Menggunakan tabel global Amazon DynamoDB

AWS Bimbingan Preskriptif

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Bimbingan Preskriptif: Menggunakan tabel global Amazon DynamoDB

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 pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Pengantar ... 1
Gambaran Umum ... 2
Fakta kunci ... 2
Kasus penggunaan ... 3
Mode tulis ... 5
Menulis ke mode Region (tidak ada primer) ... 5
Menulis ke satu mode Region (primer tunggal) ... 8
Menulis ke mode Wilayah Anda (campuran primer) ... 10
Strategi perutean ... 13
Perutean permintaan berbasis klien ... 13
Perutean permintaan lapisan komputasi ... 15
Perutean permintaan Route 53 ... 17
Perutean permintaan Global Accelerator ... 18
Proses evakuasi ... 20
Mengevakuasi Wilayah langsung ... 20
Mengevakuasi offline ... 20
Perencanaan kapasitas throughput ... 23
Daftar periksa persiapan ... 25
Pertanyaan yang Sering Diajukan ... 27
Berapa harga untuk tabel global? ... 27
Wilayah mana yang didukung tabel global? ... 27
Bagaimana GSI ditangani dengan tabel global? ... 27
Bagaimana cara menghentikan replikasi tabel global? ... 28
Bagaimana Amazon DynamoDB Streams berinteraksi dengan tabel global? ... 28
Bagaimana tabel global menangani transaksi? ... 28
Bagaimana tabel global berinteraksi dengan cache DynamoDB Accelerator (DAX)? ... 28
Apakah tag pada tabel disebarkan? ... 29
Haruskah saya membuat cadangan tabel di semua Wilayah atau hanya satu? ... 29
Bagaimana cara menerapkan tabel global dengan menggunakan? AWS CloudFormation ... 29
Kesimpulan dan sumber daya ... 31
Riwayat dokumen ... 32
Glosarium ... 33
... 33
A ... 34

B 37
C 39
D 42
E 46
F 48
G 49
H 50
I 51
L 54
M 55
O 59
P 61
Q 64
R 65
D 67
T 71
U 73
V 73
W 74
Z 75
..... lxxvi

Menggunakan tabel global Amazon DynamoDB

Jason Hunter, Amazon Web Services ()AWS

Maret 2024 (sejarah dokumen)

Tabel global dibangun di atas jejak Amazon DynamoDB global untuk memberi Anda basis data multi-aktif yang dikelola sepenuhnya, Multi-wilayah, dan multi-aktif yang memberikan kinerja baca dan tulis lokal yang cepat dan lokal untuk aplikasi global yang diskalakan secara besar-besaran. Tabel global mereplikasi tabel DynamoDB Anda secara otomatis di seluruh pilihan Anda. Wilayah AWS Tidak ada perubahan aplikasi yang diperlukan karena tabel global menggunakan DynamoDB API yang ada. Tidak ada biaya atau komitmen di muka untuk menggunakan tabel global, dan Anda hanya membayar untuk sumber daya yang Anda gunakan.

Panduan ini menjelaskan cara menggunakan tabel global DynamoDB secara efektif. Ini memberikan fakta kunci tentang tabel global, menjelaskan kasus penggunaan utama fitur, memperkenalkan taksonomi dari tiga model penulisan berbeda yang harus Anda pertimbangkan, menelusuri empat pilihan perutean permintaan utama yang mungkin Anda terapkan, membahas cara untuk mengevakuasi Wilayah yang hidup atau Wilayah yang offline, menjelaskan cara memikirkan perencanaan kapasitas throughput, dan menyediakan daftar periksa hal-hal yang perlu dipertimbangkan saat Anda menerapkan tabel global.

