



Praktik yang terbukti untuk mengembangkan strategi multicloud

AWS Bimbingan Preskriptif



AWS Bimbingan Preskriptif: Praktik yang terbukti untuk mengembangkan strategi multicloud

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Pengantar	1
1. Seajarkan sasaran multicloud dengan strategi Anda	3
Merger dan akuisisi	3
Keinginan untuk memanfaatkan kemampuan terdiferensiasi jangka panjang dari CSP lain	3
Multicloud di perusahaan induk dan cloud primer di perusahaan yang beroperasi atau lini bisnis	4
2. Perhatikan kesalahpahaman multicloud	6
Semua orang mengadopsi strategi multicloud	6
Multicloud mengurangi risiko penguncian vendor	6
Multicloud meningkatkan ketersediaan dan ketahanan	7
Multicloud memberikan harga yang lebih baik	8
3. Memiliki strategi dan tata kelola yang jelas untuk mendukungnya	11
4. Jangan menyebarkan beban kerja yang berdekatan di awan	13
5. Memiliki Strategi Integrasi Jangka Panjang	14
6. Gunakan wadah secara strategis	16
7. Memiliki CCo E tunggal, tetapi mengkhususkan diri di dalamnya	17
8. Pastikan keamanan selalu menjadi prioritas utama	19
9. Merangkul pendekatan 80/20 atas distribusi yang sama	21
Kesimpulan	23
Sumber daya	24
Riwayat dokumen	25
Glosarium	26
#	26
A	27
B	30
C	32
D	35
E	39
F	41
G	43
H	44
I	45
L	48
M	49

O	53
P	56
Q	59
R	59
D	62
T	66
U	67
V	68
W	68
Z	69
.....	lxxi

Praktik yang terbukti untuk mengembangkan strategi multicloud

Tom Godden dan Ellie Tamari, Amazon Web Services

September 2025 ([riwayat dokumen](#))

Organizations saat ini menghadapi pesan yang bertentangan tentang adopsi multicloud. Beberapa menyarankan untuk tidak melakukannya sepenuhnya, sementara yang lain mengklaim bahwa setiap orang beralih ke lingkungan multicloud. Kenyataannya terletak di antara ekstrem ini: Alasan yang sah ada baik untuk dan melawan strategi multicloud, dan kesuksesan bergantung pada keseimbangan nilai bisnis potensial terhadap kompleksitas dan risiko yang melekat.

Pada AWS, komitmen kami terhadap interoperabilitas adalah alasan utama banyak pelanggan memilih platform kami. Kami percaya dalam memberi Anda kebebasan untuk berinovasi di mana pun beban kerja Anda berada dan memberdayakan Anda untuk memilih teknologi yang paling sesuai dengan kebutuhan Anda. Di AWS, kami telah berada di garis depan dalam mengembangkan solusi yang memungkinkan Anda untuk membangun dan menyebarkan aplikasi di lingkungan apa pun. Pendekatan yang berpusat pada pelanggan ini sangat penting bagi AWS Cloud, yang dipercaya oleh jutaan pelanggan di seluruh dunia.

Kami memahami bahwa pelanggan membutuhkan platform cloud yang bekerja secara mulus dengan alat yang ada dan pilihan teknologi masa depan. Anda tidak perlu membangun kembali semuanya ketika Anda menambahkan kemampuan dari penyedia lain. Cloud Anda akan membantu Anda terhubung, mengamankan, dan mengelola beban kerja di seluruh lingkungan tanpa memaksa Anda untuk menjadi ahli di setiap platform. AWS membangun titik koneksi langsung ke layanannya untuk membantu Anda beroperasi secara efektif, baik strategi Anda menggunakan secara AWS eksklusif atau mengikuti pendekatan multicloud selektif.

Kami menyadari bahwa setiap organisasi memiliki persyaratan bisnis unik yang mendorong keputusan strategi cloud mereka. Baik Anda menjalankan beban kerja terutama pada AWS, menjalankannya di beberapa cloud, atau menggunakan AWS sebagai bagian dari arsitektur multicloud yang lebih luas, kami berkomitmen untuk membantu Anda sukses. AWS memberikan kedalaman dan luasnya perkakas dan kemampuan untuk membantu Anda membangun, bermigrasi, dan beroperasi dengan lebih mudah dan cepat, di mana pun beban kerja Anda berada. AWS alat menyederhanakan manajemen di seluruh penyedia sekaligus memaksimalkan kinerja dan nilai investasi cloud Anda.

Paper ini berfokus pada prinsip-prinsip yang telah terbukti untuk berhasil dengan strategi multicloud, termasuk kapan dan di mana pendekatan multicloud masuk akal dan bagaimana AWS membantu perusahaan berhasil dengan strategi multicloud mereka. Ini memberikan panduan preskriptif untuk membantu eksekutif membuat strategi informasi dan pilihan pengambilan keputusan yang terkait dengan adopsi multicloud. Paper ini tidak menawarkan diskusi teknis dan mendalam tentang implementasi multicloud. Untuk dukungan implementasi teknis dan bantuan dengan tantangan spesifik Anda, kami sarankan Anda [bekerja dengan arsitek AWS solusi Anda](#).

Paper ini menyajikan sembilan prinsip yang terbukti untuk kesuksesan multicloud berdasarkan pengalaman kami dengan AWS pelanggan perusahaan. Setiap prinsip membahas aspek penting dari strategi multicloud, mulai dari menyelaraskan tujuan bisnis hingga implementasi keamanan. Dengan menerapkan prinsip-prinsip ini, organisasi dapat menavigasi kompleksitas multicloud dengan percaya diri.

- [Prinsip 1. Sejajarkan sasaran multicloud dengan strategi Anda](#)
- [Prinsip 2. Waspada terhadap kesalahpahaman multicloud](#)
- [Prinsip 3. Memiliki strategi dan tata kelola yang jelas untuk mendukungnya](#)
- [Prinsip 4. Jangan menyebarkan beban kerja yang berdekatan di awan](#)
- [Prinsip 5. Memiliki strategi integrasi jangka panjang](#)
- [Prinsip 6. Gunakan kontainer secara strategis](#)
- [Prinsip 7. Memiliki CCo E tunggal, tetapi mengkhususkan diri di dalamnya](#)
- [Prinsip 8. Pastikan keamanan selalu menjadi prioritas utama](#)
- [Prinsip 9. Merangkul pendekatan 80/20 atas distribusi yang sama](#)

Prinsip 1. Sejajarkan sasaran multicloud dengan strategi Anda

Penelitian oleh Gartner dan tren industri menunjukkan bahwa organisasi semakin mengadopsi pendekatan multicloud untuk memenuhi kebutuhan bisnis tertentu. Skenario berikut menunjukkan kapan infrastruktur multicloud dapat menguntungkan secara strategis.

Merger dan akuisisi

Merger dan akuisisi (M&A) membuat keputusan langsung tentang strategi cloud. Meskipun mengoperasikan beberapa cloud dapat meningkatkan biaya dan kompleksitas, konsolidasi yang cepat dapat menunda nilai integrasi dan mengganggu operasi bisnis. Keputusan cloud Anda menjadi pusat untuk mewujudkan manfaat M&A.

Perencanaan integrasi harus memperhitungkan lanskap teknologi yang lengkap. Setiap beban kerja memerlukan evaluasi dalam konteks timeline integrasi dan prioritas bisnis Anda.

Bimbingan kami:

- Mengembangkan strategi konsolidasi berbasis bisnis yang menyeimbangkan kebutuhan integrasi langsung dengan efisiensi operasional jangka panjang. Pertahankan beberapa cloud pada awalnya dalam keadaan di mana konsolidasi yang tergesa-gesa dapat mengganggu operasi bisnis penting atau menunda realisasi nilai M&A.
- Buat kriteria penempatan beban kerja yang jelas yang selaras dengan timeline integrasi Anda. Memprioritaskan aplikasi yang menghasilkan pendapatan dan proses bisnis inti sambil memperhitungkan ketergantungan teknis dan persyaratan operasional.

