



Penemuan dan analisis portofolio untuk migrasi

AWS Bimbingan Preskriptif



AWS Bimbingan Preskriptif: Penemuan dan analisis portofolio untuk migrasi

Copyright © 2024 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 para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Pengantar	1
Hasil bisnis yang ditargetkan	1
Pedoman	3
Pengumpulan	3
Tentukan data apa yang perlu dikumpulkan	4
Tentukan cara mengumpulkan data	4
Pemilihan alat	6
Tahap 1	6
Tahap 2	9
Tahap 3	10
Fase 4	10
Prioritas	11
Tentukan kriteria apa yang akan digunakan	11
Tentukan cara menggunakan kriteria	12
Perencanaan	13
Buat grup migrasi	14
Ubah grup menjadi rencana	17
Sumber daya	18
Alat	18
Mitra	18
Pola SIMOLATIF	18
Pertanyaan yang Sering Diajukan	19
Bagaimana cara membangun kasus bisnis yang tepat?	19
Bagaimana cara menilai lingkungan saya secara akurat?	19
Bagaimana cara mengidentifikasi dan mengevaluasi mitra yang tepat untuk membantu saya?	19
Bagaimana cara memperkirakan biaya transisi besar seperti ini?	19
Berapa lama proses migrasi akan selesai?	20
Bagaimana cara menangani aplikasi warisan saya?	20
Bagaimana cara mempercepat upaya migrasi untuk mewujudkan manfaat bisnis dan teknologi dengan lebih cepat?	20
Ucapan Terima Kasih	21
Riwayat dokumen	22
Glosarium	23

#	23
A	24
B	27
C	28
D	31
E	35
F	37
G	38
H	39
I	40
L	43
M	44
O	47
P	49
Q	52
R	52
D	55
T	58
U	60
V	60
W	61
Z	62
.....	lxiii

Penemuan dan analisis portofolio untuk migrasi

Zach Hansen, Amazon Web Services (AWS)

Februari 2022([Riwayat dokumen](#))

Alur kerja penemuan dan analisis portofolio mendefinisikan pendekatan untuk menentukan, mengumpulkan, dan menganalisis data yang diperlukan untuk membuat rencana migrasi. Alur kerja ini membantu Anda memperkirakan tingkat upaya dan biaya yang terkait dengan migrasi portofolio aplikasi Anda, serta perkiraan tarif jangka tahunan untuk sumber daya komputasi dan penyimpanan di Amazon Web Services (AWS).

Aliran kerja ini juga mendefinisikan kumpulan data lengkap yang diperlukan untuk menganalisis, mengelompokkan, dan memprioritaskan migrasi aplikasi dengan benar, menyarankan alat pengumpulan data yang sesuai jika data Anda tidak lengkap atau tidak akurat, menentukan kriteria untuk memilih beban kerja migrasi percontohan, dan memberikan prioritas umum migrasi. Panduan ini memandu Anda melalui langkah-langkah tingkat tinggi untuk mencapai tujuan ini.

Hasil bisnis yang ditargetkan

Panduan ini membahas AWS model preskriptif untuk penemuan portofolio dan analisis. Meskipun metode lain tersedia, mereka dapat membahayakan kelengkapan atau kedalaman hasil.

Ada tiga hasil yang diharapkan dari penemuan dan analisis portofolio:

- Pengetahuan rinci tentang lingkungan migrasi
- Pemahaman tentang saling ketergantungan portofolio
- Rencana gelombang migrasi yang akan berisi detail tentang beban kerja mana yang akan dimigrasi dan kapan

Untuk membuat keputusan tentang beban kerja yang Anda migrasi, Anda harus memahami sepenuhnya lingkungan migrasi. Anda harus terbiasa dengan setiap beban kerja individu, serta bagaimana setiap beban kerja bergantung pada beban kerja lain untuk mendukung bisnis Anda.

Mengembangkan rencana yang dipikirkan dengan baik yang membutuhkan revisi minimal membantu mendukung tujuan bisnis Anda dengan mengurangi risiko seperti downtime yang tidak direncanakan dan mengurangi biaya migrasi. Meskipun Anda akan mengetahui lebih lanjut tentang beban kerja

selama penemuan portofolio, dan perlu memperbarui rencana yang sesuai, panduan ini akan membantu Anda meminimalkan frekuensi dan tingkat revisi tersebut.

Pedoman

Anda mungkin menemukan diri Anda secara bersamaan menjalankan berbagai bagian dari alur kerja ini dengan beban kerja yang berbeda. Misalnya, Anda tidak perlu menyelesaikan pengumpulan semua data untuk semua beban kerja sebelum mempertimbangkan cara memprioritaskan beban kerja. Bahkan, Anda mungkin akan melakukan keduanya pada saat bersamaan. Kuncinya adalah terus-menerus bertanya “bisakah saya memajukan ini?” dan melakukannya jika jawaban Anda adalah “ya!”

Tiga bagian utama dari alur kerja ini adalah pengumpulan data, prioritas, dan perencanaan.

- Pengumpulan data difokuskan pada pemahaman data apa yang perlu dikumpulkan dan bagaimana data itu akan dikumpulkan.
- Prioritas difokuskan pada penggunaan data yang telah Anda kumpulkan tentang beban kerja migrasi untuk menyelaraskan rencana migrasi ke driver bisnis dan teknis Anda untuk migrasi.
- Perencanaan difokuskan pada mengambil prioritas beban kerja dan menggabungkannya dengan analisis ketergantungan.

Pengumpulan

Mulailah proyek dengan menemukan lingkungan Anda. Tingkat detail dalam data yang Anda kumpulkan tergantung pada kebutuhan bisnis Anda. Jika Anda perlu mendukung kasus bisnis atau memberikan perkiraan keuangan untuk tujuan lain, mulailah dengan mengumpulkan data yang diperlukan untuk membuat perkiraan tarif tahunan dan perkiraan biaya migrasi. Jika Anda tidak memerlukan perkiraan keuangan, fokuslah pada data aplikasi dan infrastruktur yang diperlukan. Data ini melengkapi data inti yang diperlukan untuk membuat perkiraan biaya dan akan digunakan dalam tahap analisis.

