapat mematuhi pembaruan keamanan terbaru.

Tetapkan alur kerja bagi pembangun untuk mengirimkan produk baru untuk ditinjau melalui portal swalayan. Pembangun dapat menggunakan portal untuk menentukan audiens untuk produk dan untuk mengidentifikasi kelompok pengguna yang harus memiliki akses ke produk. Untuk setiap pengiriman, gunakan proses yang Anda tentukan untuk meninjau, menyetujui, dan mempublikasikan produk ke portal swalayan.

Maju

Buat portal swalayan

Buat portal swalayan untuk mendistribusikan, menelusuri, dan menggunakan produk cloud yang disetujui. Pengguna dalam organisasi dapat menggunakan portal ini untuk mencari produk yang mereka butuhkan untuk membangun infrastruktur mereka dan untuk menyebarkan aplikasi ke lingkungan mereka. Tetapkan batasan izin bagi pengguna yang memiliki akses ke produk di portal, dan tetapkan batasan berapa kali pengguna dapat mengkonsumsi produk berlisensi. Tentukan kumpulan sumber daya dasar yang dapat langsung disediakan atau tersedia sebagai model layanan mandiri di setiap akun spoke Anda, karena akun dibuat dengan menggunakan solusi seperti Kustomisasi untuk. AWS Control Tower

Aktifkan pasar pribadi

Pasar pribadi menyediakan katalog produk yang dibeli (perangkat lunak, data, dan layanan profesional) dan diimplementasikan dalam hub-and-spoke pola (dengan satu akun manajemen dan beberapa akun anggota) sehingga akun spoke hanya dapat berlangganan perangkat lunak yang disetujui. Tata kelola produk ini membantu mengontrol biaya perangkat lunak dan merampingkan ulasan hukum dan kontrak. Buat pasar pribadi di tingkat akun manajemen untuk berfungsi sebagai hub utama.

Kelola hak

Aktifkan kontrol yang hanya mengizinkan pengguna dan beban kerja yang berwenang untuk menggunakan lisensi dalam batas yang ditentukan vendor. Ini membantu mengurangi risiko audit yang mahal dan penyetelan lisensi yang tidak terduga.

Unggul

Integrasi dengan sistem pengadaan

Lengkapi proses pengadaan Anda yang ada dengan mengintegrasikannya ke dalam. AWS Marketplace Ini dilakukan dengan memperluas sistem pengadaan Anda (Coupa atau SAP Ariba) ke pasar pribadi sehingga pengguna Anda dapat mengikuti proses pengadaan dan persetujuan yang ada untuk mendapatkan perangkat lunak. Buat izin yang dikelola IAM yang sesuai, gunakan AWS Marketplace untuk menghasilkan informasi yang diperlukan untuk mengonfigurasi solusi pengadaan Anda, dan konfigurasikan solusi pengadaan Anda untuk menyelesaikan integrasi. Misalnya, Anda dapat mengatur punchout, melampirkan pesanan pembelian ke AWS faktur Anda, dan kemudian menyelaraskan proses pengadaan Anda untuk menggunakan solusi penyediaan standar.

Memungkinkan pembangun Anda untuk mengakses produk yang telah disetujui sebelumnya melalui API internal, sehingga pengguna dapat memasukkan produk ke dalam aplikasi mereka atau membuat portal pribadi mereka sendiri untuk tim mereka untuk mengkonsumsi produk. Integrasikan proses pengiriman dan publikasi untuk membuat produk baru, dan memungkinkan pengguna untuk meminta lisensi baru dan akses ke produk melalui. APIs

Integrasikan dengan alat ITSM Anda

Jika berlaku, sambungkan dengan alat manajemen layanan TI (ITSM) dan otomatiskan pembaruan apa pun ke database manajemen konfigurasi (CMDB) Anda. Tetapkan proses dan mekanisme

untuk mengevaluasi produk yang digunakan organisasi Anda. Menetapkan mekanisme untuk menginformasikan pengguna produk yang telah disetujui sebelumnya yang perlu mereka perbarui untuk kepatuhan. Gunakan alat ITSM Anda untuk menganalisis lingkungan Anda dan untuk mendorong pembaruan keamanan dan kepatuhan terhadap produk di seluruh organisasi Anda saat pembaruan penting diperlukan.

Menerapkan manajemen siklus hidup dan sistem distribusi versi

Pertahankan versi templat IAC, dan versi layanan yang disediakan dari templat, sepanjang siklus pengembangan mereka. Anda dapat menggunakan hub-and-spoke model yang Anda terapkan untuk katalog untuk menentukan apakah pembaruan paksa diperlukan pada tingkat spoke (misalnya, jika versi bersamaan tersedia untuk penyediaan layanan mandiri), dan versi mana yang perlu ditandai untuk usang. Menggunakan hub-and-spoke katalog juga membantu mengelola audit dan distribusi versi baru sesuai kebutuhan.

Pengembangan aplikasi modern

Bangun aplikasi cloud-native yang dirancang dengan baik.