Keinginan untuk memanfaatkan kemampuan terdiferensiasi jangka panjang dari CSP lain

Ketakutan akan kehilangan membuat beberapa perusahaan menginginkan sedikit dari setiap cloud. Keputusan penempatan beban kerja mempengaruhi seluruh organisasi—mulai dari tim teknik hingga keuangan hingga operasi keamanan.

Oleh karena itu, Organisasi perlu memeriksa alasan mereka untuk mengejar banyak awan. Beberapa berpendapat bahwa setiap beban kerja harus hidup di penyedia layanan cloud (CSP) yang paling

memenuhi kebutuhannya. Namun, optimalisasi beban kerja individu harus seimbang dengan dampak organisasi yang lebih luas. Setiap penyedia cloud tambahan berisiko meningkatkan kompleksitas operasional, menciptakan persyaratan bakat baru, dan memperkenalkan pertimbangan keamanan yang memengaruhi seluruh organisasi teknologi.

Bimbingan kami:

- Ikuti pendekatan 80/20: Pilih penyedia utama untuk sebagian besar beban kerja dan pertimbangkan penyedia tambahan hanya untuk kasus penggunaan spesifik dan bernilai tinggi. Strategi ini memaksimalkan efisiensi dan retensi bakat sekaligus mengurangi kompleksitas.
- Pertimbangkan total biaya operasi melintasi awan. Sertakan perkakas keamanan, produk tata kelola, sistem manajemen keuangan, dan overhead operasional dalam analisis Anda.
- Evaluasi dependensi dan interaksi setiap beban kerja. Beban kerja jarang beroperasi secara terpisah; mereka berbagi data, kontrol keamanan, dan proses operasional.
- Melakukan analisis harga-kinerja menyeluruh di seluruh penyedia. Bandingkan tidak hanya biaya langsung tetapi juga overhead mengelola beberapa lingkungan.

Multicloud di perusahaan induk dan cloud primer di perusahaan yang beroperasi atau lini bisnis

Perusahaan ekuitas swasta dan perusahaan induk menghadapi pertimbangan strategi cloud yang unik. Perusahaan portofolio mereka sering mempertahankan strategi cloud independen, sering kali dihasilkan dari aktivitas M&A sebelumnya. Struktur ini mengurangi kompleksitas yang biasanya terkait dengan operasi multicloud, karena setiap unit bisnis beroperasi secara independen. Namun, kemandirian ini dapat membatasi peluang untuk memanfaatkan diskon volume dan insentif pembelian di seluruh perusahaan.

Efektivitas strategi cloud di tingkat perusahaan induk tergantung pada otonomi perusahaan portofolio dan kebutuhan teknologi masing-masing. Meskipun konsolidasi dapat menciptakan leverage pembelian, itu mungkin bertentangan dengan model operasi independen yang khas dari perusahaan induk dan portofolio ekuitas swasta.

Bimbingan kami:

- Memahami struktur diskon volume CSP. Setiap penyedia menawarkan mekanisme untuk menambah atau menghapus anak perusahaan dari perjanjian perusahaan dan memintal unit bisnis menjadi entitas yang terpisah. Ini mewakili [keputusan pintu dua arah](#).

- Rencanakan komitmen pembelian cloud dengan hati-hati. Libatkan tim akun CSP Anda lebih awal, atau hubungi [kompetensi AWS Cloud Operations](#) untuk mendapatkan bantuan. AWS Partner
- Seimbangkan independensi dengan efisiensi. Pertimbangkan layanan bersama atau perjanjian pembelian yang menguntungkan perusahaan portofolio tanpa membatasi operasi mereka.
- Fokus pada tujuan bisnis terlebih dahulu. Kembangkan strategi teknologi yang mendukung model operasi Anda daripada mengejar strategi multicloud untuk kepentingannya sendiri.
- Mengevaluasi strategi cloud melalui lensa manajemen portofolio. Pertimbangkan bagaimana pilihan cloud memengaruhi potensi divestasi atau akuisisi di masa depan.

Prinsip 2. Perhatikan kesalahpahaman multicloud

Saat Anda mengembangkan strategi multicloud Anda, hindari kesalahpahaman umum yang dibahas di bagian berikut.

Semua orang mengadopsi strategi multicloud

Perusahaan penasihat dan perusahaan media melukiskan gambaran kompleks adopsi multicloud. Penelitian menunjukkan minat yang luas pada pendekatan multicloud, tetapi pola pengeluaran sering menceritakan kisah yang berbeda. Dalam praktiknya, banyak perusahaan mempertahankan lingkungan cloud tunggal atau hubungan primary/secondary CSP yang jelas. Pemutusan ini menyoroti pentingnya melihat melampaui berita utama dan berfokus pada kebutuhan spesifik organisasi Anda.

Bimbingan kami:

- Buat keputusan cloud berdasarkan kebutuhan bisnis spesifik Anda alih-alih mengikuti tren industri. Fokus pada biaya dan risiko yang terukur untuk organisasi Anda.
- Periksa kasus penggunaan multicloud dalam konteks industri Anda. Strategi cloud yang bekerja untuk perusahaan teknologi konsumen mungkin tidak diterjemahkan ke layanan keuangan, manufaktur, atau lingkungan game.
- Pertimbangkan gravitasi data sebagai faktor utama dalam keputusan penempatan beban kerja. Lokasi dan pergerakan data sering menentukan arsitektur cloud yang paling efektif.
- Lihatlah melampaui statistik adopsi untuk memahami pola pengeluaran. Tingkat adopsi multicloud yang dilaporkan tinggi sering menutupi pola pengeluaran aktual.
- Evaluasi kendala teknis sebelum berkomitmen pada lingkungan multicloud. Beberapa beban kerja berkinerja terbaik ketika komponennya tetap berada dalam satu lingkungan cloud.

Multicloud mengurangi risiko penguncian vendor

Fleksibilitas vendor adalah pertimbangan yang sah dalam pengembangan strategi cloud. Organizations menghargai kemampuan untuk menyesuaikan pilihan teknologi mereka saat kebutuhan bisnis berkembang. Kekhawatiran ini mencerminkan pengalaman sebelumnya dengan investasi TI tradisional yang menciptakan komitmen jangka panjang yang mengikat. Layanan cloud menawarkan dinamika yang berbeda seputar fleksibilitas penyedia. AWS menyediakan layanan open source yang kompatibel dan opsi portabilitas data yang mengurangi hambatan teknis untuk migrasi.

Namun, trade-off antara fleksibilitas dan efisiensi operasional tetap penting. Organizations harus mempertimbangkan nilai bisnis dari mempertahankan opsi penyedia terhadap keuntungan teknis dari integrasi mendalam dengan layanan khusus dari penyedia utama.

Beberapa pelanggan mencoba menghindari penguncian dengan merekayasa solusi cloud-agnostik yang menggunakan wadah. Pendekatan ini sering membatasi mereka pada layanan komputasi dan penyimpanan dasar, dan melewati keunggulan kemampuan cloud canggih. Pengalaman kami menunjukkan bahwa strategi ini menambah kompleksitas yang cukup besar karena peningkatan waktu pengembangan dan sumber daya yang dibutuhkan, dibandingkan dengan menggunakan layanan asli.

