



Menerapkan Kerangka Kerja AWS Well-Architected ke Aplikasi Amazon WorkSpaces

AWS Bimbingan Preskriptif



AWS Bimbingan Preskriptif: Menerapkan Kerangka Kerja AWS Well-Architected ke Aplikasi Amazon WorkSpaces

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
Audiens yang dituju	1
Tujuan	2
Pilar keunggulan operasional	3
Mengatur tim di sekitar hasil bisnis	3
Menerapkan observabilitas untuk wawasan yang dapat ditindaklanjuti	4
Otomatiskan dengan aman jika memungkinkan	5
Buat perubahan yang sering, kecil, dan reversibel	7
Perbaiki prosedur operasi sesering mungkin	8
Mengantisipasi kegagalan	9
Belajar dari semua peristiwa dan metrik operasional	10
Gunakan layanan terkelola	10
Pilar keamanan	11
Menerapkan fondasi identitas yang kuat	11
Pertahankan ketertelusuran	12
Terapkan keamanan di semua lapisan	13
Mengotomatiskan praktik terbaik keamanan	14
Jauhkan orang dari data	14
Mempersiapkan acara keamanan	15
Pilar keandalan	17
Secara otomatis pulih dari kegagalan	17
Uji prosedur pemulihan	18
Skala horizontal untuk meningkatkan ketersediaan beban kerja agregat	18
Berhenti menebak kapasitas	19
Kelola perubahan melalui otomatisasi	19
Pilar efisiensi performa	20
Demokratisasi teknologi canggih	20
Menjadi global dalam hitungan menit	21
Gunakan arsitektur tanpa server	21
Eksperimen lebih sering	22
Pertimbangkan simpati mekanis	22
Pilar optimasi biaya	24
Menerapkan manajemen keuangan cloud	24
Tambahkan model konsumsi	24

Ukur efisiensi keseluruhan	25
Berhenti menghabiskan uang untuk angkat berat yang tidak berdiferensiasi	25
Analisis dan atribut pengeluaran	26
Pilar keberlanjutan	27
Pahami dampak Anda	27
Menetapkan tujuan keberlanjutan	27
Maksimalkan pemanfaatan	28
Mengantisipasi dan mengadopsi penawaran perangkat keras dan perangkat lunak baru yang lebih efisien	28
Layanan terkelola yang digunakan	28
Mengurangi dampak hilir beban kerja cloud Anda	29
Sumber Daya	30
AWS dokumentasi	30
AWS posting blog	30
Riwayat dokumen	32
Glosarium	33
#	33
A	34
B	37
C	39
D	42
E	46
F	48
G	50
H	51
I	52
L	55
M	56
O	61
P	63
Q	66
R	67
D	70
T	74
U	75
V	76

W	76
Z	77
.....	Ixxix

Menerapkan Kerangka Kerja AWS Well-Architected ke Aplikasi Amazon WorkSpaces

Mohamed Wali, Amazon Web Services

Juli 2025 ([sejarah dokumen](#))

[Panduan ini mencakup praktik terbaik untuk menerapkan Kerangka Kerja AWS Well-Architected saat Anda menggunakan Aplikasi Amazon. WorkSpaces](#) WorkSpaces Aplikasi adalah layanan streaming aplikasi yang dikelola sepenuhnya yang memungkinkan Anda melakukan streaming aplikasi desktop ke pengguna tanpa menulis ulang.

AWS Well-Architected Framework membantu arsitek cloud membangun infrastruktur yang aman, berkinerja tinggi, tangguh, dan efisien untuk berbagai aplikasi dan beban kerja. Ini juga menyediakan pendekatan yang konsisten bagi pengguna dan AWS Mitra untuk mengevaluasi arsitektur dan menerapkan desain yang dapat diskalakan.

The AWS Well-Architected Framework dibangun di sekitar enam pilar:

- Keunggulan operasional
- Keamanan
- Keandalan
- Efisiensi kinerja
- Optimalisasi biaya
- Keberlanjutan

Panduan ini membahas bagaimana pilar dan praktik terbaik ini berlaku untuk menggunakan WorkSpaces Aplikasi.

Audiens yang dituju

Panduan ini untuk:

- Arsitek dan insinyur cloud yang merancang dan mengimplementasikan solusi WorkSpaces Aplikasi, dan perlu memastikan bahwa arsitekturnya mengikuti praktik terbaik AWS Well-Architected Framework.

- Tim operasi TI yang mengelola dan memelihara lingkungan WorkSpaces Aplikasi; menangani manajemen armada, penskalaan, dan pemantauan; dan perlu mengoptimalkan biaya dan kinerja.
- Organizations atau bisnis yang sedang mempertimbangkan atau sudah menggunakan WorkSpaces Applications, ingin melakukan streaming aplikasi desktop ke penggunaanya, dan perlu membangun infrastruktur yang aman, berkinerja tinggi, tangguh, dan efisien.

Tujuan

Mengikuti praktik terbaik dalam panduan ini membantu Anda:

- Bangun infrastruktur yang aman, berkinerja tinggi, tangguh, dan efisien untuk streaming aplikasi desktop di aplikasi desktop. AWS Cloud
- Terapkan pendekatan yang konsisten saat mengevaluasi arsitektur WorkSpaces Aplikasi dan menerapkan desain yang dapat diskalakan.

Pilar keunggulan operasional

Operational Excellence (OE) mewakili dedikasi untuk menyusun solusi perangkat lunak berkualitas tinggi yang secara konsisten memenuhi dan melampaui harapan pengguna. [Pilar keunggulan operasional](#) dari AWS Well-Architected Framework mencakup strategi yang telah terbukti untuk organisasi tim yang efektif, desain beban kerja yang kuat, operasi skala besar yang efisien, dan adaptasi tanpa batas terhadap perubahan persyaratan dari waktu ke waktu. Dengan mematuhi prinsip-prinsip ini, organisasi dapat memastikan bahwa sistem mereka tetap tangguh, berkinerja, dan selaras dengan kebutuhan bisnis yang berkembang.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Pemantauan dan observabilitas
- Otomatisasi dan DevOps
- Prosedur dan dokumentasi operasional
- Support dan manajemen insiden

Mengatur tim di sekitar hasil bisnis

Buat model operasi yang selaras dengan cloud dengan komitmen kepemimpinan yang kuat, di mana tujuan bisnis dan indikator kinerja utama (KPIs) mendorong transformasi organisasi melalui orang, proses, dan teknologi yang dioptimalkan.

- Struktur tim. Bangun tim khusus yang selaras dengan hasil streaming aplikasi. Contoh:
 - Tim manajemen gambar bertanggung jawab atas pengemasan aplikasi dan pengoptimalan gambar.
 - Tim operasi armada mengelola kapasitas, kinerja, dan penskalaan.
 - Tim pengalaman pengguna menangani dukungan dan kepuasan pengguna akhir.
- KPIs dan metrik. Tentukan dan lacak metrik yang selaras dengan bisnis seperti:
 - Tarif ketersediaan aplikasi
 - Saatnya menyebarkan aplikasi baru
 - Biaya per jam streaming aplikasi
- Model operasi. Buat proses yang jelas untuk:
 - Orientasi dan pembaruan aplikasi

- Manajemen kapasitas armada
- Penyediaan akses pengguna
- Respon dan resolusi insiden

Menerapkan observabilitas untuk wawasan yang dapat ditindaklanjuti

Menerapkan pemantauan dan observabilitas yang komprehensif untuk melacak KPIs dan kesehatan beban kerja. Prinsip ini memungkinkan keputusan berbasis data dan peningkatan proaktif di seluruh kinerja, keandalan, dan biaya.