Praktik pengembangan aplikasi modern sangat penting bagi organisasi untuk membangun aplikasi cloud-native yang dirancang dengan baik dan tetap kompetitif. Bisnis dapat menggunakan teknologi cloud-native seperti kontainer dan komputasi tanpa server untuk membuat aplikasi yang skalabel dan gesit yang beradaptasi dengan perubahan permintaan pasar. Teknologi ini memungkinkan organisasi untuk mengoptimalkan pemanfaatan sumber daya, mengurangi biaya, dan meningkatkan kinerja aplikasi mereka.

Saat Anda merancang aplikasi modern Anda, kembangkan solusi tangkas untuk operasi dan pengembangan. Aplikasi modern secara otomatis bereaksi terhadap perubahan permintaan pelanggan dan tahan terhadap kegagalan. Insinyur dapat mengembangkan dan menerapkan perubahan dengan cepat dan memantau kinerja aplikasi. Aplikasi modern dirancang untuk menyembuhkan diri sendiri dan mampu menskalakan ke tingkat lalu lintas besar atau kecil, termasuk tidak ada lalu lintas dengan biaya nol, bila diperlukan.

Membangun aplikasi cloud-native yang dirancang dengan baik membutuhkan pemahaman mendalam tentang teknologi yang mendasari dan praktik terbaiknya. Organizations harus mengadopsi arsitektur layanan mikro dan merancang aplikasinya agar modular dan digabungkan secara longgar, memungkinkan penyebaran dan skalabilitas independen. Pendekatan ini memungkinkan organisasi untuk memecah aplikasi mereka menjadi komponen yang lebih kecil dan lebih mudah dikelola yang dikembangkan, diuji, dan digunakan dengan cepat dan mandiri.

Mulai

Jelajahi pendekatan modern

Mulailah dengan menyelidiki kontainer, teknologi tanpa server, dan pendekatan lain yang memungkinkan pengembangan layanan mikro, yang meningkatkan efisiensi sumber daya, membantu meningkatkan keamanan, dan meminimalkan biaya infrastruktur. Pilih untuk memodernisasi aplikasi pembeda dan perusahaan Anda yang ada untuk meningkatkan efisiensi dan memaksimalkan nilai investasi Anda yang ada. Pertimbangkan replatforming (mentransisikan kontainer, database, atau pialang pesan yang dikelola sendiri ke layanan cloud terkelola) dan refactoring (mengembangkan kembali aplikasi Anda untuk mengadopsi arsitektur cloud-native) berdasarkan pengambilan keputusan berdasarkan nilai.

Saat Anda memperbarui aplikasi berbasis cloud yang ada, pendekatan yang berhasil melibatkan penggunaan pola ara pencekik untuk secara progresif menguraikan arsitektur Anda menjadi layanan mikro. Prosedur ini membantu dalam mengadopsi metodologi aplikasi kontemporer, sehingga Anda dapat menyadari manfaat yang melekat dan menunjukkan nilainya kepada organisasi yang lebih besar. Pertimbangkan untuk membangun aplikasi Anda sebagai layanan mikro berbeda yang memanfaatkan arsitektur berbasis peristiwa jika berlaku. Pastikan arsitektur Anda memperhitungkan kuota layanan dan sumber daya fisik yang tidak dapat diubah untuk menghindari pengaruh kinerja atau keandalan beban kerja.

Mengadopsi kemampuan komputasi cloud-native

Kemampuan komputasi cloud-native adalah kunci untuk pengembangan aplikasi modern. Pendekatan ini mengharuskan organisasi untuk mempertimbangkan bagaimana mereka ingin unit komputasi mereka di-host dan mengidentifikasi opsi terbaik untuk setiap kasus penggunaan atau layanan. Misalnya, AWS Lambdamenawarkan mekanisme tanpa server untuk menjalankan kode aplikasi Anda dan memainkan peran kunci dalam arsitektur berbasis peristiwa. Fungsi Lambda diluncurkan sesuai permintaan dan dijalankan secara paralel hingga konkurensi maksimum yang ditentukan, sehingga mereka dapat menskalakan untuk melakukan berbagai tugas.

Gunakan kontainerisasi

Dalam pengembangan perangkat lunak modern, mengelola aplikasi dan dependensinya telah menjadi tugas yang semakin kompleks, terutama ketika Anda mempertimbangkan perlunya menjaga konsistensi di berbagai lingkungan. Untuk mengatasi tantangan ini, teknologi containerization seperti Docker telah muncul sebagai solusi efektif untuk aplikasi pengemasan dan dependensinya. Container memastikan penerapan yang konsisten dan dapat direproduksi terlepas dari lingkungan runtime aplikasi Anda, sehingga pengembangan di lingkungan lokal Anda berperilaku dengan cara yang sama seperti pengembangan produksi di lingkungan cloud. Pendekatan ini mengurangi kesalahan yang mungkin disebabkan oleh ketidakcocokan dalam lingkungan atau konfigurasinya.