Bimbingan kami:

- Pertimbangkan biaya penuh arsitektur cloud-agnostik. Overhead teknik tambahan mungkin tidak membenarkan manfaat portabilitas.
- Gunakan kemampuan cloud-native untuk nilai maksimum. Layanan komputasi dan penyimpanan dasar saja sering mengorbankan keuntungan signifikan dalam keamanan, skalabilitas, dan inovasi.
- Rencanakan strategi cloud berdasarkan kebutuhan bisnis. Ketika implementasi multicloud menambah nilai yang jelas, seperti kemampuan untuk melayani pengguna di berbagai platform, investasi teknik tambahan menjadi bermanfaat.
- Evaluasi skenario dan biaya keluar yang realistis. Bandingkan kemungkinan dan biaya perubahan penyedia dengan manfaat menggunakan rangkaian lengkap Layanan AWS.
- Membangun di atas fondasi open source AWS. AWS Layanan terkelola seperti [Amazon Relational Database Service \(Amazon RDS\)](#) memberi Anda fleksibilitas dan keunggulan operasional, serta mendukung mesin database yang Anda gunakan saat ini.
- Manfaatkan alat migrasi komprehensif yang disediakan oleh AWS. Kami membantu Anda memindahkan beban kerja ke segala arah dan memberikan jalan keluar data gratis jika Anda pergi AWS untuk menggunakan penyedia lain. Untuk informasi lebih lanjut, lihat posting AWS blog [Transfer data gratis ke internet saat pindah AWS](#).

Multicloud meningkatkan ketersediaan dan ketahanan

Keyakinan akan peralihan beban kerja yang mulus antara penyedia cloud selama pemadaman mendorong beberapa organisasi menuju strategi multicloud. Pola pikir ini menciptakan pandangan yang terlalu disederhanakan tentang ketahanan infrastruktur cloud yang mengabaikan realitas teknis mendasar.

Berdasarkan pengalaman bertahun-tahun bekerja dengan pelanggan multicloud AWS, kami telah melihat bahwa mempertahankan portabilitas beban kerja penuh antar penyedia seringkali menciptakan kompleksitas substansif tanpa memberikan semua manfaat yang diharapkan. Aplikasi intensif data menghadapi tantangan yang tidak dapat diatasi karena kendala gravitasi data. Faktanya, dalam pandangan kami, hampir tidak mungkin bagi organisasi untuk berhasil menerapkan failover multicloud yang benar-benar mulus untuk beban kerja yang berat data.

Lydia Leong, Analis Wakil Presiden Terhormat di Gartner, memperkuat perspektif ini dalam [posting media sosial](#): “Failover multicloud rumit dan mahal sampai-sampai hampir selalu tidak praktis, dan ini bukan cara yang sangat efektif untuk mengatasi risiko ketahanan cloud.” Diferensiasi yang melekat antara penyedia dalam jaringan, penyimpanan, database, pembelajaran mesin, dan keamanan membuat portabilitas yang sebenarnya hampir tidak mungkin. Menyebarkan beban kerja di seluruh penyedia dapat meningkatkan risiko, karena kegagalan di kedua lingkungan dapat memicu pemadaman di semua lingkungan.

Bimbingan kami:

- Fokus pada penguasaan AWS kemampuan untuk beban kerja individu alih-alih mengejar arsitektur multicloud yang kompleks.
- Bangun ketahanan melalui Wilayah AWS dan Availability Zones alih-alih mencoba failover lintas penyedia. Untuk selami teknis yang mendalam tentang bagaimana AWS secara otomatis gagal atas beban kerja antar pusat data fisik, lihat posting AWS blog, [Pergeseran otomatis Zonal — Secara otomatis mengalihkan lalu lintas Anda dari Availability Zone saat kami mendeteksi](#) potensi masalah.
- Migrasikan beban kerja secara strategis ke AWS, dan fokus pada satu aplikasi sekaligus untuk memaksimalkan kesuksesan.

Multicloud memberikan harga yang lebih baik

Daya saing harga mungkin merupakan argumen terlemah dari semua untuk lingkungan multicloud. Pengalaman organisasi dengan perangkat lunak atau kontrak pusat data yang rumit dan mahal yang mengunci mereka ke dalam perjanjian multi-tahun telah membuat mereka waspada saat mendapatkan layanan TI. Pendekatan pengadaan tradisional belum disesuaikan dengan pay-as-you-go pembelian, diskon volume, atau realitas persaingan harga di cloud. (Per Januari 2025, AWS telah menurunkan harga 151 kali sejak awal.)

Penggerak tunggal terbesar pengurangan biaya adalah lingkungan cloud yang dikelola dengan baik dan dioptimalkan. Perusahaan melihat optimalisasi biaya yang lebih baik dengan bekerja terutama dengan penyedia yang layanannya menawarkan keunggulan kinerja harga (seperti contoh komputasi yang didasarkan pada chip yang dirancang khusus seperti [AWS Graviton](#)) dan memiliki solusi manajemen keuangan cloud yang unggul. Menurut [studi Hackett Group tahun 2022](#) terhadap lebih dari 1.000 organisasi, pengeluaran infrastruktur sebagai persentase dari total pengeluaran TI adalah 20% lebih rendah untuk AWS pelanggan dibandingkan dengan organisasi multicloud.

Pengalaman kami menunjukkan bahwa perusahaan tidak mengantisipasi biaya tambahan dan kompleksitas operasi di beberapa cloud, juga tidak menimbang biaya ini dengan tepat terhadap keuntungan yang dirasakan dalam keterlibatan head-to-head sumber.

Bimbingan kami:

- Bangun strategi pengoptimalan biaya Anda di Pilar AWS Pengoptimalan Biaya [Kerangka Well-Architected](#). Ada lima prinsip desain:
 - Menerapkan manajemen keuangan cloud: Untuk mencapai kesuksesan finansial dan mempercepat realisasi nilai bisnis di cloud, Anda harus berinvestasi dalam manajemen keuangan cloud. Organisasi Anda harus mendedikasikan waktu dan sumber daya yang diperlukan untuk membangun kapabilitas di domain teknologi baru dan manajemen penggunaan ini. Seperti halnya kemampuan keamanan atau operasi Anda, Anda perlu menumbuhkan kemampuan melalui pembangunan pengetahuan, program, sumber daya, dan proses untuk membantu menjadi organisasi yang hemat biaya.
 - Adopsi model konsumsi: Bayar untuk sumber daya komputasi yang Anda gunakan saja, dan tingkatkan atau turunkan penggunaan tergantung kebutuhan bisnis Anda. Misalnya, lingkungan pengembangan dan pengujian biasanya digunakan hanya selama delapan jam sehari selama minggu kerja. Anda dapat menghentikan sumber daya ini ketika mereka tidak digunakan untuk penghematan biaya potensial 75% (40 jam versus 168 jam).
 - Ukur efisiensi keseluruhan: Ukur output bisnis dari beban kerja Anda dan biaya yang terkait dengan pengiriman. Gunakan data ini untuk memahami keuntungan-keuntungan yang dapat Anda peroleh dengan melakukan peningkatan output, peningkatan fungsionalitas, dan penurunan biaya.
 - Berhenti menghabiskan uang untuk angkat berat yang tidak berdiferensiasi: CSPs lakukan pengangkatan berat operasi pusat data seperti racking, stacking, dan powerering server. Mereka juga menghilangkan beban operasional mengelola sistem operasi dan aplikasi dengan menggunakan layanan terkelola. Ini memungkinkan Anda untuk fokus pada pelanggan dan proyek bisnis Anda alih-alih infrastruktur TI.

- Analisis dan atribusikan pengeluaran: Cloud memudahkan Anda melakukan identifikasi biaya dan penggunaan beban kerja secara akurat, sehingga kemudian memungkinkan Anda melakukan atribusi biaya IT ke aliran pendapatan secara transparan ke pemilik beban kerja masing-masing. Hal ini akan membantu Anda dalam mengukur laba atas investasi (ROI) dan memberikan peluang kepada pemilik beban kerja untuk mengoptimalkan sumber daya mereka dan memangkas biaya.
- Mengingat overhead keuangan beroperasi di berbagai penyedia, kami memandu pelanggan untuk berinvestasi besar-besaran dalam otomatisasi dan alat pengoptimalan biaya. Setiap CSP menawarkan alat asli yang luas di bidang ini, seperti [Hub Optimisasi Biaya AWS](#). Sebagian besar alat asli memberikan kemampuan luar biasa bagi pelanggan di lingkungan cloud mereka. Namun, untuk memahami pengeluaran di beberapa CSPs, Anda dapat memilih dari serangkaian produk ISV dan perangkat lunak sebagai layanan (SaaS) yang kaya yang memperluas kemampuan ini untuk memberikan satu pengalaman untuk pengoptimalan biaya.
- Menipiskan daya beli melalui strategi ekuitas belanja tidak menghasilkan nilai bisnis. Ini dapat merusak diskon volume potensial dan berpotensi merusak desain teknis. Cara paling efisien untuk menggunakan layanan cloud adalah dengan menggunakan penyedia utama untuk sebagian besar operasi Anda dan menggunakan yang lain CSPs hanya di mana ia menambah nilai bisnis.