Memahami dependensi antara aplikasi dan infrastruktur (yaitu, aplikasi ke aplikasi, aplikasi ke infrastruktur) sangat penting untuk menentukan dampak beban kerja bergerak. Jumlah data yang diperlukan juga dapat bervariasi berdasarkan faktor-faktor seperti dampak bisnis atau cakupan dampak jika aplikasi tidak tersedia, karena migrasi biasanya memerlukan jendela pemadaman yang direncanakan. Sangat jarang mengumpulkan semua data, jadi gunakan penilaian yang baik untuk memutuskan kapan Anda memiliki cukup data untuk melanjutkan ke tahap berikutnya.

Tentukan data apa yang perlu dikumpulkan

Anda mungkin tidak memerlukan data atau kumpulan data lengkap, tergantung pada kasus penggunaan untuk data. Misalnya, jika Anda menjelajahi biaya migrasi, Anda memerlukan tidak lebih dari pemahaman tingkat tinggi tentang lingkungan lokal Anda (misalnya, jumlah server dengan rincian oleh sistem operasi). Perkiraan yang dihasilkan akan berkorelasi erat dengan keakuratan input data, jadi ingatlah persyaratan keluaran Anda.

Jika Anda perlu memutuskan bagaimana dan kapan aplikasi akan bermigrasi, Anda memerlukan kumpulan data yang lengkap dan akurat yang mencakup dokumentasi terperinci tentang aplikasi dan infrastruktur yang akan dimigrasi. Kami sangat merekomendasikan agar Anda menggunakan alat penemuan aplikasi/infrastruktur otomatis untuk memastikan kelengkapan dan keakuratan. Lihat [Sumber daya](#) untuk daftar ini.

Pertimbangkan apa tujuannya dan apa driver Anda, dan kemudian tentukan data yang perlu Anda kumpulkan. Pertimbangan utama untuk memutuskan apakah akan mengumpulkan data tertentu adalah berapa banyak waktu yang diperlukan untuk mengumpulkan.

Tentukan cara mengumpulkan data

Ketika Anda telah menentukan data apa yang harus dikumpulkan, tentukan cara mendapatkannya. Ingatlah untuk mempertimbangkan berapa lama dan berapa banyak usaha yang diperlukan untuk mendapatkan data.

Keputusan utama Anda pada saat ini adalah apakah akan menginstal alat pengumpulan data untuk membantu Anda mengumpulkan data dengan cepat. Kecuali jika Anda memiliki alasan kuat untuk tidak menggunakannya, kami sarankan Anda menggunakan alat penemuan karena alat ini dapat mempercepat penemuan secara signifikan. Berikut adalah beberapa pertanyaan untuk ditanyakan yang akan membantu membenarkan keputusan Anda:

- Apakah ahli materi pelajaran saya tahu jawaban atas pertanyaan?
- Apakah saya memiliki beban kerja lama di mana orang yang tahu tentang beban kerja tersebut tidak lagi bersama organisasi?
- Apa yang akan dikumpulkan alat penemuan? Apakah ini selaras dengan data yang telah saya putuskan untuk dikumpulkan?
- Data apa yang perlu saya kumpulkan secara manual?
- Berapa lama saya harus mengharapkan pengumpulan data untuk mengambil? Berapa lama aku punya?

- Apa proses peninjauan keamanan untuk menginstal alat? Bisakah kita menginstal agen untuk menemukan beban kerja?
- Berapa lama pengadaan? Dapatkah saya pintas ini dengan menggunakan alat gratis atau AWS Penawaran Marketplace?
- Seberapa akurat dataset saya? Dapatkah saya mengambil kata seseorang untuk itu atau haruskah saya mengumpulkan data empiris yang lebih akurat dan tepat?

Pertanyaan terakhir adalah keputusan kunci yang harus dibuat oleh tim kepemimpinan: Apa toleransi risiko untuk membuat keputusan yang salah? Keputusan yang salah terjadi ketika Anda memiliki data yang tidak lengkap atau tidak akurat.

Ketika Anda telah memutuskan apakah akan menggunakan alat penemuan, Anda harus menentukan proses untuk menyusun sumber data Anda. Alat penemuan bermanfaat tetapi tidak dapat memberikan semua yang Anda butuhkan. Pahami apa yang akan dan tidak akan disediakan oleh alat ini. Biasanya dibutuhkan dua hingga empat minggu untuk mendapatkan data yang baik. Saat Anda menunggu alat penemuan mengumpulkan data, kumpulkan informasi tambahan yang akan Anda gunakan di fase migrasi future. Berikut ini beberapa contoh data untuk dikumpulkan di luar alat penemuan:

- Siapa yang memiliki atau mendukung aplikasi?
- Unit bisnis apa yang didukung aplikasi ini?
- Apa kepentingan relatif (kekritisian atau tingkat) dari aplikasi untuk bisnis?

Warning

Jika Anda tidak berhati-hati, penemuan bisa menjadi pengurasan sumber daya yang tak terbatas. Pada titik tertentu, Anda harus memutuskan untuk bergerak maju dengan data yang tidak lengkap. Hampir tidak mungkin untuk mendapatkan 100 persen data yang akurat dan lengkap selama penemuan. Tujuan untuk fase ini bukanlah akurasi lengkap tetapi, lebih tepatnya, data yang cukup baik yang dapat membawa Anda ke fase berikutnya, meminimalkan churn yang akan Anda alami selama fase mendatang. Setelah investasi penemuan minimum satu hingga dua bulan, jumlah data baru yang Anda temukan berkurang dengan cepat.