Gunakan database modern

Saat Anda menggunakan database modern, setiap layanan mikro dalam aplikasi Anda dapat menggunakan basis data yang dibuat khusus yang memenuhi persyaratannya, yang meningkatkan kelincahan dan kinerja sekaligus menurunkan biaya. Misalnya, satu layanan mikro mungkin menggunakan database NoSQL untuk mencapai throughput tinggi saat menyimpan data sesi, layanan mikro lain mungkin menggunakan database relasional untuk melakukan gabungan tabel

yang kompleks, dan layanan mikro lain mungkin menggunakan database buku besar kuantum untuk melacak perubahan pada blockchain.

Database modern menawarkan skalabilitas dan fleksibilitas. Mereka juga membantu memberikan keamanan, kepatuhan, dan keandalan yang lebih baik daripada database tradisional. Mereka memungkinkan organisasi untuk menyimpan dan mengelola data mereka secara lebih efisien dan memastikan bahwa aplikasi dapat mengakses data yang tepat pada waktu yang tepat, yang mengarah ke kinerja dan pengalaman pengguna yang lebih baik.

Migrasi ke database modern adalah komponen penting dari pengembangan aplikasi modern. Dengan menggunakan solusi penyimpanan data yang tepat, organisasi dapat mengoptimalkan kemampuan manajemen data mereka dan memberikan aplikasi yang lebih efisien dan andal. Dengan membuat setiap layanan mikro independen dan memilih teknologi yang tepat untuk setiap layanan mikro, organisasi dapat lebih mengoptimalkan kemampuan data mereka untuk mencapai efisiensi dan skalabilitas maksimum sambil meminimalkan biaya.

Maju

Optimalkan arsitektur modern Anda

Untuk mencapai pengoptimalan lebih lanjut, perbaiki implementasi teknologi tanpa server Anda dan kembangkan arsitektur yang dapat diskalakan dan digunakan secara independen dengan menggunakan layanan seperti AWS Amazon API Gateway dan.AWS Lambda Menerapkan penemuan layanan dengan menggunakan Amazon Route 53 dan AWS Cloud Mapuntuk memastikan komunikasi yang mulus antar komponen.

Mengadopsi versi API, caching, dan pembatasan kecepatan untuk menjaga kompatibilitas dan kinerja di berbagai versi aplikasi. Meningkatkan keamanan dengan AWS Identity and Access Management (IAM) dan kebijakan sumber daya. Ini membantu memastikan bahwa infrastruktur Anda dilindungi dan akses hanya diberikan kepada entitas yang berwenang.

Jika memungkinkan, gunakan layanan tanpa server untuk menjalankan kontainer tanpa harus mengelola infrastruktur yang mendasarinya. Hal ini memungkinkan Anda untuk fokus pada pengembangan aplikasi inti Anda dan memungkinkan untuk manajemen sumber daya yang lebih baik dan kinerja. Ini juga membantu Anda memanfaatkan sepenuhnya manfaat skalabilitas, fleksibilitas, dan efisiensi biaya.

Dengan menyelam lebih dalam ke seluk-beluk arsitektur tanpa server dan menggabungkan praktik-praktik lanjutan ini, organisasi dapat mengungkap peluang untuk perbaikan dan penyempurnaan, dan

pada akhirnya memaksimalkan potensi aplikasi cloud-native mereka. Pengejaran ini memfasilitasi adopsi pola aplikasi yang lebih canggih yang semakin meningkatkan pengalaman pengguna secara keseluruhan. Ini juga memberdayakan organisasi untuk menjadi lebih gesit dan efisien dalam proses pengembangan perangkat lunak mereka.

Gunakan teknologi mesh layanan

Ketika organisasi semakin mengadopsi arsitektur layanan mikro untuk membangun dan menyebarkan aplikasi, mengelola kompleksitas, keamanan, dan komunikasi di antara layanan ini menjadi penting. Teknologi mesh layanan seperti Istio, Linkerd, atau Konsul memainkan peran penting dalam membantu meningkatkan keamanan, observabilitas, dan keandalan layanan mikro.

Pastikan visibilitas dan keterlacakan

Praktik modern memberikan visibilitas dan keterlacakan yang lebih besar dalam proses pengembangan, dan membuatnya lebih mudah untuk mematuhi standar industri dan praktik terbaik. Visibilitas dan pemantauan sangat penting untuk pengembangan aplikasi modern. Menerapkan solusi pemantauan dan pencatatan untuk memberikan wawasan berharga tentang kinerja aplikasi memungkinkan organisasi mengidentifikasi area untuk perbaikan dan mengoptimalkan aplikasi mereka. Kami menyarankan Anda bekerja dengan tim teknik platform Anda untuk memastikan bahwa alat tersedia untuk memberikan end-to-end visibilitas dan pemantauan kesalahan, kinerja, dan kepatuhan aplikasi, sehingga Anda dapat mendeteksi, mendiagnosis, dan menyelesaikan masalah dengan cepat.

Unggul

Merangkul layanan mikro

Bagi banyak organisasi, pengembangan aplikasi modern identik dengan kesuksesan bisnis. Layanan mikro adalah jantung dari transformasi ini, dan organisasi dapat mengambil manfaat dari merangkul pola arsitektur yang kuat ini.