Prinsip 3. Memiliki strategi dan tata kelola yang jelas untuk mendukungnya

Memutuskan untuk mengejar strategi multicloud tidak cukup; Anda harus menetapkan strategi untuk mencapai tujuan Anda, termasuk tata kelola yang jelas untuk beban kerja mana yang akan pergi ke mana dan mengapa. Kriteria evaluasi harus digunakan untuk mengoptimalkan beban kerja dan dependensinya. Jika evaluasi diserahkan kepada individu, penyebaran yang tidak terkoordinasi kemungkinan CSPs akan mengikis nilai strategi multicloud. Kami menyarankan Anda mengevaluasi kinerja beban kerja CSP secara teratur dan menggunakan penilaian Anda sebagai masukan utama untuk pemilihan CSP, kriteria, dan penggunaan future.

Strategi tata kelola yang efektif membutuhkan visibilitas ke dalam jumlah total layanan, aplikasi, dan komponen yang digunakan di seluruh perusahaan. Integral dari ini adalah strategi penandaan yang kuat yang mencakup CSPs dan menetapkan kepemilikan, penggunaan, dan lingkungan yang jelas (seperti pengembangan, QA, pementasan, dan produksi) untuk semua sumber daya yang digunakan. Semuanya harus ditandai ke pemilik; jika tidak ditandai atau pemilik tidak dapat diidentifikasi, itu harus dihapus. Kami bekerja sama dengan organisasi jasa keuangan utama yang secara otomatis menemukan dan menghapus sumber daya yang tidak ditandai, dan menganggap ini sebagai praktik terbaik, terlepas dari ketidaknyamanan yang ditimbulkannya kepada tim pengembangan. Pendekatan penandaan ini mengkodifikasikan aturan tata kelola dan mengotomatiskan penegakan alih-alih membuat blok untuk kemajuan (yaitu, menerapkan pagar pembatas, bukan gerbang). Biaya, operasi, dan keamanan harus dilacak, dipantau, dan ditindaklanjuti dengan cara yang sama, dengan kedalaman data dan transparansi yang sama. CSPs

Ketika Anda menerapkan strategi multicloud, membangun struktur akun yang jelas dan konsisten di seluruh penyedia cloud sangat penting untuk menjaga kontrol operasional dan keamanan. Kami merekomendasikan mengadopsi hub-and-spoke model, di mana Anda membuat terpisah Akun AWS untuk unit bisnis yang berbeda. Ini ditambatkan oleh dua akun pusat penting: security/audit akun untuk kepatuhan terkonsolidasi dan pemantauan keamanan, dan akun jaringan pusat untuk mengelola interkoneksi. (Pendekatan ini dikodifikasi dalam desain. [AWS Control Tower](#) Namun, prinsip-prinsip hak istimewa dan pemisahan tugas sama-sama berlaku untuk awan lainnya. The [AWS Well-Architected](#) Framework membahas konsep-konsep ini panjang lebar, dan sangat direkomendasikan untuk khalayak teknis.) Pendekatan dasar ini harus dicerminkan di seluruh penyedia cloud untuk menjaga konsistensi dalam tata kelola dan operasi. Akun beban kerja harus diatur oleh lingkungan (pengembangan, pementasan, produksi) atau fungsi, dengan proses yang jelas ditetapkan untuk pembuatan dan penghapusan akun.

Bimbingan kami:

- Menerapkan strategi penandaan yang komprehensif untuk mempertahankan pola kepemilikan dan penggunaan yang jelas di semua sumber daya cloud. Lacak lingkungan, pusat biaya, aplikasi, dan unit bisnis melalui kebijakan penandaan yang konsisten. Hapus sumber daya yang tidak memiliki tag yang tepat untuk menegakkan standar tata kelola dan menjaga kejelasan lingkungan.
- Buat kerangka kerja kepatuhan terpadu yang memetakan persyaratan peraturan di seluruh lingkungan multicloud Anda. Menjaga dokumentasi yang jelas tentang bagaimana kontrol dan sertifikasi masing-masing penyedia cloud mendukung kewajiban kepatuhan Anda.
- Otomatiskan penegakan tata kelola melalui otomatisasi alih-alih menggunakan proses persetujuan manual. Kode aturan tata kelola Anda ke dalam sistem otomatis yang mencegah pelanggaran kebijakan sebelum terjadi. Ini menghilangkan kesalahan manusia sambil mempertahankan kecepatan pengembangan.
- Struktur akun dalam hub-and-spoke model dengan keamanan terpusat dan kontrol jaringan. Buat akun khusus untuk audit keamanan dan manajemen jaringan untuk memusatkan fungsi penting. Yayasan ini memungkinkan kebijakan keamanan yang konsisten dan konektivitas jaringan di seluruh organisasi.
- Untuk mempertahankan batasan operasional, buat akun, langganan, atau proyek terpisah (tergantung pada nomenklatur CSP Anda) untuk lingkungan dan fungsi yang berbeda. Membagi beban kerja dengan pengembangan, pementasan, dan lingkungan produksi. Pemisahan ini mencegah insiden keamanan menyebar dan mempertahankan domain operasional yang jelas.
- Pantau biaya, operasi, dan keamanan melalui metrik yang konsisten di seluruh lingkungan. Menerapkan pemantauan terpadu untuk pemanfaatan sumber daya, peristiwa keamanan, dan pola pengeluaran. Gunakan data ini untuk mengoptimalkan penempatan beban kerja dan keputusan alokasi sumber daya.
- Mencegah penggunaan cloud yang tidak sah melalui kebijakan organisasi dan kontrol otomatis. Tentukan proses yang jelas untuk pembuatan akun dan penyediaan sumber daya. Menerapkan [kebijakan pengendalian layanan \(SCPs\)](#) untuk menegakkan kepatuhan terhadap standar organisasi di semua akun.
- Menetapkan kontrol detektif dan preventif untuk mencegah bayangan TI muncul melalui akun penyedia yang tidak sah. Pantau penggunaan cloud yang tidak sah melalui laporan pengeluaran dan lalu lintas jaringan. Blokir akses penyedia yang tidak sah sambil mempertahankan jalur inovasi yang disetujui.

Prinsip 4. Jangan menyebarkan beban kerja yang berdekatan di awan

Menyebarkan beban kerja yang berdekatan di beberapa penyedia cloud menciptakan kompleksitas, risiko, dan biaya yang tidak perlu. Ketika beban kerja yang memproses dan menganalisis data bersama-sama menjangkau beberapa penyedia, organisasi menghadapi tantangan dalam pergerakan data, sinkronisasi, dan konsistensi. Tim harus menavigasi antarmuka manajemen APIs, model keamanan, dan proses operasional yang berbeda untuk setiap penyedia, yang meningkatkan kemungkinan kesalahan dan menambahkan overhead operasional. Kompleksitas ini meningkatkan kemungkinan kesalahan dan overhead operasional, dan dapat menghambat kelincahan dan skalabilitas.

Namun, dalam beberapa skenario praktis, organisasi mungkin perlu mendistribusikan beban kerja yang berdekatan di cloud karena persyaratan bisnis atau teknis tertentu. Dalam kasus ini, kami menyarankan Anda untuk menetapkan kriteria dan prinsip panduan yang jelas untuk mengevaluasi trade-off, dan memastikan bahwa pendekatan tersebut selaras dengan strategi multicloud organisasi Anda secara keseluruhan.