Panduan ini cocok dengan konteks penyebaran AWS Multi-wilayah yang lebih besar, seperti yang tercakup dalam whitepaper AWS Multi-Region Fundamentals dan pola desain ketahanan data dengan video. AWS

Daftar Isi

- Ikhtisar
- Tulis mode
- Strategi perutean
- Proses evakuasi
- Perencanaan kapasitas throughput
- Daftar periksa persiapan
- FAQ
- Kesimpulan dan sumber daya

Ikhtisar tabel global

Fakta kunci

- Ada dua versi tabel global: versi 2017.11.29 (warisan) (kadang-kadang disebut v1) dan versi 2019.11.21 (saat ini) (kadang-kadang disebut v2). Panduan ini berfokus secara eksklusif pada versi saat ini.
- DynamoDB (tanpa tabel global) adalah layanan Regional, yang berarti bahwa layanan ini sangat tersedia dan secara intrinsik tahan terhadap kegagalan infrastruktur, termasuk kegagalan seluruh Availability Zone. Tabel DynamoDB Single-region dirancang untuk ketersediaan 99,99%. Untuk informasi lebih lanjut, lihat Perjanjian tingkat layanan (SLA) DynamoDB.
- Tabel global DynamoDB mereplikasi datanya antara dua Wilayah atau lebih. Tabel DynamoDB Multi-wilayah dirancang untuk ketersediaan 99,999%. Dengan perencanaan yang tepat, tabel global dapat membantu menciptakan arsitektur yang tangguh terhadap kegagalan Regional.
- Tabel global menggunakan model replikasi aktif-aktif. Dari perspektif DynamoDB, tabel di setiap Wilayah memiliki kedudukan yang sama untuk menerima permintaan baca dan tulis. Setelah menerima permintaan tulis, tabel replika lokal mereplikasi operasi penulisan ke Wilayah terpencil lainnya yang berpartisipasi di latar belakang.
- Item direplikasi secara individual. Item yang diperbarui dalam satu transaksi mungkin tidak direplikasi bersama.
- Setiap partisi tabel di Wilayah sumber mereplikasi operasi penulisannya secara parallel dengan setiap partisi lainnya. Urutan operasi tulis dalam Wilayah terpencil mungkin tidak cocok dengan urutan operasi tulis yang terjadi di dalam Wilayah sumber. Untuk informasi selengkapnya tentang partisi tabel, lihat posting blog Scaling DynamoDB: Bagaimana partisi, tombol pintas, dan split untuk kinerja dampak panas.
- Satu item baru yang ditulis biasanya disebarkan ke semua tabel replika dalam satu detik. Daerah terdekat cenderung merambat lebih cepat.
- Amazon CloudWatch menyediakan `ReplicationLatency` metrik untuk setiap pasangan Wilayah. Ini dihitung dengan melihat item yang tiba, membandingkan waktu kedatangan mereka dengan waktu penulisan awal mereka, dan menghitung rata-rata. Timing disimpan CloudWatch dalam di wilayah sumber. Melihat timing rata-rata dan maksimum dapat berguna untuk menentukan lag replikasi rata-rata dan terburuk. Tidak ada SLA pada latensi ini.
- Jika item individual diperbarui pada waktu yang hampir bersamaan (dalam `ReplicationLatency` jendela ini) di dua Wilayah yang berbeda, dan operasi penulisan kedua terjadi sebelum operasi

penulisan pertama direplikasi, ada potensi konflik penulisan. Tabel global menyelesaikan konflik tersebut dengan menggunakan mekanisme menang penulis terakhir, berdasarkan stempel waktu operasi tulis. Operasi pertama “kalah” ke operasi kedua. Konflik ini tidak dicatat dalam CloudWatch atauAWS CloudTrail.