Microservices menawarkan arsitektur aplikasi yang sangat skalabel, tangguh, dan gesit. Dengan memecah aplikasi menjadi layanan kecil yang dapat diterapkan secara independen, organisasi dapat memilih untuk melakukan iterasi dengan cepat pada komponen tertentu tanpa memengaruhi bagian lain dari aplikasi. Pola ketahanan tingkat lanjut, seperti pemutus sirkuit dan sekat, memainkan peran penting dalam memastikan ketersediaan aplikasi ini yang tinggi.

Pemutus sirkuit bertindak sebagai mekanisme keamanan yang mencegah kegagalan berjenjang dengan menghentikan sementara atau mengalihkan komunikasi dari layanan yang tidak sehat, sehingga dapat pulih. Sekat mengisolasi sumber daya dan membatasi ruang lingkup dampak kegagalan potensial. Bersama-sama, pola-pola ini menciptakan arsitektur yang kuat yang tahan terhadap gangguan tak terduga dan mempertahankan kinerja yang optimal.

Aspek penting lain dari penerapan layanan mikro adalah adopsi prinsip-prinsip desain berbasis domain (DDD). DDD berfokus pada menciptakan pemahaman bersama tentang domain bisnis dan menerjemahkannya ke dalam desain perangkat lunak yang terstruktur dengan baik. Pendekatan ini mengarah pada layanan mikro yang lebih kohesif dan dapat dipelihara, dan memastikan bahwa aplikasi berkembang sejalan dengan kebutuhan organisasi.

Mengoptimalkan komunikasi antar layanan juga penting dalam aplikasi berbasis layanan mikro. Dengan menerapkan protokol canggih seperti gRPC atau GraphQL, organisasi dapat secara signifikan meningkatkan efisiensi komunikasi antar layanan. Protokol ini menawarkan kemampuan seperti keamanan tipe, latensi rendah, dan fleksibilitas, yang membantu meningkatkan kinerja dan pemeliharaan aplikasi secara keseluruhan.

Sebuah organisasi yang mengadopsi layanan mikro menyediakan lingkungan yang mendorong inovasi, kelincahan, dan kolaborasi. Tim pengembangan biasanya diatur berdasarkan kemampuan bisnis dan memiliki fokus yang kuat pada praktik integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD). Mereka diberdayakan untuk membuat keputusan, bereksperimen, dan mengulangi dengan cepat, dan mereka merangkul budaya tanggung jawab dan akuntabilitas bersama.

Integrasi berkelanjutan dan pengiriman berkelanjutan

Berevolusi dan meningkatkan aplikasi dan layanan lebih cepat daripada organisasi yang menggunakan pengembangan perangkat lunak tradisional dan proses manajemen infrastruktur.

Mengadopsi DevOpspraktik dengan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) promotes a streamlined, automated, and efficient process for building, testing, and deploying applications. CI/CDmemungkinkan pengiriman perangkat lunak yang cepat, mengurangi risiko kesalahan penerapan, dan memastikan bahwa aplikasi selalu up to date dengan fitur terbaru dan perbaikan bug. Tujuan utamanya adalah untuk mengembangkan dan meningkatkan aplikasi dan layanan dengan kecepatan yang lebih cepat dengan berevolusi dari penggunaan pengembangan perangkat lunak tradisional dan proses manajemen infrastruktur.

Mulai

Mengadopsi manajemen komponen perangkat lunak

Manajemen komponen perangkat lunak adalah praktik mengelola semua komponen individu yang digunakan untuk membangun perangkat lunak, termasuk perpustakaan, kerangka kerja, repositori kode sumber, modul, artefak, dan dependensi pihak ketiga. Kami menyarankan Anda menggunakan sistem kontrol versi seperti Git atau Apache Subversion untuk mengelola kode sumber, mengaktifkan kolaborasi, dan mempertahankan riwayat perubahan kode. Anda dapat memantau perubahan dan peristiwa di repositori untuk mengotomatiskan proses, membuat pipeline, mengelola kode, dan mengintegrasikan alur kerja Anda dengan layanan tambahan sesuai kebutuhan.

Buat CI/CD saluran pipa

CI/CD pipelines are sets of automated instructions that are initiated by changes committed to the version control system. They typically include instructions for building the application, running automated tests, and deploying code to a specific environment. You can set up an automated CI/CDpipeline dengan menggunakan alat-alat seperti AWS CodePipeline, Jenkins, GitLab, atau CircleCI. Anda juga dapat mengaturnya secara langsung di sistem kontrol versi yang mendukung pembuatan pipa.

Mulailah dengan pipa minimum yang layak untuk integrasi berkelanjutan, dan kemudian transisi ke pipa pengiriman berkelanjutan yang mencakup lebih banyak tindakan dan tahapan. Perlakukan konfigurasi pengiriman berkelanjutan Anda sebagai kode. Anda dapat menggunakan beberapa

saluran pipa yang berbeda untuk setiap cabang dan tim, jadi pikirkan variabel konfigurasi mana yang perlu Anda atur dan cara terbaik untuk mendukung tim yang akan menggunakan saluran pipa.