Ketika organisasi memilih untuk mendistribusikan beban kerja di beberapa cloud, mengadopsi arsitektur yang berpusat pada pesan dan kopling longgar dapat meringankan banyak tantangan terkait. Ini adalah cara terbaik untuk memisahkan kekhawatiran antara cloud dan untuk mengurangi cakupan dampak jika penyedia mengalami gangguan. Operasi yang paling terikat waktu, seperti transaksi keuangan, idealnya harus disimpan dalam satu lingkungan. Pemadaman di satu lingkungan tidak boleh dibiarkan membahayakan beban kerja di lingkungan lain.

Bimbingan kami:

- Rancang beban kerja cloud untuk independensi operasional guna meminimalkan dependensi real-time antar penyedia. Ketika distribusi beban kerja diperlukan, terapkan mekanisme transfer data massal yang efisien alih-alih mempertahankan koneksi lintas cloud yang konstan.
- Mengevaluasi setiap beban kerja terdistribusi yang diusulkan terhadap kriteria bisnis yang jelas. Pertimbangkan manfaat strategis dan kompleksitas operasional yang diperkenalkan oleh distribusi.

Prinsip 5. Memiliki Strategi Integrasi Jangka Panjang

Hati-hati ketika Anda memindahkan volume data yang besar antar aplikasi di cloud yang berbeda, terutama jika sumber daya komputasi dan aplikasi Anda digunakan dalam satu CSP, dan sumber daya penyimpanan data Anda digunakan di yang lain. Situasi seperti itu dapat menambah kompleksitas dan latensi yang mungkin mengimbangi manfaat yang dirasakan. Kami berbicara dengan banyak pelanggan yang memiliki data lake di satu cloud tetapi ingin melakukan pembelajaran mesin (ML) atau analitik dengan alat dari CSP lain. Memutuskan di mana menempatkan beban kerja di lingkungan multicloud adalah salah satu keputusan yang paling penting — dan seringkali paling menantang — yang dihadapi organisasi. Kami menyarankan Anda mengevaluasi setiap keputusan penempatan beban kerja melalui tiga dimensi penting: persyaratan teknis, kebutuhan bisnis, dan kekuatan penyedia.

Mulai evaluasi teknis dengan memetakan karakteristik penting setiap beban kerja: daya komputasi, operasi data, kebutuhan waktu respons, dan persyaratan pertumbuhan. Aplikasi secara alami berkinerja terbaik ketika mereka berada di dekat data mereka. Memindahkan aplikasi dari sumber datanya menciptakan rintangan teknis yang tidak perlu dan memperlambat kinerja.

Keputusan bisnis harus memperhitungkan harga penyedia, persyaratan residensi data, dan kontrak vendor. Setiap penempatan beban kerja mempengaruhi operasi, keamanan, dan produktivitas seluruh organisasi. Melihat beban kerja secara terpisah mengarah pada keputusan yang kurang optimal.

Bimbingan kami:

- Menerapkan transfer data massal antar cloud alih-alih akses real-time. Jadwalkan penyegaran data berkala dengan menggunakan operasi massal yang efisien alih-alih menggunakan panggilan API konstan antar cloud. Pendekatan ini mengurangi biaya, meningkatkan keandalan, dan mempertahankan kinerja yang konsisten. Misalnya, ekspor merangkum data penjualan harian alih-alih menanyakan transaksi individual di cloud.
- Pertimbangkan gravitasi data saat merancang penempatan beban kerja. Jaga agar aplikasi tetap dekat dengan sumber data utama mereka untuk mempertahankan kinerja dan mengurangi biaya. Model ML, mesin analitik, dan sistem pemrosesan transaksi semuanya mendapat manfaat dari akses langsung ke data mereka. Memindahkan beban kerja ini dari data mereka menciptakan latensi dan kompleksitas jaringan yang tidak perlu.
- Evaluasi keputusan beban kerja dalam konteks strategi cloud lengkap Anda alih-alih meninjaunya secara terpisah. Pertimbangkan bagaimana setiap pilihan penempatan memengaruhi proses

operasional, kontrol keamanan, dan kemampuan tim di seluruh organisasi Anda. Keputusan yang tampaknya optimal untuk satu beban kerja dapat mempersulit pemantauan atau meningkatkan risiko keamanan jika dilihat secara holistik.

- Tentukan kebijakan kepemilikan dan tata kelola data yang jelas yang menentukan di mana berbagai jenis data dapat berada. Buat kerangka klasifikasi data yang mendorong keputusan konsisten tentang penempatan data di seluruh penyedia cloud.

Prinsip 6. Gunakan wadah secara strategis

Kontainer dapat memainkan peran berharga dalam mendukung strategi multicloud, tetapi penting untuk mengenali keterbatasannya juga. Menggunakan kontainer umumnya merupakan ide yang baik untuk aplikasi cloud-native modern, karena mereka memberikan manfaat untuk portabilitas dan konsistensi di seluruh lingkungan yang berbeda. Container adalah platform-agnostik, yang berarti bahwa mereka dapat berjalan di platform cloud atau infrastruktur apa pun yang mendukung teknologi containerization, seperti Kubernetes. Organizations yang menggunakan container dapat mengembangkan dan mengemas aplikasi mereka sekali dan kemudian menerapkannya secara konsisten di beberapa penyedia cloud atau lingkungan lokal, tanpa perlu modifikasi yang signifikan. Dengan merangkum kode aplikasi, dependensi, dan lingkungan runtime dalam wadah, Anda dapat mencapai tingkat portabilitas yang tinggi, yang memungkinkan Anda memindahkan beban kerja secara mulus antara penyedia cloud atau antara pusat data cloud dan lokal.

Namun, kontainer mungkin tidak menyelesaikan setiap kasus penggunaan atau menghilangkan semua tantangan yang mungkin dihadapi organisasi dalam mengadopsi strategi multicloud. Kontainer bekerja paling baik dengan arsitektur modern berbasis layanan mikro, tetapi mungkin tidak cocok untuk aplikasi monolitik yang besar. Selain itu, meskipun kontainer dapat mengatasi aspek portabilitas tertentu, seperti runtime aplikasi, mereka tidak secara otomatis menyelesaikan masalah seputar manajemen data, kebijakan keamanan, dan dependensi lintas cloud lainnya. Organizations masih perlu merencanakan dan merancang solusi multicloud mereka dengan hati-hati untuk memastikan manajemen data yang konsisten, kontrol keamanan terpadu, dan integrasi yang mulus antara komponen yang dihosting cloud dan komponen lokal.

Bimbingan kami:

- Gunakan kapabilitas manajemen kontainer asli masing-masing penyedia cloud untuk memaksimalkan nilai bisnis dan mempercepat pengiriman. Pendekatan ini memastikan kinerja optimal sambil menghindari kompleksitas menciptakan solusi agnostik awan yang jarang memberikan hasil yang berarti.
- Kembangkan strategi kontainer yang membahas gambaran operasional lengkap, termasuk manajemen data, keamanan, dan dependensi lintas cloud. Fokus pada hasil bisnis saat Anda membuat keputusan arsitektur kontainer.