- Melaksanakan pemantauan kinerja. Konfigurasi [Amazon CloudWatch](#) ke:
 - Pastikan kapasitas yang cukup untuk memenuhi permintaan. Misalnya, Anda dapat menggunakan metrik berikut:
 - `AvailableCapacity` untuk memantau instans streaming yang tersedia
 - `InUseCapacity` untuk melacak instance yang saat ini digunakan
 - `CapacityUtilization` untuk memantau persentase penggunaan armada
 - Pantau pengalaman dan kinerja pengguna.
 - Identifikasi dan atasi masalah layanan dengan segera.
- Lacak dan analisis laporan penggunaan WorkSpaces Aplikasi.
- Tangkap dan analisis log aplikasi. Untuk informasi selengkapnya, lihat posting AWS blog [Menggunakan Agen Kinesis untuk Linux untuk melakukan streaming log WorkSpaces aplikasi di Aplikasi dan Menggunakan Agen Kinesis untuk Microsoft Windows untuk menyimpan log peristiwa WorkSpaces Aplikasi Windows](#).
- Pantau metrik dan acara WorkSpaces Aplikasi melalui notifikasi obrolan. Untuk informasi lebih lanjut, lihat posting AWS blog [Memantau dan mengotomatiskan komputasi pengguna AWS akhir \(EUC\) dengan AWS Chatbot](#).
- Aktifkan manajemen sesi proaktif melalui isyarat visual. Untuk informasi selengkapnya, lihat posting AWS blog [Kedaluwarsa sesi tampilan dan penghitung waktu mundur di Aplikasi Amazon WorkSpaces](#)
- Buat visualisasi untuk pola dan tren penggunaan. Untuk informasi selengkapnya, lihat posting AWS blog [Ingest dan visualisasikan laporan penggunaan WorkSpaces Aplikasi Amazon di Layanan Amazon OpenSearch](#).

- Manfaatkan toolkit EUC untuk memantau sesi aktif, melacak inventaris armada, dan menghasilkan laporan sesi (ekspor CSV). Untuk informasi selengkapnya, lihat posting AWS blog [Gunakan EUC Toolkit untuk mengelola WorkSpaces Aplikasi Amazon dan Amazon](#). WorkSpaces

Otomatiskan dengan aman jika memungkinkan

Terapkan prinsip infrastruktur sebagai kode (IaC) untuk mengotomatiskan semua aspek operasi beban kerja Anda. Gunakan pagar pembatas untuk membantu memastikan eksekusi yang aman dan konsisten sekaligus mengurangi intervensi manual.

- Otomatiskan pembuatan dan konfigurasi gambar WorkSpaces Aplikasi dengan menggunakan Image Assistant CLI. Untuk informasi selengkapnya, lihat [Membuat gambar WorkSpaces Aplikasi Amazon Anda secara terprogram menggunakan operasi CLI Asisten Gambar](#) di dokumentasi Aplikasi. WorkSpaces
 - Instalasi aplikasi: Gunakan Image Assistant CLI untuk mengotomatiskan instalasi aplikasi selama pembuatan gambar.
 - Pembuatan gambar: Buat gambar WorkSpaces Aplikasi secara terprogram dengan menggunakan perintah Image Assistant CLI.
 - Manajemen konfigurasi: Otomatiskan konfigurasi pengaturan aplikasi default dan parameter peluncuran.
- Otomatiskan kustomisasi gambar WorkSpaces Aplikasi. Untuk informasi selengkapnya, lihat posting AWS blog [Membuat gambar Windows WorkSpaces Aplikasi yang disesuaikan secara otomatis](#).
- Terapkan IaC untuk menyebarkan infrastruktur dan komponen aplikasi untuk WorkSpaces Aplikasi. Untuk informasi selengkapnya, lihat posting AWS blog [Otomatisasi infrastruktur dan penerapan aplikasi untuk WorkSpaces Aplikasi Amazon dengan Terraform](#).
- Menerapkan proses otomatis untuk manajemen armada, termasuk:
 - Penskalaan armada berdasarkan permintaan. Konfigurasi kebijakan penskalaan otomatis untuk menyesuaikan kapasitas armada secara otomatis berdasarkan metrik pemanfaatan. Untuk informasi selengkapnya, lihat posting AWS blog [Gunakan AWS Lambda untuk menyesuaikan langkah penskalaan dan ambang batas untuk Aplikasi Amazon](#). WorkSpaces
 - Pembaruan gambar dasar. Manfaatkan pembaruan otomatis ke gambar dasar WorkSpaces Aplikasi yang disediakan oleh AWS.

- Optimalisasi kapasitas. Siapkan ambang batas penskalaan otomatis untuk mengoptimalkan penggunaan sumber daya berdasarkan pola permintaan.
- Konfigurasi pagar pembatas untuk mengotomatiskan kontrol keselamatan:
 - Batas ukuran armada maksimum. Tetapkan batas atas kapasitas armada untuk mencegah penyediaan berlebih.
 - Konfigurasi kebijakan penskalaan. Menerapkan penskalaan langkah atau kebijakan penskalaan pelacakan target dengan ambang batas yang sesuai.
 - Kuota layanan. Gunakan kuota AWS layanan sebagai batas bawaan untuk mencegah alokasi sumber daya yang berlebihan.
 - Perlindungan skala. Konfigurasi perlindungan penskalaan untuk mencegah penghapusan instance aktif selama peristiwa penskalaan.
- Lakukan pengujian dan validasi, termasuk pembuat gambar, armada, dan pengujian integrasi.
 - Pengujian pembuat gambar:
 - Uji aplikasi langsung di antarmuka pembuat gambar.
 - Verifikasi peluncuran dan fungsionalitas aplikasi.
 - Uji pengaturan dan konfigurasi pengguna.
 - Validasi kompatibilitas aplikasi.
 - Pengujian armada:
 - Uji sesi streaming dari perangkat klien yang berbeda.
 - Verifikasi hak dan akses pengguna.
 - Validasi kinerja aplikasi.
 - Uji pengalaman pengguna untuk elemen dan operasi seperti clipboard, transfer file, dan pencetakan.
 - Pengujian integrasi:
 - Uji Active Directory atau otentikasi berbasis SAM 2.0.
 - Uji folder rumah dan penyimpanan persisten.
 - Uji hak aplikasi.
 - Uji pengalihan perangkat USB (jika dikonfigurasi).
- Gunakan WorkSpaces Applications Applications Manager untuk mengotomatiskan pengemasan dan penyebaran aplikasi. Untuk informasi selengkapnya, lihat posting AWS blog [Streamline onboarding aplikasi dengan manajer aplikasi untuk Aplikasi Amazon WorkSpaces](#).

- Otomatiskan penerapan versi aplikasi baru dengan menggunakan pipeline continuous integration dan continuous delivery (CI/CD). Untuk informasi selengkapnya, lihat posting AWS blog [Screening Eagle: Optimalkan CI/CD dan pengalaman pengguna akhir di WorkSpaces Aplikasi Amazon](#).

Buat perubahan yang sering, kecil, dan reversibel

Bangun beban kerja yang digabungkan secara longgar dan dapat diskalakan yang memungkinkan penerapan otomatis skala kecil yang sering dengan risiko minimal dan kemampuan rollback yang mudah.

- Untuk pembaruan gambar, gunakan pembuatan gambar berversi dan pembaruan tambahan.
 - Pembuatan gambar berversi:
 - Buat gambar baru untuk setiap rangkaian perubahan dengan menggunakan pembuat gambar.
 - Pertahankan beberapa versi gambar untuk mendukung skenario rollback.
 - Gunakan [strategi AWS penandaan](#) untuk melacak versi dan atribut gambar.
 - Pembaruan tambahan:
 - Buat perubahan kecil dan bertahap pada aplikasi atau konfigurasi.
 - Uji pembaruan secara menyeluruh di pembuat gambar sebelum Anda membuat gambar baru.
 - Dokumentasikan semua perubahan yang Anda buat di setiap versi gambar baru.
 - Untuk pembaruan armada kontrol:
 - Buat armada baru dengan gambar yang diperbarui untuk pengujian.
 - Ubah atribut armada yang ada tanpa mengganggu sesi aktif.
 - Menetapkan prosedur manajemen perubahan untuk dokumentasi, protokol pengujian, alur kerja persetujuan, dan proses pemantauan.
 - Dokumentasi:
 - Pertahankan log perubahan terperinci untuk semua pembaruan gambar dan armada.
 - Dokumen prosedur pengujian dan hasil untuk setiap perubahan.
 - Gunakan [AWS CloudTrail](#) untuk melacak dan mengaudit perubahan konfigurasi.
 - Protokol pengujian:
 - Menetapkan proses pengujian yang komprehensif untuk semua perubahan.
 - Sertakan fungsionalitas aplikasi, kinerja, dan pengujian pengalaman pengguna.
-
- Buat perubahan yang sering, kecil, dan reversibel
- Lakukan pengujian di pembuat gambar sebelum Anda membuat gambar baru.