Pertimbangkan jendela penerapan – hari dan waktu mana Anda ingin menerapkan kode Anda. Pertimbangkan jam permintaan rendah sistem Anda, jadi jika Anda harus memutar kembali, itu akan berdampak paling kecil pada pelanggan Anda. Praktik terbaik lainnya termasuk menghindari penerapan pada hari Jumat dan menerapkan pembekuan kode selama tanggal puncak tinggi atau sebelum hari libur. Pertimbangkan untuk mendefinisikan aturan tentang penerapan kode saat pembuat komit tidak tersedia (misalnya, saat liburan). Ingatlah bahwa penerapan gagal dan Anda mungkin perlu mengandalkan bantuan eksternal. Evaluasi metode penyebaran yang berbeda seperti di tempat, bergulir, tidak dapat diubah, dan penerapan. blue/green Pertimbangkan untuk menggunakan layanan yang dikelola sepenuhnya untuk alur kerja pengiriman berkelanjutan untuk meningkatkan ketersediaan dan keamanan sambil meminimalkan kompleksitas dan manajemen.

Menyebarkan pengujian otomatis

Praktik modern merekomendasikan pergeseran ke kiri (memindahkan pengujian lebih dekat ke pengembang dan ke IDE, dan lebih awal dalam siklus hidup) untuk mendeteksi dan memulihkan masalah sebelum mereka berkomitmen ke repositori dan memulai pipeline. Praktik ini melibatkan loop umpan balik cepat dengan pengembang, karena kesalahan terdeteksi saat pengembang melakukan pengkodean. Pergeseran ke kiri dikaitkan dengan biaya yang lebih rendah, karena pengujian tidak memerlukan jaringan pipa yang berjalan, yang dapat menghasilkan umpan balik asinkron dan biaya operasional yang lebih tinggi.

Pengujian otomatis menangkap kesalahan di awal proses pengembangan, dan mencakup pengujian unit, tes integrasi, dan pengujian fungsional. Kami menyarankan Anda mendorong pengembang untuk menggunakan alat untuk membuat pengujian unit sedini mungkin dan menjalankannya sebelum mendorong kode ke repositori pusat. Selain itu, pastikan bahwa proses otomatis Anda mencakup analisis kode statis, pembandingan kinerja, dan pengujian aplikasi keamanan.

Buat dokumentasi

Selain menerapkan CI/CD pipeline untuk merampingkan alur kerja pengembangan, Anda harus memelihara dokumentasi yang jelas dan komprehensif untuk memastikan efektivitas, pemeliharaan, dan skalabilitas pipeline yang berkelanjutan. Dokumentasi adalah aspek penting dari pipa CI/CD, karena memberikan tim pengembangan pemahaman yang jelas tentang desain, komponen, dan proses pipa. Saat Anda membuat dokumentasi, mulailah dengan ikhtisar pipeline, jelaskan arsitektur dan pengorbanan desain, jelaskan alat dan teknologi yang sedang digunakan, tentukan konfigurasi

dan pengaturan awal, uraikan langkah-langkah keamanan dan kontrol akses, dan sertakan informasi pemecahan masalah dan pemeliharaan.

Gunakan infrastruktur sebagai kode

Gunakan alat seperti Terraform, Ansible, atau AWS CloudFormationuntuk mengelola infrastruktur dan untuk memastikan lingkungan yang konsisten dan dapat direproduksi. Perlakukan infrastruktur Anda sebagai kode, pastikan Anda melacak perubahan infrastruktur, dan hindari membuat perubahan langsung di konsol. Tentukan semua infrastruktur–termasuk penyediaan basis data–sebagai kode dan terapkan perubahan ini dengan menggunakan pipeline. Pertimbangkan untuk menjalankan integrasi database sebagai kode dalam jaringan pipa dengan sebagian kecil data produksi yang disanitasi. Jika memungkinkan, buat perubahan dan lacak perubahan dalam kode tersebut.

Seperti halnya kode perangkat lunak, ikuti praktik terbaik berikut untuk kode infrastruktur Anda:

- Gunakan kontrol versi.
- Manfaatkan sistem pelacakan bug dan tiket.
- Mintalah rekan-rekan meninjau perubahan sebelum menerapkannya.
- Menetapkan pola dan desain kode infrastruktur.
- Uji perubahan infrastruktur.

Menyimpan dan melacak metrik standar

Untuk mempertahankan tingkat kinerja yang tinggi, kembangkan dan lacak metrik utama untuk memahami dampak kesehatan dan bisnis dari jaringan pipa Anda, termasuk:

- Membangun frekuensi. Jumlah build menawarkan wawasan tentang produktivitas tim Anda dan kompleksitas perubahan.
- Frekuensi penyebaran. Penerapan reguler menunjukkan proses pengembangan yang sehat dan gesit.
- Lead time untuk perubahan. Mengukur waktu rata-rata perubahan untuk mencapai produksi dapat membantu Anda mengidentifikasi kemacetan dalam proses penerapan Anda.
- Berarti waktu melalui pipa. Waktu rata-rata dari tahap pipeline awal hingga setiap tahap berikutnya dapat membantu mengoptimalkan alur kerja Anda.
- Volume perubahan produksi. Melacak jumlah perubahan yang mencapai produksi dapat memberikan wawasan tentang stabilitas lingkungan produksi Anda.