Prinsip 7. Memiliki CCo E tunggal, tetapi mengkhususkan diri di dalamnya

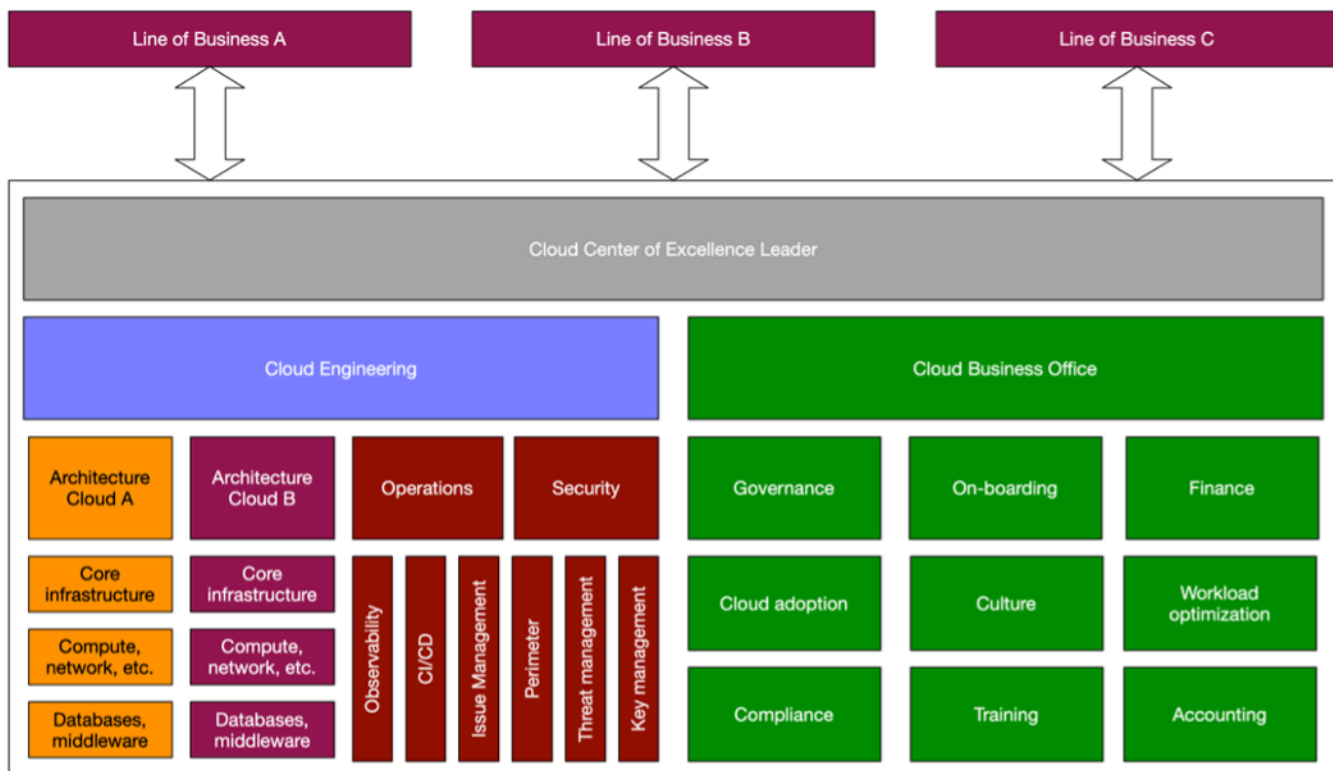
Seperti yang [kami sarankan AWS kepada banyak pelanggan](#), Anda harus membangun Cloud Center of Excellence (CCoE) dalam organisasi Anda untuk memberikan kepemimpinan, standardisasi, dan percepatan perjalanan cloud Anda. Ketika datang ke lingkungan multicloud, kami menemukan bahwa perusahaan yang paling sukses mengambil pendekatan yang seimbang dengan E. CCo

Alih-alih menetapkan CCo E terpisah untuk setiap CSP, kami menyarankan Anda memiliki CCo E tunggal terpadu yang mengawasi strategi multicloud organisasi. Ini membantu memastikan pendekatan yang terkoordinasi dan konsisten alih-alih upaya silo yang dapat menyebabkan divergensi, rekayasa ulang, dan pemborosan. Pastikan bahwa tim dalam CCo E tunggal Anda memiliki keterampilan, alat, dan mekanisme khusus yang diperlukan untuk setiap CSP yang digunakan organisasi Anda. Pengetahuan khusus ini memungkinkan CCo E untuk mengatur, mendukung, dan mempercepat penggunaan platform cloud yang berbeda secara efektif.

Misalnya, CCo E harus memiliki pakar AWS khusus yang memahami AWS Cloud, layanan, dan praktik terbaik secara mendalam, serta ahli untuk CSPs orang lain yang dapat memandu penggunaan teknologi cloud tersebut oleh organisasi. Keahlian khusus dalam CCo E tunggal ini dapat membantu organisasi Anda mendapatkan keuntungan dari koordinasi dan standardisasi pendekatan terpusat sambil memastikan bahwa setiap platform cloud digunakan secara optimal.

CCoE tunggal harus berfungsi sebagai badan pengatur pusat yang menetapkan standar, kebijakan, dan praktik terbaik untuk strategi multicloud organisasi. Implementasi aktual dari beban kerja dan proyek cloud dapat didistribusikan ke tim khusus atau unit bisnis sementara CCOE memberikan pengawasan, dukungan, dan koordinasi. Pendekatan seimbang ini membantu memastikan strategi multicloud yang kohesif sambil memberikan tingkat fleksibilitas dan otonomi yang diperlukan dalam organisasi.

Diagram berikut menggambarkan bagaimana CCo E dapat memberikan pendekatan dan tata kelola terpusat di beberapa lini bisnis (LOBs), tim teknik cloud, dan tim Cloud Business Office (CBO).



Bimbingan kami:

- Struktur CCo E Anda untuk mempertahankan pengawasan strategis sambil menanamkan keahlian khusus untuk setiap penyedia cloud. Fokus pada merekrut keahlian mendalam di platform cloud individu alih-alih mencari spesialis multicloud langka, dan mendorong berbagi pengetahuan internal untuk membangun kemampuan organisasi.
- Berdayakan CCo E Anda untuk menetapkan standar di seluruh perusahaan untuk masalah lintas sektoral seperti keamanan dan pengamatan, sambil memberi tim individu otonomi untuk mengeksekusi dalam pedoman ini dengan menggunakan alat dan layanan cloud-native.
- Kembangkan strategi bakat komprehensif yang menyeimbangkan keahlian mendalam di platform cloud primer dengan pengetahuan arsitektur yang lebih luas. Fokus pada membangun tim yang menggabungkan keterampilan khusus cloud yang kuat dengan pengalaman arsitektur perusahaan.

Prinsip 8. Pastikan keamanan selalu menjadi prioritas utama

Pendekatan multicloud membuat lebih sulit untuk memastikan keamanan dengan meningkatkan risiko akses yang tidak sah, karena postur keamanan Anda harus memperhitungkan lebih banyak permukaan serangan. Strategi multicloud sering memaksa perusahaan untuk berurusan dengan beberapa model keamanan di CSPs berbagai bidang seperti manajemen identitas, keamanan jaringan, manajemen aset, dan pencatatan audit. Kompleksitas ini berisiko membuat transparansi lebih sulit, meningkatkan beban tim keamanan, dan meningkatkan risiko.

Otomatisasi keamanan sangat penting dalam lingkungan multicloud. Manajemen identitas harus bekerja dengan mulus di seluruh lingkungan; itu harus menghubungkan penyedia identitas yang ada sambil mempertahankan kebijakan akses yang konsisten. Keamanan membutuhkan perlindungan terintegrasi di seluruh lapisan data, jaringan, dan titik akhir. Klasifikasi data, enkripsi, dan manajemen siklus hidup membentuk fondasi. Keamanan jaringan dibangun di atas desain standar dan pola koneksi. Perlindungan titik akhir melengkapi kerangka kerja melalui manajemen patch yang konsisten dan kontrol berbasis host.

Elemen-elemen dasar ini sangat penting untuk adopsi yang sukses dan aman dari beberapa penyedia cloud dan harus dipertimbangkan sejak awal dalam perencanaan strategi multicloud apa pun.