- Setiap item memiliki stempel waktu tulis terakhir yang dipegang sebagai properti sistem pribadi. Pendekatan win penulis terakhir diimplementasikan dengan menggunakan operasi penulisan bersyarat yang mengharuskan stempel waktu item masuk menjadi lebih besar dari stempel waktu item yang ada.
- Tabel global mereplikasi semua item ke semua Wilayah yang berpartisipasi. Jika Anda ingin memiliki cakupan replikasi yang berbeda, Anda dapat membuat beberapa tabel global dan menetapkan setiap tabel Wilayah yang berpartisipasi berbeda.
- Wilayah lokal menerima operasi penulisan meskipun Region replika sedang offline atau `ReplicationLatency` tumbuh. Tabel lokal terus mencoba mereplikasi item ke tabel jarak jauh hingga setiap item berhasil.
- Jika suatu Wilayah berjalan sepenuhnya offline, ketika kembali online nanti, semua replikasi keluar dan masuk yang tertunda akan dicoba ulang. Tidak ada tindakan khusus yang diperlukan untuk membawa tabel kembali sinkron. Penulis terakhir menang mekanisme memastikan bahwa data akhirnya menjadi konsisten.
- Anda dapat menambahkan Wilayah baru ke tabel DynamoDB kapan saja. DynamoDB menangani sinkronisasi awal dan replikasi yang sedang berlangsung. Anda juga dapat menghapus Region (bahkan Region asli), dan ini akan menghapus tabel lokal di Wilayah tersebut.
- DynamoDB tidak memiliki endpoint global. Semua permintaan dibuat ke endpoint Regional yang mengakses instance tabel global yang bersifat lokal ke Wilayah tersebut.
- Panggilan ke DynamoDB tidak boleh melintasi Wilayah. Praktik terbaik adalah untuk aplikasi yang homed ke satu Region untuk langsung mengakses hanya endpoint DynamoDB lokal untuk Region nya. Jika masalah terdeteksi dalam Wilayah (di lapisan DynamoDB atau di tumpukan sekitarnya), lalu lintas pengguna akhir harus dirutekan ke titik akhir aplikasi berbeda yang di-host di Wilayah yang berbeda. Tabel global memastikan bahwa aplikasi homed di setiap Wilayah memiliki akses ke data yang sama.

Kasus penggunaan

Tabel global memberikan manfaat umum ini:

- Operasi baca latensi rendah. Anda dapat menempatkan salinan data lebih dekat ke pengguna akhir untuk mengurangi latensi jaringan selama operasi baca. Data disimpan sesegar `ReplicationLatency` nilainya.
- Operasi penulisan latensi rendah. Pengguna akhir dapat menulis ke Wilayah terdekat untuk mengurangi latensi jaringan dan waktu untuk menyelesaikan operasi penulisan. Lalu lintas tulis harus dialihkan dengan hati-hati untuk memastikan bahwa tidak ada konflik. Teknik untuk routing dibahas di bagian selanjutnya.
- Peningkatan ketahanan dan pemulihan bencana. Jika suatu Wilayah mengalami penurunan kinerja atau pemadaman penuh, Anda dapat mengevakuasinya (memindahkan sebagian atau semua permintaan ke Wilayah tersebut) dan memenuhi tujuan titik pemulihan (RPO) dan tujuan waktu pemulihan (RTO) yang diukur dalam hitungan detik. Menggunakan tabel global juga meningkatkan DynamoDB SLA untuk persentase waktu aktif bulanan dari 99,99% menjadi 99,999%.
- Migrasi Wilayah Seamless. Anda dapat menambahkan Region baru dan kemudian menghapus Region lama untuk memigrasi penyebaran dari satu Region ke Region lainnya, tanpa downtime pada lapisan data.

Misalnya, Fidelity Investments disajikan di Re:Invent 2022 tentang bagaimana mereka menggunakan tabel global DynamoDB untuk Sistem Manajemen Pesanan mereka. Tujuan mereka adalah untuk mencapai pemrosesan latensi rendah yang andal pada skala yang tidak dapat mereka capai dengan pemrosesan lokal sambil juga mempertahankan ketahanan terhadap Availability Zone dan kegagalan Regional.

Menulis mode untuk tabel global

Tabel global selalu aktif aktif di tingkat tabel. Namun, Anda mungkin ingin memperlakukan mereka sebagai aktif-pasif dengan mengontrol cara Anda merutekan permintaan penulisan. Misalnya, Anda mungkin memutuskan untuk merutekan permintaan penulisan ke satu Wilayah untuk menghindari potensi konflik penulisan.