- Membangun waktu. Waktu pembuatan rata-rata dapat menunjukkan potensi masalah dalam basis kode atau infrastruktur.

Maju

Gunakan manajemen konfigurasi

Alat manajemen konfigurasi memainkan peran penting dalam mengotomatiskan penyebaran, konfigurasi, dan manajemen perangkat lunak dan infrastruktur. Mereka menyediakan pendekatan sistematis untuk menangani perubahan dan mempertahankan keadaan infrastruktur, perangkat lunak, dan konfigurasi yang diinginkan di berbagai lingkungan. Alat-alat ini memungkinkan pengembang untuk menentukan keadaan yang diinginkan dari suatu sistem dengan menggunakan bahasa deklaratif atau imperatif. Alat manajemen konfigurasi kemudian mengotomatiskan proses penerapan konfigurasi ini ke sistem target, memastikan konsistensi dan pengulangan.

Gunakan alat manajemen konfigurasi untuk mengotomatiskan penyebaran, konfigurasi, dan pengelolaan perangkat lunak dan infrastruktur. AWS Systems Manager State Manager adalah layanan manajemen konfigurasi yang aman dan terukur yang mengotomatiskan proses menjaga node terkelola dan AWS sumber daya lainnya dalam keadaan yang Anda tentukan.

Integrasikan pemantauan dan pencatatan

Mengintegrasikan solusi pemantauan dan pencatatan ke dalam saluran CD menawarkan banyak manfaat bagi tim pengembangan dan untuk proses pengembangan perangkat lunak secara keseluruhan. Solusi ini dapat memberikan wawasan real-time tentang kinerja aplikasi, memungkinkan identifikasi dan penyelesaian masalah yang lebih cepat, dan mempromosikan peningkatan berkelanjutan untuk membantu memastikan bahwa aplikasi tetap andal, berkinerja, dan dapat diskalakan sepanjang siklus hidupnya. Berinvestasi dalam solusi pemantauan dan pencatatan adalah aspek kunci untuk mempertahankan saluran CD yang kuat dan efisien, dan pada akhirnya berkontribusi pada keberhasilan pengiriman perangkat lunak berkualitas tinggi.

Buat irama untuk penggabungan

Komit atau gabungkan perubahan kode ke cabang utama (trunk atau main) setidaknya sekali setiap hari atau, idealnya, beberapa kali sehari setelah setiap tugas. Irama ini mengarah ke beberapa pemanggilan pipa harian. Model alur kerja percabangan berbasis tarik selaras dengan pendekatan ini. Gunakan bendera fitur, peluncuran gelap, dan teknik serupa untuk menyesuaikan fitur yang digunakan pelanggan Anda.

Tangkap perilaku pasca-penerapan

Setelah penerapan, tangkap perilaku produksi dengan menggunakan pengujian sintetis otomatis dan sinkronisasi hasil dengan pipeline pengiriman berkelanjutan untuk memastikan bahwa tindakan korektif terjadi segera. Prioritas utama bagi pengembang harus memperbaiki kesalahan yang ditemukan di pipeline sesegera mungkin, melakukan perubahan kode ke repositori kode sumber, dan memverifikasi resolusi kesalahan dalam pipeline.

Praktik pasca-penerapan terbaik termasuk mengamati indikator kinerja utama yang paling penting (KPIs) dan memvalidasi bahwa tidak ada kesalahan dalam lingkungan produksi. Otomatiskan penanganan kesalahan dan evaluasi pasca-penerapan KPIs untuk mengukur dampak rilis Anda. Secara otomatis menghasilkan metrik kecepatan, keamanan, dan stabilitas yang dapat digunakan pengembang untuk melakukan perbaikan. Untuk informasi selengkapnya, lihat solusi DevOps Monitoring Dashboard di AWS.

Unggul

Mengadopsi praktik dan teknologi mutakhir untuk kinerja yang optimal. Terus menyempurnakan CI/CD proses Anda membantu Anda meningkatkan kualitas perangkat lunak, mengurangi waktu ke pasar, dan meningkatkan kelincahan. Teknik dan alat baru terus muncul, yang membuatnya penting bagi organisasi Anda untuk tetap mendapat informasi dan beradaptasi untuk mempertahankan keunggulan kompetitif.

Untuk tetap adaptif, pertimbangkan hal berikut:

- Tentukan semuanya sebagai kode, termasuk aplikasi, konfigurasi, infrastruktur, data, AWS akun, dan organisasi, pipeline penerapan, jaringan, serta kontrol keamanan dan kepatuhan.
- Buat pipeline penerapan yang sesuai untuk menghitung gambar, layanan bersama, dan aplikasi.
- Pertimbangkan GitOps model di mana permintaan berbasis tarik memulai alur kerja untuk menerapkan perubahan dengan membandingkan status infrastruktur yang ada dengan status yang diinginkan, seperti yang dijelaskan dalam kode.
- Pertimbangkan untuk menggunakan saluran pipa CD untuk menyebarkan pembelajaran mesin (ML), data, Internet of Things (IoT), dan beban kerja lainnya.
- Tanda tangani semua artefak build secara digital dan simpan di repositori yang aman.
- Lacak sumber perangkat lunak dengan secara otomatis menghasilkan tagihan materi perangkat lunak yang membuat catatan semua artefak berversi dan ditandatangani secara digital yang digunakan untuk pelanggan.