Memilih dan menerapkan alat penemuan

Bagian ini ditujukan untuk para pemimpin teknis dan arsitek yang bertanggung jawab untuk merencanakan proyek migrasi dalam skala apa pun.

Saat merencanakan migrasi ke cloud, sangat penting untuk memiliki pandangan menyeluruh tentang lingkungan Anda, mulai dari komunikasi logis hingga kapasitas perangkat keras. Rincian ini mungkin tampak mendasar, tetapi mereka membuat perbedaan besar ketika Anda menentukan bagaimana dan di mana untuk bermigrasi lingkungan Anda.

Alat penemuan dirancang untuk memberi Anda informasi tentang lingkungan Anda, seperti berikut ini:

- Status siklus aktif
- Pemanfaatan kapasitas
- Dependensi aplikasi
- Standar teknis
- Informasi umum tentang setiap aset di lingkungan Anda

Selain memetakan dependensi, penting untuk menemukan pola di lingkungan Anda. Dengan menemukan pola-pola tersebut, Anda dapat melihat cara menggunakan kembali pendekatan migrasi. Misalnya, Anda dapat menggunakan kembali pendekatan migrasi tunggal untuk beberapa aplikasi yang memiliki versi serupa, perangkat keras, komunikasi, dan komponen lainnya.

Untuk menemukan pola-pola tersebut, Anda memerlukan informasi yang didokumentasikan tentang infrastruktur Anda. Alat penemuan dapat membantu Anda menemukan dan mendokumentasikan informasi tersebut.

Fase 1: Penilaian awal

Pada tahap penilaian awal, pastikan untuk melibatkan pemangku kepentingan yang tepat dalam diskusi. Anda ingin menilai jenis alat apa yang dapat memberikan wawasan berharga untuk mendukung perjalanan migrasi Anda ke cloud.

Sebagai bagian dari fase ini, jawab pertanyaan-pertanyaan di tabel berikut selengkap yang Anda bisa. Semakin banyak celah data yang Anda temukan di awal, semakin dekat Anda dengan memilih alat yang memberikan wawasan tentang kebutuhan Anda.

Pertanyaan	Contoh
<p>Apakah Anda memiliki alat database manajemen konfigurasi (CMDB) hari ini? Apakah itu dapat diandalkan? Bagaimana ini bisa membantu perjalanan Anda?</p>	<ul style="list-style-type: none"> • Kami memiliki alat CMDB Y diinstal. • 80% dari baseline adalah yang terbaru. • Nama host, nama aplikasi, kontak dukungan, alamat IP, dan versi sistem operasi (OS) (versi OS tidak up to date).
<p>Apakah Anda memiliki alat pemantauan dan kinerja? Apakah itu dapat diandalkan? Apakah itu mendukung semua aset Anda? Apa yang dimilikinya dan apa yang tidak dimilikinya?</p>	<ul style="list-style-type: none"> • Kami menggunakan aplikasi buatan rumah yang mengumpulkan kinerja server (CPU, memori, dan penggunaan disk). • Ini tidak berfungsi untuk SUSE Linux Enterprise Server (SLES) 11.4, atau untuk Windows 2003 dan 2008. • Kami tidak berpikir datanya akurat, dan kami tidak tahu server mana yang berkomunikasi satu sama lain.
<p>Apa yang ingin Anda ketahui tentang lingkungan Anda yang tidak Anda ketahui hari ini?</p>	<ul style="list-style-type: none"> • Pemetaan aplikasi. Kami tidak memiliki data paket yang diinstal di server dan ingin tahu apakah IBM WebSphere, Java, .NET, atau versi middleware lainnya diinstal pada server. • Server dikelompokkan secara manual. Terkadang kita tidak tahu apakah pengelompokannya benar atau mutakhir. Kami ingin tahu apa saja server, alamat IP, dan port yang mereka ajak bicara untuk lalu lintas masuk dan keluar.
<p>Apa tujuan Anda dengan alat penemuan? Mengapa Anda pikir Anda membutuhkannya?</p>	<ul style="list-style-type: none"> • Kami ingin bantuan dengan perencanaan migrasi kami. • Kami memiliki aplikasi lama yang kekurangan dokumentasi. • Kita perlu mengetahui kinerja semua server untuk dapat mengukur lingkungan target.

Pertanyaan	Contoh
Dapatkah salah satu alat yang ada memberi Anda informasi yang Anda inginkan?	<ul style="list-style-type: none">• CMDB dan aplikasi buatan rumah kami dapat membantu metrik CPU (penggunaan rata-rata dan puncak), metrik memori (penggunaan rata-rata dan puncak), penggunaan disk total, alat yang diinstal pada setiap server, versi aplikasi, dan informasi kontak tim.
Berapa ukuran dasar di mana Anda berniat untuk menjalankan alat penemuan?	<ul style="list-style-type: none">• 500 server Windows (400 virtual dan 100 fisik)• 1.200 server Linux (800 virtual dan 400 fisik)• 200 kontainer
Sistem operasi dan versi apa yang berjalan di lingkungan?	<ul style="list-style-type: none">• Windows 2003• IBM AIX 7.2 dan 5.3• Red Hat Enterprise Linux (RHEL) 7.1• SLES 11.4 dan 15.2
Apa hypervisor dan versi yang berjalan di lingkungan?	<ul style="list-style-type: none">• IBM Power8 dan Power7• VMware
Orkestrasi kontainer apa yang berjalan di lingkungan?	<ul style="list-style-type: none">• Kubernetes• Red OpenShift
Apakah Anda memiliki anggaran yang dialokasikan untuk kegiatan ini? Perhatikan jumlahnya jika Anda tahu.	<ul style="list-style-type: none">• Ya, kami memiliki \$ X untuk digunakan hingga tahun depan atau hingga akhir tahun depan.