- Lakukan pengujian tambahan pada armada non-produksi sebelum penyebaran penuh.
- Alur kerja persetujuan:
 - Menerapkan proses persetujuan untuk perubahan lingkungan produksi.
 - Tentukan kriteria untuk perubahan yang memerlukan persetujuan versus pembaruan standar.
 - Menetapkan peran dan tanggung jawab untuk persetujuan perubahan.
- Pemantauan dan validasi:
 - Gunakan [Amazon CloudWatch](#) untuk memantau kinerja armada dan aplikasi setelah perubahan.
 - Siapkan peringatan untuk metrik utama untuk mengidentifikasi masalah dengan cepat setelah pembaruan.
 - Lakukan tinjauan pasca-implementasi untuk memvalidasi keberhasilan perubahan dan mengumpulkan pembelajaran.

Perbaiki prosedur operasi sesering mungkin

Terus meningkatkan prosedur operasional melalui tinjauan rutin, pembaruan, dan keterlibatan tim untuk menjaga agar semua pemangku kepentingan tetap terinformasi dan selaras dengan praktik terbaik.

- Manajemen dokumentasi. Pertahankan dokumentasi prosedur WorkSpaces Aplikasi yang dikendalikan versi saat ini di lokasi pusat untuk memastikan konsistensi operasional dan berbagi pengetahuan di seluruh tim.
 - Dokumentasi yang diperlukan: Menjaga up-to-date dokumentasi untuk operasi WorkSpaces Aplikasi penting untuk pembuatan dan manajemen gambar, operasi armada, dan pemecahan masalah.
 - Tinjauan operasional: Memantau dan meninjau aspek operasional utama, termasuk metrik kinerja dan manajemen insiden.
- Perbaiki berkelanjutan. Secara sistematis meningkatkan operasi WorkSpaces Aplikasi dengan memasukkan Layanan AWS pembaruan, metrik operasional, dan praktik terbaik yang dipelajari ke dalam prosedur standar.
 - Pembaruan layanan: Memantau catatan rilis WorkSpaces Aplikasi untuk fitur baru, peningkatan layanan, pembaruan keamanan, dan ketersediaan Regional.

- Praktik terbaik: Tinjau dan gabungkan pembaruan AWS Well-Architected Framework WorkSpaces , praktik terbaik Aplikasi, arsitektur referensi AWS , dan rekomendasi keamanan AWS
- Manajemen pengetahuan: Memelihara dan memperbarui prosedur operasi standar, runbook, panduan pemecahan masalah, dan dokumentasi dukungan pengguna.

Mengantisipasi kegagalan

Lakukan pengujian skenario kegagalan secara teratur untuk memahami risiko, memvalidasi prosedur respons, dan meningkatkan kesiapan tim untuk menangani insiden nyata.

- Pengujian kegagalan. Secara teratur mensimulasikan dan menguji kegagalan seperti kelelahan kapasitas armada, kegagalan peluncuran aplikasi, dan masalah konektivitas jaringan.
 - Kelelahan kapasitas armada:
 - Pantau dan uji perilaku penskalaan armada saat mendekati batas kapasitas.
 - Konfigurasi CloudWatch alarm untuk CapacityUtilization dan AvailableCapacity metrik.
 - Menerapkan prosedur untuk menangani kendala kapasitas selama penggunaan puncak.
 - Kegagalan peluncuran aplikasi:
 - Uji perilaku peluncuran aplikasi pada instance streaming.
 - Validasi akses dan kinerja aplikasi di berbagai konfigurasi armada.
 - Masalah konektivitas jaringan:
 - Uji kinerja sesi streaming di berbagai kondisi jaringan.
 - Pantau StreamingSessionLatency masalah kualitas koneksi.
 - Pastikan konfigurasi pengaturan VPC dan grup keamanan yang tepat.
- Prosedur pemulihan. Mengembangkan dan menguji prosedur untuk:
 - Failover armada antara AWS Zona Ketersediaan. Selain itu, mendokumentasikan prosedur untuk penskalaan kapasitas armada, mengelola pembaruan armada, dan menanggapi masalah kesehatan misalnya.
 - Manajemen data pengguna:
 - Konfigurasi dan uji solusi [persistensi dan penyimpanan pengaturan aplikasi](#) untuk folder rumah di Amazon Simple Storage Service (Amazon S3) untuk armada Windows dan sistem file bersama di Amazon Elastic File System (Amazon EFS) untuk armada Linux.

- Validasi sinkronisasi data antar sesi.
- Kontinuitas layanan. Menjaga prosedur untuk membuat instance armada baru, mengelola pembaruan gambar, dan menangani pemutusan sesi.
- Manajemen risiko. Identifikasi dan mitigasi:
 - Kendala kapasitas dengan menetapkan kapasitas minimum armada yang sesuai, mengonfigurasi kebijakan penskalaan otomatis berdasarkan pola permintaan, dan memantau tren pemanfaatan armada dengan menggunakan CloudWatch metrik seperti `InUseCapacity` dan `AvailableCapacity`.
 - Kemacetan kinerja dengan melacak metrik utama seperti `StreamingSessionLatency` dan mengonfigurasi alarm yang sesuai. CloudWatch

Belajar dari semua peristiwa dan metrik operasional

Menumbuhkan budaya perbaikan berkelanjutan dengan berbagi pelajaran yang dipetik dari peristiwa operasional dan kegagalan di seluruh organisasi. Tekankan dampaknya pada hasil bisnis.

- Analisis peristiwa. Mendokumentasikan dan menganalisis gangguan layanan, penurunan kinerja, keluhan pengguna, dan masalah kapasitas.
- Ulasan metrik. Menganalisis pola penggunaan, tren kinerja, metrik biaya, dan data kepuasan pengguna secara teratur.
- Berbagi pengetahuan. Menetapkan proses untuk sesi pembelajaran tim, dokumentasi praktik terbaik, transfer pengetahuan lintas tim, dan retrospektif insiden.

Gunakan layanan terkelola

Minimalkan overhead operasional dengan menggunakan layanan AWS terkelola dan membangun prosedur standar di sekitarnya. Integrasikan dengan layanan AWS terkelola berikut:

- [AWS Systems Manager](#) untuk otomatisasi
- [Amazon CloudWatch](#) untuk pemantauan
- [AWS Identity and Access Management \(IAM\)](#) untuk kontrol akses
- [Amazon S3](#) untuk penyimpanan pengguna untuk armada Windows
- [Amazon EFS](#) untuk penyimpanan pengguna untuk armada Linux
- [AWS Directory Service](#) untuk otentikasi pengguna

Pilar keamanan

[Pilar keamanan](#) dari AWS Well-Architected Framework berfokus pada memanfaatkan kemampuan cloud untuk membantu membangun mekanisme perlindungan yang kuat untuk informasi, infrastruktur, dan sumber daya Anda. Prinsip-prinsip ini membantu meningkatkan postur keamanan Anda secara keseluruhan sekaligus memungkinkan inovasi.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Integritas dan kerahasiaan data
- Mengelola izin pengguna
- Membangun kontrol untuk mendeteksi peristiwa keamanan

Menerapkan fondasi identitas yang kuat

Gunakan izin minimum yang diperlukan untuk mengakses AWS sumber daya sambil memusatkan manajemen identitas dan menghindari kredensi jangka panjang.

- Berikan izin istimewa paling sedikit untuk sumber daya WorkSpaces Aplikasi:
 - Buat peran IAM khusus untuk armada WorkSpaces Aplikasi dengan izin minimal yang diperlukan.
 - Konfigurasi izin IAM terbatas untuk pembuat gambar.
 - Membatasi akses administratif ke fungsi manajemen WorkSpaces Aplikasi.
 - Tentukan izin granular untuk manajemen tumpukan dan armada.
- Menerapkan mekanisme otentikasi pengguna yang tepat:
 - Konfigurasi federasi SAMP 2.0 untuk integrasi penyedia identitas perusahaan.
 - Siapkan [AWS IAM Identity Center](#) untuk manajemen pengguna.
 - Gunakan pialang identitas khusus hanya jika diperlukan untuk skenario otentikasi tertentu.
 - Menerapkan otentikasi multi-faktor (MFA) jika didukung.
- Kontrol akses pengguna ke aplikasi:
 - Konfigurasi hak aplikasi untuk membatasi akses ke aplikasi tertentu.
 - Buat grup penugasan aplikasi berdasarkan peran pengguna.
 - Kelola akses aplikasi melalui izin tumpukan.