- Setelah Anda menghilangkan semua aktivitas manual dalam proses pengiriman perangkat lunak, hapus papan peninjau manual.

Untuk aplikasi dan layanan yang telah mengotomatiskan seluruh proses pengiriman perangkat lunak mereka, pertimbangkan penerapan berkelanjutan di mana tim menerapkan perubahan yang melewati semua pemeriksaan dalam pipeline kepada pelanggan dalam produksi. Untuk visualisasi, lihat diagram pertama di Apa itu Pengiriman Berkelanjutan? di situs AWS web.

Integrasikan AI/ML teknologi

Integrasi teknologi kecerdasan buatan (AI) dan pembelajaran mesin (ML) ke dalam CI/CD jaringan pipa menawarkan beberapa manfaat, termasuk yang berikut:

- Pembuatan uji otomatis
- Prioritas tes cerdas
- Analisis prediktif untuk deteksi masalah
- Deteksi anomali dan analisis akar penyebab
- Tinjauan kode dan jaminan kualitas
- Optimalisasi penyebaran

Untuk informasi selengkapnya, lihat Menambahkan kecerdasan ke operasi pengembang Anda di AWS situs web.

Mengadopsi praktik rekayasa kekacauan

Rekayasa kekacauan melibatkan sengaja menyuntikkan kegagalan ke dalam sistem untuk menguji kemampuan mereka untuk bertahan dan pulih dari peristiwa tak terduga. Dengan mengidentifikasi kelemahan dan mengatasinya secara proaktif, organisasi dapat meningkatkan keandalan sistem mereka secara keseluruhan dan meminimalkan dampak dari masalah potensial.

Mengadopsi praktik rekayasa kekacauan untuk menguji ketahanan sistem Anda dengan menggunakan alat seperti Gremlin, Chaos Monkey, atau Lakmus. Jalankan eksperimen terkontrol secara teratur untuk mengidentifikasi kerentanan, memvalidasi toleransi kesalahan, dan memastikan bahwa aplikasi Anda menangani kegagalan yang tidak terduga dengan anggun. Pendekatan proaktif ini membantu meningkatkan keandalan sistem dan berkontribusi pada CI/CD pipa yang lebih kuat.

Optimalkan performa

Optimalkan kinerja aplikasi Anda secara terus menerus dengan menggunakan alat pembuatan profil, pemantauan waktu nyata, dan loop umpan balik. Terapkan teknik seperti berikut ini untuk memastikan bahwa aplikasi Anda dapat menangani peningkatan lalu lintas dan permintaan:

- Pengoptimalan kode
- Profiling
- Pemantauan waktu nyata
- Loop umpan balik
- Pembuatan cache
- Penyeimbangan beban
- Skalabilitas dan pengujian kinerja

Menerapkan observabilitas tingkat lanjut

Meningkatkan observabilitas infrastruktur cloud Anda melampaui dasar-dasar pengumpulan, agregasi, dan analisis metrik, log, dan jejak. Ketika observabilitas ditingkatkan dengan alat-alat seperti Amazon CloudWatch dan AWS X-Ray, itu berkembang menjadi praktik strategis yang mendorong pengiriman dan inovasi berkelanjutan.

Dalam CI/CD pipeline yang kuat, observabilitas tingkat lanjut memungkinkan Anda untuk mengungkap wawasan, tidak hanya tentang aplikasi dan infrastruktur Anda, tetapi juga tentang kinerja dan kesehatan seluruh sistem Anda, termasuk pipeline itu sendiri. Wawasan ini membantu Anda:

- Mengidentifikasi, memahami, dan mengatasi masalah potensial dengan cepat untuk meningkatkan stabilitas aplikasi dan mengurangi waktu henti
- Rampingkan CI/CD proses Anda untuk membuat pengiriman yang lebih cepat dan lebih andal
- Dapatkan wawasan yang lebih dalam tentang dampak perubahan kode dan penerapan untuk mendorong pengambilan keputusan yang tepat
- Mengoptimalkan pemanfaatan sumber daya untuk meningkatkan efisiensi operasional dan efektivitas biaya

Untuk meningkatkan observabilitas:

- Sematkan observabilitas ke setiap lapisan aplikasi dan infrastruktur Anda untuk membuat tampilan komprehensif tentang kinerja, perilaku, dan kesehatan sistem Anda.
- Pusatkan pengumpulan, penyimpanan, dan analisis data dengan alat seperti Amazon CloudWatch untuk menyatukan data observabilitas Anda agar mudah diakses dan diinterpretasi.
- Gunakan AWS X-Ray untuk penelusuran terdistribusi untuk memahami kinerja aplikasi Anda dan layanan dasarnya.
- Buat loop umpan balik untuk perbaikan berkelanjutan, dan gunakan data observabilitas Anda untuk mendorong peningkatan berulang ke sistem Anda.