Pertanyaan	Contoh
Bagaimana Anda akan menggunakan output alat penemuan untuk membantu perjalanan migrasi Anda?	<ul style="list-style-type: none">• Kinerja: Kami berharap untuk menggunakan informasi kinerja untuk mengukur aset di lingkungan target.• TCO: Dengan input dari kinerja, kami ingin dapat menghitung total biaya kepemilikan (TCO).• Aplikasi: Kami berharap untuk mengetahui paket dan versi apa yang berjalan di server.• Jaringan : Kami berharap dapat melihat dengan siapa server berbicara (masuk dan keluar) dan merencanakan migrasi yang lancar ke cloud.

Fase 2: Penilaian

Pada tahap kedua, penilaian alat, Anda sudah memahami apa yang Anda cari dan mengapa, sehingga Anda dapat membuat keputusan yang lebih baik selama pemilihan alat. Untuk mengevaluasi opsi, Anda dapat membandingkan solusi yang ada dengan alat yang tersedia AWS dan AWS Partner

Penting untuk mempertimbangkan aspek-aspek berikut, yang dapat membuat perbedaan selama fase berikutnya, perencanaan peluncuran:

- Lisensi
- SaaS atau dikerahkan pelanggan
- Tanpa agen, berbasis agen, atau berbasis login
- Sistem operasi yang didukung

Untuk memastikan bahwa alat akan memberikan hasil yang Anda harapkan, kami sarankan meminta penyedia perangkat lunak untuk demo alat tersebut.

Fase 3: Perencanaan peluncuran

Tahap perencanaan peluncuran sering diremehkan, karena demo evaluasi dan pengujian pada server lab dapat tampak sederhana dibandingkan dengan migrasi penuh.

Saat meluncurkan alat penemuan, penting untuk memiliki proses yang terdefinisi dengan baik dan mengetahui tim mana yang perlu dilibatkan. Sangat penting untuk memiliki lingkungan dipetakan dan diatur sedini mungkin sehingga Anda dapat menjalankan alat cukup lama untuk mendapatkan wawasan tentang lingkungan Anda. Anda kemudian dapat menggunakan wawasan tersebut saat merencanakan migrasi.

Strategi peluncuran Anda mungkin bergantung pada pilihan perkakas Anda. Anda dapat menggunakan solusi seperti shell scripting, PowerShell, AWS Systems Manager, Ansible, Chef, atau alat konfigurasi perangkat lainnya. Sebaiknya luncurkan terlebih dahulu di semua server non-produksi Anda. Ketika Anda merasa yakin bahwa peluncuran tidak memengaruhi sistem apa pun, gunakan alat ini di lingkungan produksi.

Buat dokumentasi yang jelas dan lengkap yang menjelaskan hal berikut:

- Prasyarat
- Instalasi
- Instalasi ulang
- Menghapus instalasi
- Berkas log
- Proses validasi
- Masalah diketahui
- Titik kontak

Sertakan informasi teknis lainnya tentang alat penemuan yang telah Anda pilih.

Fase 4: Menganalisis

Setelah alat penemuan berjalan dan melaporkan sesuai rancangan, pertanyaan umum adalah, “Berapa lama saya harus menyimpan pelaporan aset saya untuk memberikan hasil yang tepat?” Jawaban atas pertanyaan itu dipengaruhi oleh faktor-faktor berikut:

- Pemakaian puncak: Jika sistem Anda memiliki penggunaan puncak hanya di awal bulan, Anda mungkin perlu menunggu hingga 30 hari untuk mendapatkan metrik penggunaan puncak.

- Pemetaan jaringan: Jika lamaran Anda memiliki pekerjaan yang dijadwalkan untuk berjalan satu kali dalam sebulan, Anda mungkin perlu menunggu hingga 30 hari.
- Kegiatan terjadwal: Jika Anda memiliki aktivitas terjadwal tertentu yang berjalan satu kali setiap kuartal, Anda mungkin perlu menunggu hingga 3 bulan. Perlakukan ini sebagai pengecualian. Anda tidak perlu menunggu 3 bulan untuk semua aplikasi Anda.

Biasanya, perusahaan mengumpulkan setidaknya 2 minggu data untuk merencanakan migrasi. Untuk data yang lebih baik, kami sarankan untuk mengumpulkan data setidaknya selama 4 minggu. Rentang itu berfungsi untuk sebagian besar kasus dan perilaku aplikasi. Namun, pastikan untuk merencanakan jumlah minggu berdasarkan kebutuhan aplikasi Anda.

Prioritas

Kunci untuk memahami dan menganalisis portofolio adalah menyelaraskan kumpulan data Anda dengan driver bisnis dan teknis organisasi Anda. Sejajarkan driver ini ke elemen data yang Anda kumpulkan dan gunakan untuk menentukan peringkat setiap aplikasi yang Anda rencanakan untuk bermigrasi. Pengemudi umum termasuk peningkatan kelincuhan, pengurangan biaya, dan ketahanan operasional. Jika kelincuhan adalah driver utama, Anda mungkin melihat berapa banyak penerapan aplikasi telah atau akan memiliki dalam setahun. Anda kemudian dapat menggunakan titik data tersebut selama prioritas. Untuk pengurangan biaya, Anda mungkin melihat penghematan tahunan yang diharapkan dari ukuran lingkungan yang tepat. Untuk ketahanan, Anda mungkin menginginkan titik data yang mewakili pendapatan yang diharapkan hilang per jam waktu henti.

Tentukan kriteria apa yang akan digunakan

Pada fase ini, Anda bekerja dengan pemangku kepentingan untuk menentukan driver bisnis dan teknis untuk memprioritaskan beban kerja yang bermigrasi. Tetapkan prioritas beban kerja migrasi sesuai dengan dampak beban kerja terhadap driver bisnis dan teknis Anda. Pilih kandidat migrasi pilot dan penggerak pertama. Ini mungkin bukan aplikasi teratas atau grup migrasi berdasarkan prioritas portofolio, karena ada kriteria khusus untuk memastikan bahwa tim migrasi dan operasi mendapat manfaat dari gelombang migrasi pilot. Anda kehendak belajar pelajaran selama setiap cutover, jadi itu adalah tipikal untuk memulai dengan beban kerja risiko yang lebih rendah.