- Menerapkan kebijakan sesi untuk mengontrol perilaku aplikasi.
- Amankan sesi pengguna dengan kontrol yang sesuai:
 - Konfigurasi kebijakan batas waktu sesi.
 - Setel tindakan batas waktu pemutusan sambungan.
 - Menerapkan persyaratan ketekunan sesi.
 - Kontrol izin pengalihan sistem file.
- Konfigurasi otentikasi berbasis sertifikat untuk Aplikasi WorkSpaces Untuk informasi selengkapnya, lihat posting AWS blog [Sederhanakan otentikasi berbasis sertifikat untuk WorkSpaces Aplikasi dan WorkSpaces dengan AWS Private CA Konektor untuk Direktori Aktif](#).
- Gunakan tag sesi untuk menerapkan kontrol akses berbutir halus. Untuk informasi selengkapnya, lihat posting AWS blog [Gunakan tag sesi untuk menyederhanakan izin WorkSpaces Aplikasi](#).

Pertahankan ketertelusuran

Menerapkan pemantauan real-time dan sistem respons otomatis untuk semua perubahan dan aktivitas lingkungan.

- Konfigurasi [CloudWatch logging](#) untuk log aplikasi untuk memantau peristiwa khusus aplikasi, termasuk peluncuran aplikasi, crash, dan kesalahan. Konfigurasi log sesi untuk melacak informasi sesi streaming, termasuk sesi dimulai, berhenti, dan peristiwa koneksi pengguna.
- [Aktifkan CloudTrail untuk mencatat semua panggilan API WorkSpaces Aplikasi](#) dan untuk melacak peristiwa manajemen seperti pembuatan dan modifikasi armada, operasi pembuat gambar, konfigurasi tumpukan, dan aktivitas manajemen pengguna.
- Memantau aktivitas instance WorkSpaces Aplikasi:
 - Konfigurasi pencatatan instance untuk menangkap peristiwa tingkat sistem.
 - Lacak peluncuran dan kegagalan aplikasi.
 - Memantau penggunaan dan kinerja sumber daya sistem.
- Lacak aktivitas pengguna:
 - Pantau upaya dan kegagalan otentikasi pengguna. Gunakan CloudWatch metrik dan CloudWatch Log untuk melacak upaya login pengguna, waktu mulai dan akhir sesi, dan peristiwa pemutusan sesi.

- Lacak pola penggunaan aplikasi. [Aktifkan laporan penggunaan WorkSpaces Aplikasi](#) untuk mengambil informasi seperti durasi sesi, waktu mulai dan akhir, jenis instans yang digunakan, dan aplikasi yang diakses.
- Rekam aktivitas sistem file melalui folder rumah yang diaktifkan.
- Konfigurasi pengaturan clipboard dan operasi pencetakan untuk mencapai tujuan pencegahan kehilangan data Anda.
- Konfigurasi [CloudWatchalarm](#) untuk metrik terkait keamanan seperti otentikasi pengguna yang gagal, pola sesi yang tidak biasa, dan pelanggaran akses sumber daya.
- Gunakan toolkit EUC untuk melacak sesi dan status aktif, memantau alamat IP untuk sesi aktif yang sedang digunakan, dan mengeksport data sesi untuk audit. Untuk informasi selengkapnya, lihat posting AWS blog [Gunakan toolkit EUC untuk mengelola WorkSpaces Aplikasi Amazon dan Amazon. WorkSpaces](#)

Terapkan keamanan di semua lapisan

Terapkan beberapa lapisan kontrol keamanan di semua komponen infrastruktur Anda, dari tepi jaringan hingga kode aplikasi.

- Konfigurasi keamanan lapisan jaringan:
 - Menerapkan aturan kelompok keamanan yang ketat.
 - Tempatkan instance armada WorkSpaces Aplikasi di subnet pribadi yang tidak memiliki akses internet langsung. Kontrol akses internet melalui perangkat NAT.
 - Gunakan titik akhir virtual private cloud (VPC) untuk mengakses yang didukung Layanan AWS (seperti Amazon S3).
 - Menerapkan daftar kontrol akses jaringan (ACLs) sebagai lapisan keamanan jaringan tambahan.
 - Batasi akses port streaming (TCP 8443 untuk HTTPS dan WebSocket Aman) ke rentang IP tertentu.
- Konfigurasi keamanan lapisan akses:
 - Menerapkan kebijakan batas waktu sesi untuk secara otomatis memutuskan sambungan pengguna yang tidak aktif.
 - Menerapkan kontrol akses berbasis atribut dengan menggunakan tag sesi. Untuk informasi selengkapnya, lihat posting AWS blog [Gunakan tag sesi untuk menyederhanakan izin WorkSpaces Aplikasi](#).

- Konfigurasi keamanan lapisan aplikasi:
 - Konfigurasi hak aplikasi untuk mengontrol pengguna mana yang dapat mengakses aplikasi tertentu.
 - Aktifkan kontrol pengalihan sistem file untuk membatasi akses ke drive lokal.
 - Konfigurasi clipboard, transfer file, dan izin pencetakan berdasarkan persyaratan keamanan.
 - Siapkan kontrol akses perangkat USB sesuai dengan kebijakan keamanan.
- Konfigurasi keamanan lapisan gambar:
 - Buat dan pertahankan gambar dasar yang diperkeras yang memenuhi persyaratan keamanan.
 - Tetap perbarui gambar dasar dengan tambalan keamanan terbaru.
 - Konfigurasi pengaturan keamanan Windows dalam gambar dasar.
 - Nonaktifkan layanan dan fitur Windows yang tidak perlu dalam gambar dasar.

Mengotomatiskan praktik terbaik keamanan

Gunakan kontrol keamanan otomatis yang ditentukan kode dalam templat yang dikendalikan versi untuk mengaktifkan penerapan infrastruktur yang aman dan terukur.

- Gunakan infrastruktur sebagai kode (IaC) dengan menggunakan layanan seperti AWS CloudFormation untuk menerapkan konfigurasi keamanan yang konsisten di semua penerapan armada. Untuk informasi selengkapnya, lihat posting AWS blog [Secara otomatis melampirkan grup keamanan tambahan ke WorkSpaces Aplikasi Amazon dan Amazon WorkSpaces](#).
- Otomatiskan proses keamanan pembuatan gambar dengan menggunakan Image Assistant CLI.
- Konfigurasi respons otomatis untuk ambang batas pemanfaatan kapasitas yang terlampaui, upaya akses tidak sah, dan perubahan grup keamanan dengan menggunakan alarm Amazon, EventBridge aturan Amazon CloudWatch, dan fungsi untuk respons otomatis. AWS Lambda

Jauhkan orang dari data

Mengotomatiskan proses penanganan data untuk meminimalkan akses manusia secara langsung dan mengurangi risiko kesalahan atau kesalahan penanganan.

- Konfigurasi hak aplikasi untuk mengontrol pengguna mana yang dapat mengakses aplikasi tertentu.

- Gunakan [framework aplikasi dinamis](#) untuk membangun penyedia aplikasi dinamis agar aplikasi tersedia secara dinamis berdasarkan atribut pengguna.
- Konfigurasi pengalihan sistem file untuk mengontrol drive lokal mana yang dapat diakses pengguna, untuk membatasi akses ke folder tertentu, dan untuk mengelola izin transfer file antara sesi lokal dan streaming.
- Menerapkan pembatasan clipboard untuk menonaktifkan berbagi clipboard antara sesi lokal dan streaming, mengaktifkan aliran clipboard satu arah jika diperlukan, dan mencegah penyalinan data yang tidak sah.
- Konfigurasi persistensi pengaturan aplikasi untuk menyimpan dan memulihkan konfigurasi aplikasi secara otomatis, menghilangkan kebutuhan konfigurasi manual, dan mempertahankan pengalaman pengguna yang konsisten.

Mempersiapkan acara keamanan

Kembangkan dan praktikkan rencana respons insiden dengan menggunakan alat otomatis untuk memungkinkan deteksi cepat, investigasi, dan pemulihan dari peristiwa keamanan.