Bimbingan kami:

- Terapkan kerangka kerja keamanan terintegrasi di lingkungan multicloud Anda yang berfokus pada tiga elemen inti: perlindungan data melalui klasifikasi dan enkripsi standar, keamanan jaringan melalui pola desain yang konsisten, dan perlindungan titik akhir melalui kontrol sistematis dan manajemen patch.
- Buat model operasi keamanan terpadu yang memanfaatkan kemampuan keamanan asli masing-masing penyedia cloud sambil mempertahankan visibilitas dan kontrol terpusat melalui alat dan proses standar.
- Memusatkan pengumpulan dan analisis data keamanan dengan menggunakan [Amazon Security Lake](#). Platform ini mengumpulkan informasi keamanan dari AWS, penyedia cloud lainnya, aplikasi SaaS, dan sistem lokal ke dalam satu tampilan. Ini mendukung Open Cybersecurity Schema Framework (OCSF) dan memungkinkan analisis standar di seluruh lingkungan hybrid dan multicloud Anda. Pendekatan terpusat ini meningkatkan deteksi dan respons ancaman sambil menyederhanakan operasi keamanan.

- Terapkan alat keamanan asli masing-masing penyedia untuk meningkatkan kemampuan perlindungan Anda. Layanan yang dibuat khusus ini menangani fitur khusus penyedia sambil memasukkan data kembali ke platform keamanan terpusat Anda. Kombinasi alat asli dan visibilitas terpusat membantu menyediakan cakupan keamanan yang komprehensif di seluruh infrastruktur Anda.
- Terapkan strategi observabilitas terpadu yang memberikan visibilitas komprehensif di seluruh lanskap cloud Anda, termasuk data operasional dan keamanan, dari bawah ke atas. Standarisasi pendekatan pemantauan terdepan di industri yang memungkinkan pelacakan layanan bisnis secara konsisten di mana pun mereka beroperasi.
- Tetapkan standar di seluruh perusahaan untuk pengumpulan dan visualisasi data operasional yang memungkinkan identifikasi dan penyelesaian masalah cepat di seluruh lingkungan multicloud Anda. Fokus pada menciptakan satu sumber kebenaran untuk wawasan operasional yang melayani pemangku kepentingan teknis dan bisnis.

Prinsip 9. Merangkul pendekatan 80/20 atas distribusi yang sama

Bagaimana Anda mendistribusikan beban kerja di seluruh penyedia secara fundamental menentukan keberhasilan multicloud Anda. Banyak organisasi secara keliru mengejar kesetaraan dalam distribusi cloud mereka, dan mencoba menyebarkan beban kerja secara merata di seluruh penyedia. Pendekatan ini meningkatkan kompleksitas tanpa memberikan manfaat proporsional. Distribusi yang sama memecah kemampuan teknis Anda, melemahkan daya beli Anda, dan menciptakan overhead operasional yang tidak perlu. Tim berjuang untuk mengembangkan keahlian yang mendalam ketika mereka dipaksa untuk mempertahankan kompetensi di berbagai platform secara bersamaan.

Pendekatan 80/20 memberikan hasil yang terbukti lebih baik daripada distribusi yang sama di seluruh awan. Berkonsentrasi 80% dari investasi Anda dengan satu penyedia utama sementara secara selektif menggunakan yang lain untuk kemampuan tertentu menciptakan strategi seimbang yang mengurangi biaya dan kompleksitas. Pendekatan terkonsentrasi ini mempercepat inovasi karena tim Anda dapat mengembangkan keahlian mendalam dengan layanan canggih platform utama Anda. Staf teknis Anda dapat menjadi spesialis dalam satu arsitektur alih-alih mempertahankan pengetahuan tingkat permukaan di berbagai lingkungan. Ketika insinyur menguasai satu platform, mereka membangun lebih efisien, memecahkan masalah lebih cepat, dan menerapkan solusi yang lebih canggih.

Perusahaan yang mengikuti pendekatan 80/20 biasanya melaporkan retensi bakat yang lebih baik karena tim mereka mengembangkan keahlian yang berharga dan dapat dipasarkan alih-alih membentangi tipis di berbagai teknologi. Strategi terkonsentrasi ini juga membantu menyederhanakan manajemen keamanan dengan membatasi kompleksitas model keamanan yang berbeda di seluruh penyedia. Cloud utama menerima sebagian besar investasi Anda dalam alat keamanan, solusi pemantauan, dan proses operasional. Ini menciptakan fondasi keamanan yang lebih kuat daripada yang mungkin dengan sumber daya yang terbagi rata.

Bimbingan kami:

- Pilih penyedia cloud utama yang selaras dengan sebagian besar kebutuhan bisnis dan teknis Anda. Penyedia ini harus mendukung setidaknya 80% dari beban kerja Anda dan menjadi dasar strategi cloud Anda. Fokuskan investasi pelatihan, standar arsitektur, dan proses operasional Anda untuk memaksimalkan nilai dari platform utama ini.
- Kembangkan kriteria yang jelas untuk beban kerja yang menjamin penempatan di cloud sekunder. Kriteria ini harus fokus pada nilai bisnis tertentu yang tidak dapat dicapai pada penyedia utama

Anda. Tahan menempatkan beban kerja di cloud sekunder hanya untuk menjaga ekuitas pengeluaran atau keseimbangan buatan antara penyedia.

- Struktur perjanjian bisnis Anda untuk mencerminkan pendekatan 80/20 Anda. Negosiasikan diskon volume dengan penyedia utama Anda berdasarkan pengeluaran terkonsentrasi, dan pertahankan fleksibilitas dengan penyedia sekunder untuk kasus penggunaan tertentu. Pendekatan ini memaksimalkan leverage pembelian Anda dan biasanya menghasilkan harga keseluruhan yang lebih baik daripada membagi pengeluaran Anda secara merata.
- Sejajarkan strategi bakat Anda dengan pendekatan 80/20 Anda. Investasikan dalam mengembangkan keahlian mendalam dengan layanan penyedia utama Anda sambil mempertahankan pengetahuan yang cukup tentang platform sekunder untuk mendukung beban kerja tertentu. Strategi bakat terfokus ini meningkatkan produktivitas, mempercepat pengiriman, dan mengurangi risiko kesenjangan keterampilan kritis.
- Ukur hasil bisnis dari strategi multicloud Anda secara teratur. Lacak metrik yang menunjukkan nilai yang diperoleh dari masing-masing penyedia dan sesuaikan distribusi Anda jika perlu. Tujuannya bukan untuk menghindari multicloud sepenuhnya tetapi untuk mengimplementasikannya secara strategis di mana beban kerja tertentu benar-benar mendapat manfaat dari kemampuan yang unik untuk penyedia lain.

Kesimpulan

Paper ini telah menguraikan sembilan prinsip utama untuk mengembangkan strategi multicloud yang efektif. Organizations mencapai kesuksesan terbesar melalui pendekatan cloud primer dengan penggunaan strategis penyedia tambahan di mana kebutuhan bisnis tertentu menuntutnya. Pendekatan 80/20 kami telah menggambarkan keseimbangan fokus dengan fleksibilitas dan memungkinkan organisasi untuk mengembangkan keahlian yang lebih dalam, mempertahankan hubungan penyedia yang lebih kuat, dan membangun bakat yang lebih berharga sambil tetap menangani persyaratan multicloud yang sah.

Implementasi multicloud yang sukses membutuhkan penilaian yang jelas tentang kebutuhan bisnis alih-alih mengikuti tren industri. Perusahaan harus menetapkan tata kelola yang kuat, menjaga keamanan sebagai prioritas utama, menghindari penyebaran beban kerja yang terhubung ke seluruh penyedia, menjaga aplikasi dengan data transaksional mereka, mengenali batasan wadah, dan memelihara Cloud Center of Excellence yang terpadu namun terspesialisasi.

AWS Pendekatan terhadap cloud pada dasarnya dibangun di atas pilihan pelanggan dan interoperabilitas. Kami telah merancang alat dan layanan kami untuk bekerja dengan mulus di seluruh lingkungan karena kami memahami bahwa kebutuhan bisnis Anda seringkali melampaui satu penyedia. Dari solusi konektivitas hybrid hingga orkestrasi kontainer yang mencakup lingkungan, AWS menghadirkan kemampuan yang membantu Anda beroperasi secara efektif di seluruh lanskap teknologi Anda.