Pilih 2—10 titik data untuk memprioritaskan beban kerja Anda. Titik-titik data ini harus mewakili perbedaan yang signifikan dalam seberapa cepat Anda akan mendapatkan nilai dari memiliki beban kerja yang dimigrasi sehingga Anda mendapatkan nilai lebih cepat. Contoh yang baik adalah memilih kekritisitas bisnis (misalnya, misi kritis atau penting) jika Anda ingin mulai bermigrasi untuk

mendapatkan pengalaman. Contoh yang buruk akan memilih unit bisnis (misalnya, pemasaran, keuangan, atau fasilitas) ketika tidak ada driver yang jelas untuk bermigrasi satu unit sebelum yang lain.

Tentukan cara menggunakan kriteria

Setelah Anda memilih poin data yang bermakna, tentukan skema penilaian untuk setiap nilai setiap titik data. Tetapkan skor yang lebih tinggi untuk aplikasi yang sangat rendah dan prioritaskan mereka untuk bermigrasi terlebih dahulu.

Setelah mencetak nilai untuk setiap titik data, Anda dapat membandingkan setiap titik data dengan yang lain. Langkah opsional ini adalah agar Anda tidak perlu khawatir untuk memastikan bahwa Anda memiliki setiap nilai setiap titik data yang selaras persis dengan prioritas Anda karena dapat menjadi luar biasa untuk tetap selaras saat Anda melakukan iterasi pada penetapan skor ke nilai.

Untuk membandingkan titik data, gunakan pengganda untuk setiap titik data. Misalnya, Anda dapat membedakan titik data Business Critical dari titik data Unit Bisnis dengan melakukan hal berikut:

0.2x	0,4x	0,6x	0,8x	1x
		Unit bisnis		Kritis bisnis

Dalam contoh ini, skor Business Critical akan tetap sama (dikalikan dengan 1) dan skor Unit Bisnis akan menjadi 60% dari skor yang ditetapkan (dikalikan dengan 0,6). Hal ini menunjukkan bahwa beban kerja nilai Business Critical lebih penting daripada nilai Unit Bisnis.

Setelah Anda menetapkan skor untuk setiap beban kerja, lihat distribusi skor di seluruh portofolio. Skor itu sendiri tidak penting. Ini adalah perbedaan antara skor yang penting. Misalnya, Anda mungkin menemukan bahwa skor teratas adalah 8.000 dan skor terbawah adalah 800.

Kami menyarankan Anda merencanakan skor sebagai histogram, sehingga Anda dapat memverifikasi bahwa Anda memiliki distribusi yang baik. Distribusi yang ideal akan terlihat seperti kurva lonceng standar, dengan beberapa beban kerja prioritas yang sangat tinggi dan beberapa beban kerja prioritas yang sangat rendah. Mayoritas beban kerja akan berada di suatu tempat di tengah.

⚠ Important

Inti dari latihan ini adalah memahami beban kerja mana yang paling berharga untuk dimigrasi terlebih dahulu. Jadi, pastikan Anda tidak memiliki skor yang sama atau serupa untuk sebagian besar beban kerja karena itu berarti bahwa semuanya memiliki prioritas yang sama. Kami menyarankan Anda melihat apa yang ada di bagian atas dan bawah daftar dan melihat apakah Anda setuju. Jika Anda umumnya tidak setuju, Anda mungkin ingin meninjau kembali kriteria yang Anda gunakan untuk mencetak beban kerja.

Perencanaan

Anda akan menetapkan setiap aplikasi, server, dan database yang ingin Anda migrasi ke gelombang. Setiap gelombang akan memiliki tanggal mulai dan tanggal akhir (cutover). Waktu antara tanggal mulai dan akhir bisa sesingkat 2 minggu atau selama 20 minggu, tergantung pada aktivitas yang diperlukan saat bersiap untuk memotong. Membuat peta jalan tingkat tinggi memungkinkan Anda untuk fokus pada detail teknis migrasi beban kerja tertentu dan cara memigrasi kumpulan beban kerja pada saat yang bersamaan. Rencana gelombang harus account untuk dependensi antara aplikasi dan infrastruktur. Membangun rencana migrasi ini akan membantu Anda memahami seperti apa timeline migrasi nantinya dan membantu Anda menetapkan harapan dalam organisasi Anda. Maksud dari peta jalan adalah untuk menjawab pertanyaan umum, “Beban kerja mana yang bergerak dan kapan?” dan “Berapa lama proyek migrasi ini akan berlangsung?” Peta jalan juga memberikan kejelasan dan fokus tentang apa yang perlu dikerjakan oleh tim migrasi inti serta tim pendukung.

Ketika pengumpulan data selesai atau cukup (yaitu, itu lengkap untuk satu beban kerja tetapi tidak semua beban kerja), Anda beralih ke tahap perencanaan. Dalam fase ini, Anda melakukan hal berikut:

- Menetapkan aplikasi dan elemen infrastruktur ke grup migrasi
- Identifikasi [Strategi migrasi](#)
- Menentukan jenis instans Elastic Compute Cloud (Amazon EC2) dan volume Elastic Block Store (Amazon EBS)

Pada titik ini, Anda dapat kembali ke tahap estimasi untuk memvalidasi atau memperbaiki perkiraan, meskipun ini tidak diperlukan.

Membuat grup migrasi adalah langkah penting untuk memastikan bahwa Anda akan dapat memenuhi tanggal akhir migrasi Anda. Saat Anda membuat grup migrasi, pertimbangkan setiap dependensi di lingkungan Anda sehingga bisnis Anda tidak mengalami hasil yang tidak terduga saat Anda bermigrasi ke cloud. Kelompok dianggap unit migrasi atom, sehingga mereka harus konsisten dengan beban kerja yang harus bermigrasi pada saat bersamaan.