Mengadopsi observabilitas tingkat lanjut bukan hanya tentang mempertahankan sistem Anda—ini adalah langkah strategis menuju pencapaian keunggulan operasional dan mendorong inovasi berkelanjutan dalam organisasi Anda.

Menerapkan GitOps praktik

Menerapkan GitOps praktik untuk mengelola infrastruktur dan konfigurasi aplikasi dengan menggunakan repositori Git sebagai sumber kebenaran tunggal. Pendekatan ini menyederhanakan manajemen perubahan, meningkatkan keterlacakan, dan memastikan konsistensi di seluruh lingkungan.

Kesimpulan

Panduan ini berfungsi sebagai buku pedoman untuk keberhasilan implementasi dan pengelolaan fondasi untuk adopsi cloud yang sukses. Ini membahas bagaimana:

- Secara langsung mengatasi tantangan teknis dan seluk-beluk dalam arsitektur platform untuk menetapkan pedoman dan prinsip yang kuat untuk lingkungan cloud Anda dan data yang berada di dalamnya.
- Bangun rekayasa platform dengan penyediaan dan orkestrasi yang kuat.
- Aktifkan penggunaan lingkungan cloud multi-akun yang sesuai yang mengelola dan mendistribusikan produk cloud yang disetujui kepada pengguna dengan cara yang terukur dan dapat diulang.
- Mendukung keputusan arsitektur data dengan alat yang diperlukan untuk rekayasa data untuk mendorong pengambilan keputusan berbasis data.
- Pasangkan kemampuan ini dengan strategi pengembangan aplikasi modern dan proses CI/CD untuk mempromosikan kelincahan, efisiensi, dan inovasi dalam organisasi Anda.
- Bangun hubungan lintas fungsi dan ambil masukan dari perspektif AWS CAF lain dalam pengambilan keputusan Anda sendiri untuk memastikan keberhasilan platform Anda dan tim di belakangnya.

Sumber bacaan lebih lanjut

AWS Sumber daya Cloud Adoption Framework (AWS CAF):

- eBook
- Buku audio
- Infografis
- AWS CAF untuk Artificial Intelligence, Machine Learning, dan Generative AI
- Perspektif bisnis
- Perspektif orang
- Perspektif tata kelola
- Perspektif operasi
- Perspektif keamanan

Sumber daya tambahan:

- AWS Pusat Arsitektur
- AWS studi kasus
- AWS Referensi Umum
- AWS glosarium
- AWS Pusat Pengetahuan
- AWS Bimbingan Preskriptif
- AWS Solusi Mitra (sebelumnya Mulai Cepat)
- AWS dokumentasi keamanan
- AWS Pustaka Solusi
- AWS Pelatihan dan Sertifikasi
- AWS Well-Architected
- AWS whitepaper dan panduan
- Memulai dengan AWS
- Ikhtisar Amazon Web Services

Kontributor

Kontributor untuk panduan ini meliputi:

- Tony Santiago, Arsitek Solusi Mitra Senior, AWS
- Matias Undurraga, Ahli Teknologi Perusahaan, AWS
- Alex Torres, Arsitek Solusi Senior, AWS
- Michael Rhyndress, Konsultan Senior DevSecOps , AWS
- Alex Livingstone, Arsitek dan Spesialis Solusi Utama, CloudOps AWS
- Bruce Cooper, Kepala Sekolah SDE, AWS
- Ravinder Thota, Konsultan Penasihat Senior, AWS
- Sausan Yazji, Manajer Praktek Senior, AWS
- Paul Duvall, Direktur,, DevSecOps AWS
- Jeremy Tennant, Manajer Pengiriman Cloud Utama, AWS
- Sneh Shah, Pimpinan Infrastruktur Utama, AWS
- Sasa Baskarada, Pemimpin Seluruh Dunia, Kerangka Adopsi AWS Cloud, AWS

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	—	25 Oktober 2023

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

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

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

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

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

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

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat database manajemen konfigurasi.

repositori kode

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

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

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan 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, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat Batas Isolasi AWS Kesalahan.

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

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

FM

Lihat model pondasi.

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

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

strategi greenfield

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

pagar pembatas

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

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

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

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

data penahanan

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

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

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

aplikasi idle

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

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretabilitas

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

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

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

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

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

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan 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 Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

Teknologi AI generatif di mana LLM merujuk sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan

pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat Apa itu RAG.

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

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

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

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

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

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

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

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

SPOF

Lihat satu titik kegagalan.

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

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

variabel target

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

daftar tugas

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

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang

memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

gerbang transit

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan

ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. Untuk informasi lebih lanjut, lihat panduan Mengukur ketidakpastian dalam sistem pembelajaran mendalam.

tugas yang tidak terdiferensiasi

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

lingkungan atas

Lihat lingkungan.

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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