- Siapkan CloudWatch alarm untuk upaya otentikasi yang gagal, perubahan pada grup keamanan armada, modifikasi konfigurasi gambar, dan pola sesi streaming yang tidak biasa.
- Prosedur respons dokumen untuk skenario keamanan WorkSpaces Aplikasi umum seperti:
 - Upaya akses tidak sah
 - Deteksi: Pantau kegagalan otentikasi.
 - Tanggapan: Cabut hak pengguna, tinjau log sesi, dan perbarui kebijakan akses.
 - Instans streaming yang dikompromikan
 - Deteksi: Pantau perilaku instance.
 - Respons: Hentikan sesi yang terpengaruh, ganti instance armada, dan tinjau konfigurasi grup keamanan.
 - Upaya eksfiltrasi data
 - Deteksi: Pantau aktivitas transfer file.
 - Tanggapan: Tinjau papan klip dan log transfer file, sesuaikan izin transfer file, dan perbarui kebijakan perlindungan data.
- Menerapkan proses pemulihan otomatis untuk penggantian instance armada, pemulihan grup keamanan, konfigurasi ulang akses pengguna, dan pemulihan pengaturan aplikasi.

- Gunakan Layanan AWS untuk manajemen keamanan, seperti AWS Security Hub CSPM untuk temuan keamanan dan Amazon GuardDuty untuk deteksi ancaman.

Pilar keandalan

[Pilar keandalan](#) dari AWS Well-Architected Framework membahas seberapa baik suatu sistem mempertahankan fungsionalitas dan tingkat kinerja yang dimaksudkan selama periode operasional yang diharapkan sepanjang masa pakainya. Ini memberikan pedoman komprehensif untuk membangun dan memelihara sistem yang dapat diandalkan AWS, termasuk strategi untuk pengujian dan validasi di semua tahapan siklus hidup beban kerja.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Manajemen dan penskalaan armada
- Keandalan sesi
- Ketersediaan aplikasi
- Prosedur pemulihan

Secara otomatis pulih dari kegagalan

Pantau nilai bisnis KPIs untuk memicu respons otomatis yang dapat memprediksi, mencegah, atau memulihkan dari kegagalan sebelum berdampak pada operasi.

- Pastikan bahwa alokasi subnet IP Anda memperhitungkan ekspansi dan ketersediaan.
- Pantau CloudWatch metrik penting untuk memastikan ketersediaan dan kinerja layanan, termasuk metrik kapasitas armada seperti `AvailableCapacity` dan `InUseCapacity`, dan metrik kualitas streaming seperti `StreamingSessionLatency`
- Konfigurasi peringatan untuk ambang kapasitas, metrik kesehatan sesi, penurunan kinerja, dan perubahan status kesehatan armada.
- Gunakan kemampuan penskalaan otomatis WorkSpaces Aplikasi bawaan untuk:
 - Konfigurasi kapasitas armada minimum dan maksimum.
 - Tetapkan kebijakan penskalaan berdasarkan pemanfaatan kapasitas.
 - Tentukan ambang batas scale-out dan scale-in berdasarkan metrik pengalaman pengguna dan persyaratan bisnis, bukan hanya metrik teknis.
- Bangun lingkungan pemulihan bencana untuk lingkungan WorkSpaces Aplikasi Anda. Untuk informasi lebih lanjut, lihat posting AWS blog [Pertimbangan pemulihan bencana dengan WorkSpaces Aplikasi Amazon](#).

Uji prosedur pemulihan

Lingkungan cloud memungkinkan pengujian otomatis skenario kegagalan dan prosedur pemulihan. Kemampuan ini membantu Anda mengidentifikasi dan memperbaiki kerentanan sebelum kegagalan nyata terjadi.

- Pengujian pemulihan armada. Menerapkan pengujian pemulihan armada yang komprehensif di beberapa skenario:
 - Simulasikan penghentian instans untuk memverifikasi respons penskalaan otomatis.
 - Validasi pemeliharaan kapasitas minimum armada.
 - Uji waktu penggantian instance dan pengalihan pengguna.
 - Validasi efektivitas kebijakan penskalaan.
 - Uji batas kapasitas armada dan penanganan luapan.
- Pengujian pemulihan sesi. Menerapkan prosedur validasi pemulihan sesi:
 - Uji skenario pemutusan dan sambungkan kembali.
 - Verifikasi pelestarian status aplikasi.
 - Uji berbagai skenario gangguan jaringan.
 - Validasi perilaku batas waktu sesi.
 - Verifikasi persistensi otentikasi pengguna.
 - Verifikasi penanganan penyimpanan sementara.

Skala horizontal untuk meningkatkan ketersediaan beban kerja agregat

Distribusikan beban kerja Anda di beberapa sumber daya yang lebih kecil untuk meminimalkan dampak kegagalan individu dan untuk menghilangkan satu titik kegagalan.

- Menerapkan instance armada di beberapa Availability Zone.
- Konfigurasi kapasitas armada minimum yang sesuai.
- Konfigurasi penskalaan otomatis untuk armada dan tetapkan ambang batas penskalaan yang sesuai.
- Pantau pemanfaatan kapasitas di seluruh armada.

- Menerapkan tumpukan WorkSpaces Aplikasi di beberapa Wilayah. Untuk informasi selengkapnya, lihat posting AWS blog [Optimalkan pengalaman pengguna dengan perutean berbasis latensi untuk Aplikasi Amazon](#). WorkSpaces

Berhenti menebak kapasitas

Gunakan kemampuan penskalaan otomatis cloud untuk menyesuaikan sumber daya secara dinamis berdasarkan permintaan. Ini membantu mencegah kejenuhan sumber daya sambil mempertahankan efisiensi optimal.

- Pantau metrik kunci seperti `CapacityUtilization`, `AvailableCapacity`, dan `InUseCapacity` untuk memahami kebutuhan kapasitas.
- Lacak tren pemanfaatan armada di berbagai periode waktu. Pantau pola harian, variasi mingguan, tren bulanan, dan puncak musiman.
- Siapkan kebijakan penskalaan dan konfigurasi ambang batas penskalaan.
- Pastikan ada celah yang cukup antara kuota saat ini dan penggunaan maksimum untuk mengakomodasi failover.
- Mengakomodasi kuota dan kendala layanan tetap melalui arsitektur Anda.

Kelola perubahan melalui otomatisasi

Menerapkan perubahan infrastruktur melalui otomatisasi, termasuk perubahan yang dikendalikan versi pada kode otomatisasi itu sendiri.

- Gunakan IAc untuk konfigurasi armada.
- Menerapkan kebijakan penskalaan yang konsisten.
- Gunakan [Image Assistant CLI untuk pembuatan](#) gambar yang konsisten.

Pilar efisiensi performa

[Pilar efisiensi kinerja](#) dari AWS Well-Architected Framework berfokus pada pengoptimalan penggunaan sumber daya cloud untuk memenuhi atau melampaui tujuan kinerja sambil memastikan kemampuan beradaptasi terhadap permintaan yang berfluktuasi dan teknologi yang muncul. Ini menekankan pentingnya sistem fine-tuning terus menerus untuk mempertahankan efisiensi puncak dalam lingkungan cloud yang dinamis.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Pemilihan dan pengoptimalan tipe instans
- Pengoptimalan kinerja streaming
- Manajemen kapasitas armada

Demokratisasi teknologi canggih

Manfaatkan layanan yang dikelola vendor cloud untuk teknologi yang kompleks sehingga tim Anda dapat fokus pada pengembangan produk alih-alih manajemen infrastruktur.

- Konfigurasi jenis instans yang sesuai berdasarkan persyaratan aplikasi:
 - Pilih instans berkemampuan GPU untuk aplikasi intensif grafis.
 - Pilih [keluarga GPU](#) yang sesuai (seperti Graphics G4dn atau Graphics G5) berdasarkan kebutuhan aplikasi.
- Pilih dan konfigurasi salah satu metode otentikasi berikut:
 - Siapkan integrasi dengan penyedia identitas berbasis SAMP 2.0.
 - Konfigurasi pengaturan kumpulan pengguna.
 - Integrasikan dengan [AWS Directory Service](#).
- Aktifkan dan konfigurasi opsi penyimpanan berdasarkan kebutuhan pengguna:
 - Siapkan folder rumah di [Amazon S3](#) untuk armada berbasis Windows.
 - Siapkan sistem file bersama di [Amazon EFS](#) untuk armada berbasis Linux.
 - Konfigurasi izin penyimpanan persisten.
 - Aktifkan ketekunan pengaturan aplikasi.

Menjadi global dalam hitungan menit

Gunakan penyebaran multi-wilayah untuk meningkatkan pengalaman pengguna global melalui pengurangan latensi.