Pertahankan grup migrasi sekecil mungkin, sekaligus menjaga integritas kinerja sistem selama dan setelah migrasi. Grup migrasi kecil lebih mudah untuk beralih daripada grup yang lebih besar. Kelompok-kelompok kecil memungkinkan Anda menangani konflik yang tak terhindarkan saat mereka muncul. Misalnya, jika ada pemadaman prioritas 1 menit terakhir dan Anda harus memindahkan grup pada waktu yang berbeda, Anda dapat menyesuaikan dengan grup kecil. Kelompok besar tidak memungkinkan fleksibilitas ini.

Sebagai aturan umum, grup migrasi tidak boleh melebihi 20 aplikasi, 150 server, dan 30 database. Migrasi lebih dari ini dalam satu cutover dapat membuktikan sangat menantang logistik. Ini juga secara signifikan meningkatkan peluang kegagalan Anda, dan mungkin memerlukan rollback atau jendela pemadaman yang diperpanjang. Kami menyarankan Anda untuk membenarkan dan memeriksa secara kritis setiap pengecualian terhadap aturan ini.

Fokus sebagian besar waktu Anda pada pemisahan kelompok terbesar. Cari tahu apakah ada kemenangan mudah untuk dibagi menjadi kelompok-kelompok yang lebih kecil. Misalnya, memahami dependensi tunggal memungkinkan Anda membagi grup menjadi dua. Ingat bahwa kelompok yang lebih kecil memungkinkan lebih banyak kelincahan dalam fase migrasi. Namun, ini adalah latihan yang kompleks untuk dilakukan pada keseluruhan portofolio, dan analisis harus fokus pada di mana Anda akan mendapatkan kelincahan yang berharga. Anda tidak perlu menganalisis setiap kelompok. Tetapi Anda harus meninjau setiap kelompok, bahkan jika hanya untuk beberapa saat.

Buat grup migrasi

Membuat kelompok migrasi adalah seni dan sains. Anda harus membuat keputusan berbasis data dan intuitif. Ide mendasar di balik kelompok migrasi adalah bahwa Anda menetapkan batasan yang jelas tentang apa yang dilakukan dan tidak perlu bermigrasi pada saat yang sama. Untuk melakukannya, Anda harus melihat secara kritis pada dependensi beban kerja Anda.

Berikut ini adalah beberapa poin data utama yang harus Anda evaluasi sebagai bagian dari menilai grup migrasi:

- Database bersama
- Server bersama

- Komunikasi Aplikasi

Ada jenis dependensi lain yang bisa menjadi faktor untuk kelompok migrasi. Tetapi panduan ini berfokus pada database bersama, server bersama, dan komunikasi aplikasi karena mereka hadir di hampir setiap lingkungan dan memiliki dampak paling signifikan pada bagaimana grup Anda akan dibuat.

Asalkan mereka tidak memiliki dependensi satu sama lain, cobalah untuk menjaga lingkungan aplikasi yang berbeda dalam kelompok yang berbeda. Misalnya, letakkan lingkungan produksi CRM dalam kelompok yang berbeda dari lingkungan pengembangan dan pengujian CRM. Hal ini memungkinkan Anda untuk memigrasi lingkungan non-produksi terlebih dahulu, sehingga Anda dapat menguji proses migrasi dan konfigurasi AWS.

Mengevaluasi database

Saat Anda mengevaluasi database bersama, fokuslah pada hal berikut:

- Pertama, putuskan apakah Anda berencana untuk memigrasi database atau server dengan semua database-nya. Umumnya, kami sarankan untuk memindahkan database dengan sendirinya karena membantu membuat grup yang lebih kecil, terutama jika Anda memiliki server database bersama yang sangat besar dengan banyak database yang tidak terkait.
- Kedua, memutuskan apakah aplikasi harus bergerak dengan database. Server aplikasi harus berada di dekat server database untuk alasan performa. Umumnya, jawabannya adalah “ya, itu harus bergerak bersama,” tetapi meneliti pertanyaan ini dapat memberi Anda fleksibilitas yang signifikan dalam migrasi.

Misalnya, dan Aplikasi A, dan C semuanya terhubung ke db01. A dan B keduanya erat digabungkan ke database, dan mereka melakukan banyak menulis dan membaca operasi setiap detik. Aplikasi C, bagaimanapun, menjalankan pekerjaan batch setiap malam pada jam-jam sibuk. Anda menentukan bahwa Aplikasi A dan B harus bergerak dengan db01, tetapi Aplikasi C dapat bergerak secara terpisah. Oleh karena itu, Aplikasi A, Aplikasi B, dan db01 ditempatkan di Grup 1 dan Aplikasi C ditempatkan di Grup 2.

Mengevaluasi server

Mengevaluasi server bersama seringkali jauh lebih mudah daripada mengevaluasi database bersama. Sebagian besar organisasi menjaga server aplikasi yang didedikasikan untuk lingkungan

aplikasi dengan beberapa pengecualian umum. Server web adalah contoh server bersama yang umum. Saat memutuskan cara menangani server bersama, coba cari tahu apakah fungsi server mudah diduplikasi. Dalam kasus server web, ini sering benar karena sepele untuk membuat server baru, menginstal perangkat lunak server web, dan memperbarui konfigurasi.

Misalnya, Aplikasi A, B, dan C semuanya menggunakan server web Apache untuk lalu lintas web mereka. Aplikasi A dan B juga berbagi server aplikasi lain, sedangkan Application C memiliki server aplikasi khusus sendiri. Anda menentukan bahwa Aplikasi A dan B akan bermigrasi dengan server web Apache di Grup 1. Aplikasi C akan menggunakan server web baru bersih dan bermigrasi di Grup 2.