Alih-alih memaksa Anda untuk menjadi ahli di berbagai platform, AWS menyederhanakan manajemen multicloud melalui alat intuitif dan antarmuka yang konsisten. Kami fokus untuk menghilangkan kompleksitas sehingga Anda dapat fokus pada inovasi. Kemampuan ini membantu Anda menerapkan strategi multicloud Anda dengan istilah Anda sendiri—apakah itu berarti menggunakan AWS secara eksklusif atau menggunakan spesifik bersama lingkungan lain. Layanan AWS

Cloud harus memberdayakan strategi bisnis Anda, bukan membatasinya. Dengan menerapkan prinsip-prinsip yang diuraikan dalam paper ini dan memanfaatkan kemampuan AWS interoperabilitas, Anda dapat membangun pendekatan cloud yang memaksimalkan nilai, meminimalkan kompleksitas yang tidak perlu, dan memposisikan organisasi Anda untuk kesuksesan jangka panjang dalam lingkungan bisnis yang dinamis saat ini.

Untuk mempelajari lebih lanjut tentang AWS solusi yang dapat membantu menyederhanakan manajemen di lingkungan hybrid dan multicloud, lihat [AWS solusi](#) untuk multicloud.

Sumber daya

Referensi

- [Menggunakan Cloud Center of Excellence \(CCOE\) untuk Mengubah Seluruh Perusahaan](#) (AWS posting blog)
- [AWS Kerangka Well-Architected](#)
- [Mengidentifikasi peluang dengan Cost Optimization Hub](#) (AWS Cost Management dokumentasi)
- [Nilai Bisnis Migrasi ke Amazon Web Services](#) (Grup Hackett, Februari 2022)
- [Transfer data gratis ke internet saat pindah dari AWS](#) (posting AWS blog)

Alat

- [Zonal autoshift - Secara otomatis mengalihkan lalu lintas Anda dari Availability Zone ketika kami mendeteksi potensi masalah](#) (AWS posting blog)
- [AWS solusi untuk multicloud](#)

AWS Mitra

- [AWS Cloud Kompetensi operasi](#)

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	—	September 3, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat [kontrol akses berbasis atribut](#).

layanan abstrak

Lihat [layanan terkelola](#).

ASAM

Lihat [atomisitas, konsistensi, isolasi, daya tahan](#).

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat [kecerdasan buatan](#).

AIOps

Lihat [operasi kecerdasan buatan](#).

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

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

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

[Bot](#) yang dimaksudkan untuk mengganggu atau menyebabkan kerugian bagi individu atau organisasi.

BCP

Lihat [perencanaan kontinuitas bisnis](#).

grafik perilaku

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

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga [endianness](#).

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

penyebaran kenari

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

CCoE

Lihat [Cloud Center of Excellence](#).

CDC

Lihat [mengubah pengambilan data](#).

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat [integrasi berkelanjutan dan pengiriman berkelanjutan](#).

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat [database manajemen konfigurasi](#).

repositori kode

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

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

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

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

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

CV

Lihat [visi komputer](#).

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat [bahasa definisi database](#).

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat [lingkungan](#).

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat [bahasa manipulasi basis data](#).

desain berbasis domain

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

DR

Lihat [pemulihan bencana](#).

deteksi drift

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

DVSM

Lihat [pemetaan aliran nilai pengembangan](#).

E

EDA

Lihat [analisis data eksplorasi](#).

EDI

Lihat [pertukaran data elektronik](#).

komputasi tepi

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

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat [Apa itu Pertukaran Data Elektronik](#).

enkripsi

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

kunci enkripsi

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

endianness

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

titik akhir

Lihat [titik akhir layanan](#).

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

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

epik

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

ERP

Lihat [perencanaan sumber daya perusahaan](#).

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat [cabang](#).

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol [akses berbutir halus](#).

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

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

FM

Lihat [model pondasi](#).

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat [pembatasan geografis](#).

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

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

strategi greenfield

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

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (OU). 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.

|

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.

IloT

Lihat [Internet of Things industri](#).

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAC)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretasi

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

IoT

Lihat [Internet of Things](#).

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat [perpustakaan informasi TI](#).

ITSM

Lihat [manajemen layanan TI](#).

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat [kontrol akses berbasis label](#).

hak istimewa paling sedikit

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

angkat dan geser

Lihat [7 Rs](#).

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga [endianness](#).

LLM

Lihat [model bahasa besar](#).

lingkungan yang lebih rendah

Lihat [lingkungan](#).

M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat [Program Percepatan Migrasi](#).

mekanisme

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

akun anggota

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

MES

Lihat [sistem eksekusi manufaktur](#).

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

informasi selengkapnya, lihat [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server](#).

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat [pembelajaran mesin](#).

modernisasi

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

penilaian kesiapan modernisasi

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

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

aplikasi monolitik (monolit)

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

MPA

Lihat [Penilaian Portofolio Migrasi](#).

MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat [kontrol akses asal](#).

OAI

Lihat [identitas akses asal](#).

OCM

Lihat [manajemen perubahan organisasi](#).

migrasi offline

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

OI

Lihat [integrasi operasi](#).

OLA

Lihat [perjanjian tingkat operasional](#).

migrasi online

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

OPC-UA

Lihat [Komunikasi Proses Terbuka - Arsitektur Terpadu](#).

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat [panduan integrasi operasi](#).

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat [tinjauan kesiapan operasional](#).

OT

Lihat [teknologi operasional](#).

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

Lihat informasi yang [dapat diidentifikasi secara pribadi](#).

buku pedoman

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

PLC

Lihat [pengontrol logika yang dapat diprogram](#).

PLM

Lihat [manajemen siklus hidup produk](#).

kebijakan

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

persistensi poliglot

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

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat [Mengevaluasi kesiapan migrasi](#).

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona host pribadi

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

kontrol proaktif

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

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat [lingkungan](#).

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

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

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

LAP

Lihat [Retrieval Augmented Generation](#).

ransomware

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

Matriks RASCI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

RCAC

Lihat [kontrol akses baris dan kolom](#).

replika baca

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

arsitek ulang

Lihat [7 Rs](#).

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat [7 Rs](#).

Region

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

regresi

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

rehost

Lihat [7 Rs](#).

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat [7 Rs](#).

memplatform ulang

Lihat [7 Rs](#).

pembelian kembali

Lihat [7 Rs](#).

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat [7 Rs](#).

pensiun

Lihat [7 Rs](#).

Retrieval Augmented Generation (RAG)

Teknologi [AI generatif](#) di mana [LLM](#) mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin

melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat [Apa itu RAG](#).

rotasi

Proses memperbarui [rahasia](#) secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensial.

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat [tujuan titik pemulihan](#).

RTO

Lihat [tujuan waktu pemulihan](#).

buku runbook

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

D

SAML 2.0

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

PENIPUAN

Lihat [kontrol pengawasan dan akuisisi data](#).

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadatanya. 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 kredensial.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator [tingkat layanan](#).

model tanggung jawab bersama

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

SIEM

Lihat [informasi keamanan dan sistem manajemen acara](#).

titik kegagalan tunggal (SPOF)

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

SLA

Lihat [perjanjian tingkat layanan](#).

SLI

Lihat [indikator tingkat layanan](#).

SLO

Lihat [tujuan tingkat layanan](#).

split-and-seed model

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

SPOF

Lihat [satu titik kegagalan](#).

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

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

variabel target

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

daftar tugas

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

lingkungan uji

Lihat [lingkungan](#).

pelatihan

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

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

gerbang transit

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian:

ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

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

lingkungan atas

Lihat [lingkungan](#).

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat [menulis sekali, baca banyak](#).

WQF

Lihat [AWS Kerangka Kualifikasi Beban Kerja](#).

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan [zero-day](#).

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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