- Konfigurasi armada dalam beberapa Wilayah AWS dengan menerapkan armada di Wilayah yang paling dekat dengan pengguna Anda sambil membuat tumpukan terpisah untuk setiap Wilayah.
- Terapkan pengalihan lintas wilayah untuk secara otomatis mengarahkan pengguna WorkSpaces Aplikasi ke AppStream tumpukan yang paling dekat dengan lokasi mereka saat ini.
- Jika Anda menggunakan salah satu fitur opsional dalam WorkSpaces Aplikasi, seperti persistensi pengaturan aplikasi, folder rumah, atau armada elastis, Anda perlu mengonfigurasi replikasi Amazon S3 Cross-region untuk data pengguna untuk armada berbasis Windows dan replikasi lintas wilayah untuk armada berbasis Linux.
- Replikasi gambar di seluruh Wilayah. Untuk informasi selengkapnya, lihat [Menyalin gambar yang Anda miliki ke yang lain Wilayah AWS di WorkSpaces Aplikasi Amazon](#) dalam AWS dokumentasi.
- Untuk armada yang bergabung dengan domain, pastikan infrastruktur Direktori Aktif, termasuk Layanan Federasi Direktori Aktif (AD FS) (kecuali jika Anda menggunakan SAMP 2.0 dan Amazon Cognito sebagai alternatif), dikonfigurasi dengan benar di Wilayah lain, dan yang Anda gunakan untuk kemampuan replikasi Multi-wilayah. [AWS Directory Service for Microsoft Active Directory](#)
- Arahkan pengguna ke titik akhir Aplikasi latensi terendah WorkSpaces . Untuk informasi selengkapnya, lihat posting AWS blog [Optimalkan pengalaman pengguna dengan perutean berbasis latensi untuk Aplikasi Amazon](#). WorkSpaces

Gunakan arsitektur tanpa server

Arsitektur tanpa server menghilangkan overhead manajemen server dan mengurangi biaya dengan menggunakan layanan yang dikelola cloud untuk fungsi komputasi.

Gunakan layanan AWS tanpa server seperti berikut ini:

- [AWS Lambda](#) untuk mengotomatiskan tugas dan mengintegrasikan logika khusus melalui fungsi yang digerakkan oleh peristiwa
- [Amazon S3](#) menyediakan penyimpanan yang dapat diskalakan untuk data pengguna WorkSpaces Aplikasi, file aplikasi, dan artefak sesi

- [Amazon CloudWatch](#) menyediakan pemantauan, pencatatan, dan peringatan untuk metrik kinerja dan penggunaan WorkSpaces Aplikasi
- [Amazon Cognito](#) untuk memfasilitasi otentikasi pengguna dan kontrol akses untuk aplikasi Aplikasi WorkSpaces
- [Amazon API Gateway](#) RESTful APIs untuk membuat antarmuka antara WorkSpaces Aplikasi dan layanan lain atau aplikasi khusus

Eksperimen lebih sering

Infrastruktur cloud memungkinkan pengujian cepat berbagai konfigurasi sumber daya untuk mengoptimalkan kinerja dan biaya.

- Uji berbagai jenis instans untuk mengoptimalkan kinerja dan biaya:
 - Bandingkan kinerja streaming di berbagai kelompok instans.
 - Evaluasi instans GPU vs non-GPU untuk aplikasi grafis.
 - Uji instans yang dioptimalkan memori untuk aplikasi intensif memori.
- Menguji konfigurasi aplikasi dengan menggunakan Image Builder:
 - Buat gambar uji dengan konfigurasi aplikasi yang berbeda.
 - Validasi kinerja aplikasi sebelum penerapan.
 - Uji kompatibilitas aplikasi dengan jenis instans yang berbeda.
- Uji pengaturan armada dengan menggunakan konfigurasi kapasitas armada seperti kapasitas minimum dan maksimum, kebijakan penskalaan, pengaturan sesi seperti durasi sesi maksimum, dan pengaturan batas waktu pemutusan sambungan.

Pertimbangkan simpati mekanis

Pilih layanan cloud berdasarkan persyaratan dan pola penggunaan khusus beban kerja Anda untuk memastikan kinerja dan efisiensi yang optimal.

- Pilih instans Graphics G5 untuk aplikasi grafis intensif, aplikasi yang memerlukan DirectX, OpenGL, OpenCL, atau perangkat lunak visualisasi 3D.
- Pilih `stream.standard` instans untuk aplikasi bisnis, browser web, dan aplikasi grafis ringan
- Pantau dan sesuaikan protokol streaming berdasarkan CloudWatch metrik seperti `StreamingSessionLatency`.

- Konfigurasi WorkSpaces Aplikasi VPCs yang paling dekat dengan pengguna Anda, dan gunakan bandwidth jaringan yang sesuai berdasarkan kebutuhan aplikasi Anda.
- Pilih jenis armada yang sesuai berdasarkan perilaku aplikasi. Misalnya, pilih armada sesi tunggal untuk aplikasi yang membutuhkan sumber daya khusus dan armada multi-sesi untuk aplikasi yang dapat berbagi sumber daya secara efisien.
- Pertimbangkan kompatibilitas aplikasi dengan lingkungan multi-sesi.
- Gunakan [fitur pengalihan sistem file](#) untuk menangani interaksi antara aplikasi jarak jauh dan lokal. Untuk informasi selengkapnya, lihat posting AWS blog [Meluncurkan aplikasi lokal dari sesi streaming WorkSpaces Aplikasi Amazon](#).

Pilar optimasi biaya

[Pilar optimasi biaya](#) dari AWS Well-Architected Framework berfokus pada memaksimalkan nilai bisnis sambil meminimalkan pengeluaran. Ini membantu memastikan bahwa setiap dolar yang Anda belanjakan untuk sumber daya cloud berkontribusi secara efektif untuk mencapai tujuan organisasi Anda.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Manajemen kapasitas armada dan pemilihan tipe instans
- Optimalisasi penskalaan dan penjadwalan
- Pemantauan dan analisis pola penggunaan
- Alokasi dan pelacakan biaya

Menerapkan manajemen keuangan cloud

Membangun kemampuan organisasi khusus dalam manajemen keuangan cloud dan optimalisasi biaya melalui program dan proses terstruktur untuk memaksimalkan nilai dan efisiensi cloud.

- Pantau biaya WorkSpaces Aplikasi dengan menggunakan [AWS Cost Explorer](#) dan laporan penggunaan untuk melacak penggunaan jam streaming, menganalisis biaya instans armada, dan memantau distribusi biaya regional.
- Rencanakan dan tetapkan kontrol biaya dengan menggunakan [AWS Budgets](#) untuk menetapkan peringatan untuk keseluruhan biaya layanan WorkSpaces Aplikasi, membuat ambang anggaran untuk layanan, dan memantau pengeluaran aktual terhadap jumlah yang dianggarkan. Untuk informasi selengkapnya, lihat posting AWS blog [Cara menggunakan otomatisasi untuk mengoptimalkan dan mengontrol biaya WorkSpaces Aplikasi Amazon](#).

Tambahkan model konsumsi

Skala sumber daya komputasi dan biaya berdasarkan pola penggunaan aktual. Misalnya, Anda dapat mematikan lingkungan non-produksi selama jam kerja untuk mengoptimalkan pengeluaran.

- Pilih model harga yang sesuai. Misalnya, gunakan armada yang selalu aktif untuk penggunaan yang konsisten dan armada sesuai permintaan untuk beban kerja variabel.

- Pilih jenis instans yang optimal. Misalnya, gunakan `stream.standard` instance untuk aplikasi umum dan gunakan instance grafis (G4dn) hanya bila diperlukan.

Ukur efisiensi keseluruhan

Menghitung dan melacak output bisnis untuk mengukur peningkatan efisiensi dan memandu upaya optimasi. `cost-per-unit`

- Lacak efisiensi sesi.
- Pantau pemanfaatan armada dengan menggunakan CloudWatch metrik berikut:
 - `AvailableCapacity` untuk melacak kapasitas yang tidak terpakai
 - `InUseCapacity` untuk mengukur penggunaan aktual
- Hitung dan lacak biaya per sesi seperti biaya per jam streaming, biaya per pengguna, dan biaya per aplikasi.
- Terapkan [Pengoptimal Biaya untuk WorkSpaces Aplikasi untuk](#) memantau pembangun Anda.
- Bandingkan biaya antar jenis armada. Misalnya, bandingkan:
 - Biaya lisensi untuk sesi tunggal dan multi-sesi
 - Tingkat pemanfaatan sumber daya
 - Kepadatan pengguna per instance
- Gunakan data pelacakan proses untuk mengidentifikasi aplikasi yang kurang dimanfaatkan atau tidak perlu. Untuk informasi selengkapnya, lihat posting AWS blog [Melacak proses pengguna di sesi WorkSpaces Aplikasi Amazon](#).