Mengevaluasi komunikasi

Memahami aplikasi mana yang harus bergerak bersama adalah tempat seni proses ini mulai berperan. Jika Anda mencoba mengelompokkan segala sesuatu yang memiliki ketergantungan, Anda mungkin akan menemukan bahwa sebagian besar portofolio Anda, terkadang setinggi 80 persen, terhubung melalui rantai dependensi (misalnya, Aplikasi A memerlukan Aplikasi B, yang pada gilirannya memerlukan Aplikasi C, dan seterusnya). Karena biasanya tidak layak untuk memigrasi sebagian besar portofolio Anda sekaligus, Anda mungkin akan dipaksa untuk memisahkan aplikasi yang idealnya akan Anda migrasi pada saat yang sama.

Ini bisa menjadi percakapan yang menantang untuk dimiliki dengan ahli subjek aplikasi karena respons default sering, “Tidak, mereka perlu bermigrasi pada saat yang sama; Saya membutuhkan aplikasi itu!” Cara Anda menyampaikan pertanyaan Anda memiliki dampak signifikan dalam mendapatkan respons berbasis data. Pertanyaan apakah aplikasi harus bermigrasi pada saat yang sama paling baik dinyatakan sebagai, “Apakah aplikasi ini digabungkan erat?” Jika jawabannya ya, selami lebih dalam dan ajukan pertanyaan tindak lanjut ini: “Apa yang akan terjadi jika aplikasi ini offline pada waktu yang terpisah?” dan “Bagaimana jika mereka dipisahkan secara geografis dan memiliki (misalnya) 30 ms latensi dalam komunikasi mereka?” Ini akan membantu Anda memahami di mana batasan untuk memisahkan aplikasi ke dalam kelompok yang berbeda sebenarnya.

Sebagai contoh, Aplikasi A memerlukan Aplikasi B memerlukan Aplikasi C Anda menentukan bahwa jika Aplikasi A dan B dimigrasi secara terpisah, akan ada dampak kinerja yang signifikan untuk Aplikasi A Anda juga menentukan bahwa Aplikasi B akan kehilangan sebagian kecil fungsi sementara Aplikasi C sedang bermigrasi dan offline. Anda memutuskan untuk menempatkan Aplikasi A dan Aplikasi B di Grup 1 dan Aplikasi C di Grup 2.

Alat untuk

Proses ini bisa menantang, terutama jika Anda tidak memiliki pengalaman sebelumnya dan spreadsheet adalah satu-satunya alat yang Anda miliki. Kami menyarankan Anda [AWS Mitra](#) yang memiliki pengalaman dengan latihan ini dan dapat membantu menuntun Anda melewatinya. AWS Mitra memiliki akses ke alat dan sumber daya untuk membantu analisis ini.

Jika Anda tidak dapat menggunakan mitra, sebaiknya Anda memvisualisasikan dependensi di lingkungan Anda. Database grafik adalah alat yang secara inheren baik yang dapat membantu Anda menganalisis aplikasi, database, dan server Anda sebagai node atau simpul, dan kemudian juga menelan dependensi sebagai tepi.

Ubah grup menjadi rencana

Setelah Anda menyelesaikan latihan pengelompokan migrasi, mulailah menjadwalkan grup menjadi wave. Setiap grup migrasi harus diberi tanggal mulai dan akhir yang diantisipasi. Kelompok dengan tanggal akhir yang sama dianggap bagian dari gelombang yang sama.

Pertimbangkan beban kerja yang idealnya bermigrasi pada saat yang sama saat menetapkan tanggal ke setiap grup. Migrasi yang berhasil memerlukan kontribusi yang signifikan dari berbagai bagian bisnis, jadi publikasikan rencana migrasi di seluruh bisnis Anda. Tim akan ingin meninjau dan menjadwalkan kontribusi mereka (misalnya, menentukan kapan mereka harus mengkomunikasikan waktu singkat kepada pengguna) dan mempersiapkan diri untuk transformasi dengan pelatihan, membuat rencana operasional, dan sebagainya.

Anda mungkin memerlukan beberapa penemuan tambahan saat Anda bekerja melalui rencana migrasi. Aktivitas ini mencakup penyelaman mendalam ke dalam arsitektur aplikasi tempat Anda menangkap data tambahan yang diperlukan untuk berhasil merancang, membangun, dan memigrasi aplikasi ke AWS.

Sumber daya

Alat

- [AWSApplication Discovery Service](#)
- [Alat dari Mitra KompetensiAWS Migrasi](#)

Mitra

- [Mitra konsultasi migrasi](#)
- [Mitra pengiriman migrasi](#)

Pola SIMOLATIF

- [AWSPanduan Panduan Preskriptif](#)
- [AWSPola Bimbingan Preskriptif](#)

Pertanyaan yang Sering Diajukan

Bagian ini memberikan jawaban atas pertanyaan yang sering diajukan tentang penemuan dan analisis portofolio selama migrasi.

Bagaimana cara membangun kasus bisnis yang tepat?

Kasus bisnis Anda harus didorong oleh KPI organisasi Anda, dan pengemudi umum seperti biaya operasional, produktivitas tenaga kerja, penghindaran biaya, ketahanan operasional, dan kelincahan bisnis.

Bagaimana cara menilai lingkungan saya secara akurat?

Pertimbangkan volume sumber daya yang digunakan oleh setiap aplikasi, dan otomatisasi proses penilaian untuk mengkonfirmasi hal itu dilakukan dengan cepat dan akurat. Menilai lingkungan Anda secara manual adalah proses yang memakan waktu. Ini mengekspos organisasi Anda untuk kesalahan manusia. Mengotomatisasi proses akan membantu Anda mendapatkan wawasan tentang apa yang tidak Anda ketahui, dan membantu Anda lebih memahami dan mendefinisikan ketidakpastian ini sehingga mereka dapat diperhitungkan dalam strategi migrasi Anda.