Berhenti menghabiskan uang untuk angkat berat yang tidak berdiferensiasi

AWS mengelola operasi infrastruktur dan menawarkan layanan terkelola sehingga organisasi Anda dapat fokus pada tujuan bisnis alih-alih pemeliharaan TI.

- Membuat dan memelihara gambar aplikasi dengan menggunakan Image Builder untuk mengemas aplikasi Anda, mengkonfigurasi pengaturan aplikasi, dan menguji kompatibilitas aplikasi.
- Konfigurasi spesifikasi armada dengan memilih jenis instans yang sesuai dan menentukan ambang batas penskalaan serta menetapkan batas kapasitas yang diinginkan.

- Siapkan opsi penyimpanan persisten dengan mengonfigurasi folder beranda di [Amazon S3](#) untuk armada berbasis Windows dan sistem file bersama di [Amazon EFS](#) untuk armada berbasis Linux. Tetapkan izin penyimpanan dan tentukan kebijakan penyimpanan.

Analisis dan atribut pengeluaran

Cloud memungkinkan pelacakan penggunaan sumber daya dan biaya per beban kerja secara akurat, yang memungkinkan pengukuran laba atas investasi (ROI) yang akurat dan peluang pengoptimalan yang ditargetkan.

- Menerapkan strategi penandaan komprehensif untuk armada untuk alokasi biaya, gambar untuk pelacakan aset, pembuat gambar untuk penunjukan lingkungan, dan tumpukan untuk pengelompokan organisasi.
- Gunakan [Laporan AWS Biaya dan Penggunaan \(AWS CUR\)](#) untuk memecah biaya WorkSpaces Aplikasi dengan sumber daya yang ditandai, dan menganalisis biaya per armada, tumpukan, dan gambar.
- Gunakan [AWS Cost Explorer](#) untuk memvisualisasikan tren pengeluaran WorkSpaces Aplikasi, dan membandingkan biaya di berbagai dimensi seperti Wilayah dan jenis instans.
- Memantau dan menganalisis tingkat pemanfaatan armada, efisiensi jenis instans, dan jam streaming berdasarkan aplikasi.
- Lacak kapasitas cadangan yang tidak terpakai, armada atau tumpukan yang kurang dimanfaatkan, dan periode idle dalam penggunaan armada.
- Hitung dan lacak biaya per pengguna untuk setiap aplikasi, jam streaming per aplikasi, dan tarif adopsi pengguna untuk aplikasi streaming.
- Siapkan analisis penggunaan terperinci dengan mengonfigurasi laporan penggunaan WorkSpaces Aplikasi, menggunakan [Amazon Athena](#) untuk menanyakan data penggunaan, dan membuat visualisasi [di Amazon Quick](#) untuk wawasan biaya dan penggunaan.
- Mengevaluasi pertimbangan biaya total seperti lisensi Windows Server, model lisensi aplikasi, dan lisensi per pengguna dibandingkan dengan lisensi per perangkat.
- Gunakan Amazon Athena untuk menanyakan dan menganalisis biaya penyimpanan folder rumah dan pola penggunaan oleh pengguna. Untuk informasi selengkapnya, lihat posting AWS blog [Cara melaporkan penggunaan folder beranda WorkSpaces Aplikasi Amazon dengan Amazon Athena](#).

Pilar keberlanjutan

[Pilar keberlanjutan](#) dari AWS Well-Architected Framework menekankan meminimalkan jejak lingkungan Anda dan mengoptimalkan penggunaan dan efisiensi energi. Ini memandu arsitek untuk membuat keputusan sadar lingkungan dalam desain sistem dan strategi alokasi sumber daya mereka.

Area fokus utama untuk menerapkan pilar ini ke lingkungan streaming WorkSpaces Aplikasi Anda:

- Memahami dan mengoptimalkan alokasi sumber daya agar sesuai dengan permintaan aktual dan meminimalkan pemborosan di lingkungan streaming
- Menganalisis dan beradaptasi dengan pola konsumsi pengguna untuk meningkatkan efisiensi pengiriman aplikasi dan sesi streaming
- Memilih dan menggunakan konfigurasi perangkat keras yang sesuai untuk memaksimalkan efisiensi energi sambil memenuhi persyaratan kinerja
- Menggunakan kemampuan layanan AWS terkelola untuk mendapatkan keuntungan dari skala ekonomi dan fitur efisiensi bawaan yang ditawarkan oleh layanan ini

Pahami dampak Anda

Pantau dan optimalkan dampak lingkungan beban kerja Anda dengan mengukur efisiensi sumber daya dan emisi per unit output. Gunakan data ini untuk membangun KPIs dan memandu peningkatan keberlanjutan.

- Pantau pola pemanfaatan armada.
- Lacak jam streaming per pengguna.
- Menganalisis tren penggunaan kapasitas armada.

Menetapkan tujuan keberlanjutan

Tetapkan tujuan keberlanjutan yang terukur untuk setiap beban kerja yang selaras dengan tujuan organisasi. Fokus pada pengurangan intensitas sumber daya per transaksi saat Anda menskalakan.

- Tetapkan target untuk tingkat pemanfaatan armada, efisiensi jenis instans, dan pengoptimalan jam streaming.

- Rencanakan kapasitas berdasarkan pola penggunaan aktual.

Maksimalkan pemanfaatan

Optimalkan efisiensi beban kerja dengan mengukur sumber daya yang tepat dan memaksimalkan pemanfaatan. Mengurangi kapasitas idle untuk meminimalkan konsumsi energi dan meningkatkan keberlanjutan.

- Konfigurasi penskalaan otomatis agar sesuai dengan permintaan aktual.
- Kapasitas armada ukuran yang tepat berdasarkan pola penggunaan.
- Menerapkan batas kapasitas minimum dan maksimum yang sesuai.
- Pilih jenis instans yang sesuai untuk beban kerja.
- Pantau dan optimalkan kepadatan sesi streaming.
- Kurangi kapasitas idle selama jam-jam di luar sibuk.

Mengantisipasi dan mengadopsi penawaran perangkat keras dan perangkat lunak baru yang lebih efisien

Tetap mendapat informasi tentang, dan segera adopsi, teknologi efisien baru dari mitra dan pemasok untuk terus meningkatkan dampak lingkungan beban kerja Anda.

- Gunakan jenis instance generasi saat ini.
- Tingkatkan ke jenis instans yang lebih baru bila tersedia.
- Optimalkan pengaturan streaming aplikasi.
- Konfigurasi protokol streaming yang sesuai.
- Perbarui ke fitur WorkSpaces Aplikasi terbaru.

Layanan terkelola yang digunakan

Manfaatkan layanan cloud bersama dan solusi terkelola untuk memaksimalkan efisiensi pemanfaatan sumber daya sekaligus meminimalkan dampak lingkungan melalui penskalaan otomatis dan manajemen siklus hidup.

- Gunakan [Amazon S3](#) untuk penyimpanan pengguna untuk armada berbasis Windows dan Amazon [EFS](#) untuk sistem file bersama untuk armada berbasis Linux.
- Implementasikan [CloudWatch](#) untuk pemantauan.
- Konfigurasi [IAM](#) untuk manajemen akses.

Mengurangi dampak hilir beban kerja cloud Anda

Merancang layanan untuk meminimalkan kebutuhan sumber daya sisi klien, untuk mengurangi konsumsi energi dan memperpanjang masa pakai perangkat bagi pengguna.

- Sesuaikan durasi sesi maksimum untuk mencegah konsumsi sumber daya yang tidak perlu.
- Konfigurasi batas waktu sesi yang sesuai.
- Tetapkan kebijakan batas waktu pemutusan sambungan.
- Menerapkan kebijakan persistensi sesi jika diperlukan.