Bagaimana cara mengidentifikasi dan mengevaluasi mitra yang tepat untuk membantu saya?

Rincian tentang penawaran mitra dapat ditemukan di:

- [AWSSolusi Mitra](#)
- [Solusi diAWSMarketplace](#)

Bagaimana cara memperkirakan biaya transisi besar seperti ini?

Menggunakan [AWSKalkulator Harga](#) untuk membandingkan berapa banyak yang akan Anda habiskan untuk menjalankan aplikasi Anda di lingkungan lokal atau colocation, dengan biaya apaAWS.

Berapa lama proses migrasi akan selesai?

Migrasi perusahaan yang diselesaikan dalam waktu 18 bulan menghasilkan ROI terbesar. Durasi migrasi tergantung pada ruang lingkup dan sumber daya.

Bagaimana cara menangani aplikasi warisan saya?

Pertimbangkan untuk mengambil pendekatan tambahan untuk migrasi Anda dengan menentukan aplikasi warisan mana yang dapat dipindahkan paling mudah. Pindahkan ini ke cloud terlebih dahulu. Untuk aplikasi warisan yang memerlukan pendekatan yang lebih rumit, Anda dapat mengembangkan rencana migrasi yang efektif.

Bagaimana cara mempercepat upaya migrasi untuk mewujudkan manfaat bisnis dan teknologi dengan lebih cepat?

Mengotomatisasi proses migrasi sebanyak mungkin, dengan alat migrasi dari AWS dan Mitra APN, adalah cara terbaik untuk mempercepat upaya migrasi.

Ucapan Terima Kasih

Penulis mengakui kontribusi Douglas Mugnos, Cloud Application Architect, AWS Layanan Profesional, untuk penulisan bagian [Memilih dan menerapkan alat penemuan](#).

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan di future, Anda dapat berlangganan ke [Umpan RSS](#).

Perubahan	Deskripsi	Tanggal
Ditambahkan bagian	Kami menambahkan informasi tentang memilih dan menerapkan alat penemuan .	2 Februari 2022
Pedoman yang diperbarui	Kami menambahkan rincian lebih lanjut ke pedoman untuk pengumpulan data, prioritas, dan perencanaan.	13 Desember 2019
Publikasi awal	—	5 Agustus 2019

AWSGlosarium 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 Cloud. AWS
- 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 Cloud. AWS
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Skenario migrasi ini khusus untuk VMware Cloud onAWS, yang mendukung kompatibilitas mesin virtual (VM) dan portabilitas beban kerja antara lingkungan lokal Anda dan. AWS Anda dapat menggunakan teknologi VMware Cloud Foundation dari pusat data lokal saat memigrasikan infrastruktur ke VMware Cloud. AWS Contoh: Pindahkan hypervisor yang menghosting database Oracle Anda ke VMware Cloud on. 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

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.

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-ArchitectedAWS.

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](#).

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.

Cloud Center of Excellence (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 Enterprise.

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 AWS Cloud Enterprise Strategy. 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 AWS CodeCommit. 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 atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer

Bidang AI yang digunakan oleh mesin untuk mengidentifikasi orang, tempat, dan benda dalam gambar dengan akurasi pada atau di atas tingkat manusia. Sering dibangun dengan model pembelajaran mendalam, ini mengotomatiskan ekstraksi, analisis, klasifikasi, dan pemahaman informasi yang berguna dari satu gambar atau urutan gambar.

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 umumnya 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](#).

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.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data 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)

Proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang berdampak buruk pada 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 database](#).

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

pengecik, 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](#).

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.

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).

enkripsi amplop

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

lingkungan

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

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang pengguna akhir dapat mengakses. Dalam pipa CI/CD, 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](#).

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.

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.

G

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 disukai.

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, 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.

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

I

IAC

Lihat [infrastruktur sebagai kode](#).

kebijakan berbasis identitas

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

aplikasi idle

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

IloT

Lihat [Internet of Things industri](#).

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi 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.

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 selengkapnya, 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 VPC (dalam hal yang sama atau berbedaWilayah AWS), internet, dan jaringan lokal. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi 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 lebih lanjut, lihat [Apa itu IoT?](#)

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi selengkapnya, 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](#).

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

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

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](#).

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.

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.

layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik 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 API ringan. 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 terbaik dan pelajaran yang dipetik 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 Cloud. AWS 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 Cloud. AWS 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](#).

MPA

Lihat [Penilaian Portofolio Migrasi](#).

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 Cloud. AWS](#)

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 selengkapnya, lihat [Mengevaluasi kesiapan modernisasi untuk aplikasi](#) di Cloud. AWS

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 [Menguraikan monolit](#) menjadi layanan mikro.

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.

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-ArchitectedAWS.

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 diAWS 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](#).

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 VPC masuk, keluar, dan inspeksi 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.

kebijakan

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

ketekunan poliglott

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. Untuk informasi selengkapnya, lihat [Mengaktifkan persistensi data di layanan mikro](#).

penilaian portofolio

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

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

Sebuah entitas dalam 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 oleh Desain

Pendekatan dalam rekayasa sistem yang memperhitungkan privasi di seluruh proses rekayasa. 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 beberapa VPC. 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.

lingkungan produksi

Lihat [lingkungan](#).

pseudonimisasi

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

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 \(RACI\)](#).

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 hilangnya 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](#).

Wilayah

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 [Mengelola Wilayah AWS](#) di Referensi Umum AWS.

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](#).

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](#).

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 AWS Management Console 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.

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadatanya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat [Rahasia](#) dalam dokumentasi Secrets Manager.

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 SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai 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 fileLayanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat [Layanan AWStitik 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.

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.

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda dapat 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 VPC dan jaringan lokal Anda. 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 VPC yang memungkinkan Anda 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 saat ini 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 [Kerangka Kualifikasi Beban Kerja AWS](#).

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.

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.