Sumber Daya

AWS dokumentasi

- [AWS Kerangka Well-Architected](#)
- [Panduan Administrasi WorkSpaces Aplikasi Amazon](#)
- [Panduan CloudWatch Pengguna Amazon](#)
- [Panduan Pengguna Amazon EFSS](#)
- [Panduan Pengguna Amazon S3](#)
- [Panduan Pengguna IAM](#)

AWS posting blog

- [Penargetan WorkSpaces Aplikasi Berbasis Keanggotaan Grup Direktori Aktif](#)
- [Buat Penyedia Identitas Tunggal untuk semua Tumpukan WorkSpaces Aplikasi Amazon Anda dengan Azure AD](#)
- [Mengkonfigurasi Bantuan Jarak Jauh Windows untuk Aplikasi Amazon WorkSpaces dan Amazon WorkSpaces](#)
- [Membuat data AS2 TrustedDomains DNS TXT untuk mengarahkan klien asli Aplikasi WorkSpaces Amazon ke penyedia identitas pihak ketiga](#)
- [Membuat pencatatan khusus dan CloudWatch peringatan Amazon di Aplikasi Amazon WorkSpaces](#)
- [Pengalihan Lintas Wilayah dengan Geo Targetly dan Aplikasi Amazon WorkSpaces](#)
- [Sumber daya lintas akun dan Aplikasi Amazon WorkSpaces](#)
- [Aktifkan federasi dengan Aplikasi Bio-key dan PortalGuard Amazon WorkSpaces](#)
- [Mengaktifkan Federasi dengan Aplikasi Sederhana SAMLphp dan Amazon WorkSpaces](#)
- [Mengaktifkan federasi identitas dengan Duo Single Sign-On dan Aplikasi Amazon WorkSpaces](#)
- [Mengaktifkan Federasi Identitas dengan Aplikasi Shibboleth dan Amazon WorkSpaces](#)
- [Strategi failover untuk VDI lokal dengan Komputasi Pengguna Akhir Amazon](#)
- [Bagaimana Amazon Menggunakan WorkSpaces Aplikasi Amazon untuk Memberikan Ilmuwan Data dan Analisis Akses ke Data Sensitif](#)

- [Cara mengonfigurasi otentikasi berbasis sertifikat untuk Aplikasi Amazon WorkSpaces](#)
- [Cara menggunakan klaim Okta dengan hak aplikasi untuk Aplikasi Amazon WorkSpaces](#)
- [Mengelola Lab Komputer di WorkSpaces Aplikasi Amazon dengan Manajemen Aplikasi Virtual Sumber Terbuka](#)
- [Metode mengalokasikan biaya WorkSpaces Aplikasi Anda ke unit bisnis Anda](#)
- [Memantau WorkSpaces Aplikasi Amazon dengan OpenSearch Layanan Amazon dan Amazon Kinesis Data Firehose](#)
- [Pemisahan Jaringan dan Sanitasi Data menggunakan Amazon, WorkSpaces Aplikasi WorkSpaces Amazon, dan Amazon Macie](#)
- [OneLogin SSO dengan Aplikasi Amazon WorkSpaces](#)
- [Optimalkan jalur audio panggilan Amazon Connect Anda dengan WorkSpaces Aplikasi Amazon](#)
- [Penyimpanan persisten untuk WorkSpaces Aplikasi Amazon Armada Linux di Amazon Elastic File System](#)
- [Arahkan ulang aplikasi Okta SAMP ke klien asli Aplikasi Amazon WorkSpaces](#)
- [Sederhanakan manajemen gambar WorkSpaces Aplikasi Amazon dengan Masking Aplikasi](#)
- [Streaming aplikasi dengan biaya lebih rendah dengan armada Amazon WorkSpaces Applications Elastic dan kompatibilitas Linux](#)
- [Streaming dari titik akhir VPC antarmuka untuk lingkungan yang Diatur dengan Aplikasi WorkSpaces](#)
- [Menggunakan hak WorkSpaces aplikasi Aplikasi Amazon dengan Azure AD](#)
- [Reporter Masalah Pengguna untuk Aplikasi Amazon WorkSpaces](#)
- [Menggunakan hak WorkSpaces aplikasi Aplikasi Amazon dengan Google Workspace](#)
- [Menggunakan Auth0 dengan Microsoft Active Directory di Aplikasi Amazon WorkSpaces](#)
- [Menggunakan Microsoft AppLocker untuk mengelola pengalaman aplikasi di WorkSpaces Aplikasi Amazon](#)
- [Menggunakan Python untuk memberi daya pada GUI Asisten Pencitraan Linux WorkSpaces Aplikasi](#)
- [Opsi pengalihan aplikasi web untuk Klien WorkSpaces Aplikasi](#)

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	—	Juli 23, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat [kontrol akses berbasis atribut](#).

layanan abstrak

Lihat [layanan terkelola](#).

ASAM

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

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat [kecerdasan buatan](#).

AIOps

Lihat [operasi kecerdasan buatan](#).

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

Lokasi berbeda di dalam 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 CCoE](#) 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 CCoE, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

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

CMDB

Lihat [database manajemen konfigurasi](#).

repositori kode

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

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

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

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

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

CV

Lihat [visi komputer](#).

D

data saat istirahat

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

klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan 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 dapat diakses oleh pengguna akhir. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.

- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

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

ERP

Lihat [perencanaan sumber daya perusahaan](#).

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat [cabang](#).

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol [akses berbutir halus](#).

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui [pengambilan data perubahan](#) untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih

menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat [model pondasi](#).

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat [pembatasan geografis](#).

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur,

gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

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

pagar pembatas

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

IIoT

Lihat [Internet of Things industri](#).

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAC)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretabilitas

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

IoT

Lihat [Internet of Things](#).

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat [perpustakaan informasi TI](#).

ITSM

Lihat [manajemen layanan TI](#).

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat [kontrol akses berbasis label](#).

hak istimewa paling sedikit

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

angkat dan geser

Lihat [7 Rs](#).

sistem endian kecil

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

LLM

Lihat [model bahasa besar](#).

lingkungan yang lebih rendah

Lihat [lingkungan](#).

M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat [Program Percepatan Migrasi](#).

mekanisme

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

akun anggota

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

MES

Lihat [sistem eksekusi manufaktur](#).

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat [pembelajaran mesin](#).

modernisasi

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

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat [Mengevaluasi kesiapan modernisasi untuk](#) aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

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

MPA

Lihat [Penilaian Portofolio Migrasi](#).

MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat [kontrol akses asal](#).

OAI

Lihat [identitas akses asal](#).

OCM

Lihat [manajemen perubahan organisasi](#).

migrasi offline

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

OI

Lihat [integrasi operasi](#).

OLA

Lihat [perjanjian tingkat operasional](#).

migrasi online

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

OPC-UA

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

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

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

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat [tinjauan kesiapan operasional](#).

OT

Lihat [teknologi operasional](#).

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

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

buku pedoman

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

PLC

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

PLM

Lihat [manajemen siklus hidup produk](#).

kebijakan

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

ketekunan poliglot

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

penilaian portofolio

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

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona yang dihosting pribadi

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

kontrol proaktif

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

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat [lingkungan](#).

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

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

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat [Retrieval Augmented Generation](#).

ransomware

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

Matriks RASCI

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

RCAC

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

replika baca

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

arsitek ulang

Lihat [7 Rs](#).

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat [7 Rs](#).

Region

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan.

Untuk informasi selengkapnya, lihat [Menentukan Wilayah AWS akun yang dapat digunakan](#).

regresi

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

rehost

Lihat [7 Rs](#).

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat [7 Rs](#).

memplatform ulang

Lihat [7 Rs](#).

pembelian kembali

Lihat [7 Rs](#).

ketahanan

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

Untuk informasi lebih lanjut, lihat [AWS Cloud Ketahanan](#).

kebijakan berbasis sumber daya

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

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

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang

didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Tipe dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

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

melestarikan

Lihat [7 Rs](#).

pensiun

Lihat [7 Rs](#).

Retrieval Augmented Generation (RAG)

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

rotasi

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

SCADA

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

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensial 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 AWS sumber daya Anda](#).

variabel target

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

daftar tugas

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

lingkungan uji

Lihat [lingkungan](#).

pelatihan

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

gerbang transit

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

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

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau

memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat [lingkungan](#).

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

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

WQF

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

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bidikan zero-shot

Memberikan [LLM](#) dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembak) 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.