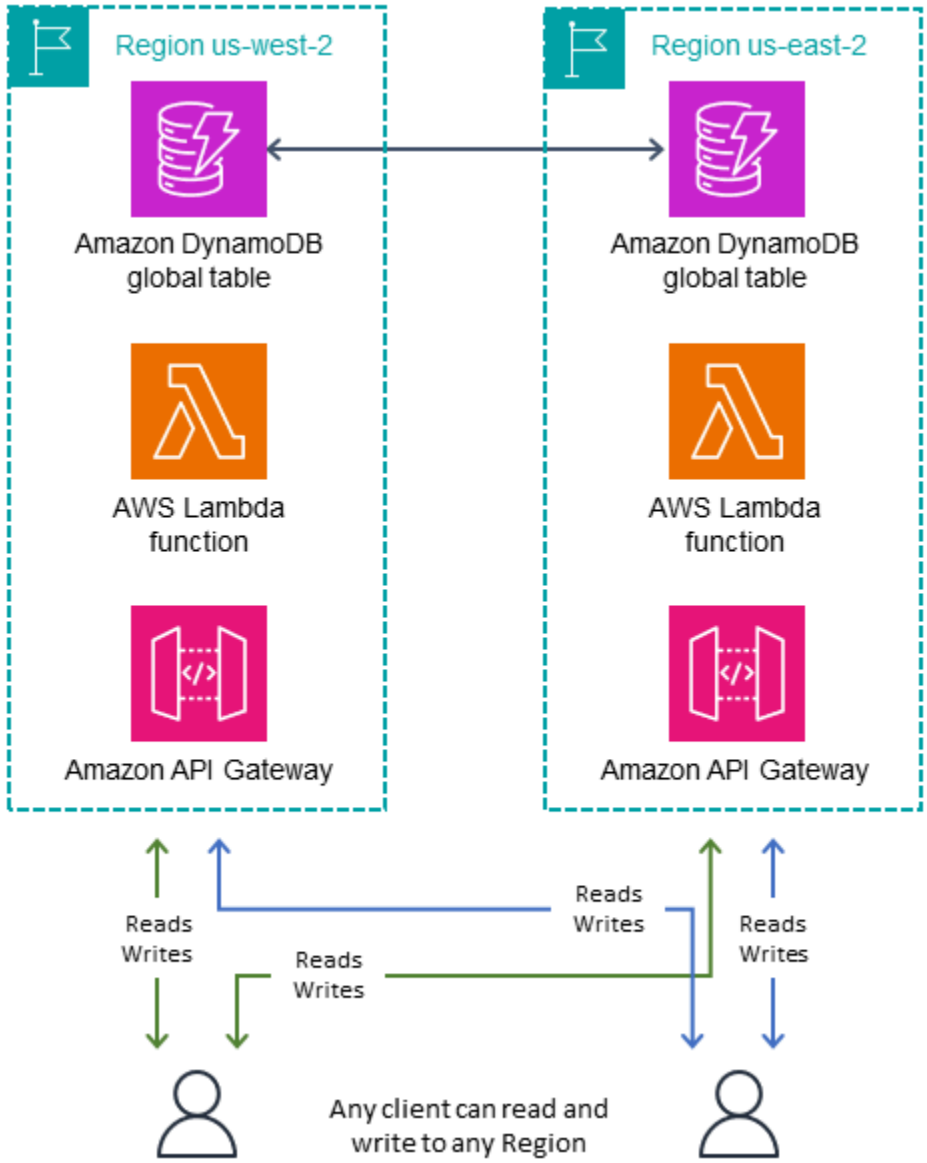
Ada tiga pola tulis terkelola utama, seperti yang dijelaskan dalam tiga bagian berikutnya. Anda harus mempertimbangkan pola tulis mana yang sesuai dengan kasus penggunaan Anda. Pilihan ini memengaruhi cara Anda merutekan permintaan, mengevakuasi Wilayah, dan menangani pemulihan bencana. Panduan di bagian selanjutnya bergantung pada mode tulis aplikasi Anda.

Topik

- Menulis ke mode Region (tidak ada primer)
- Menulis ke satu mode Region (primer tunggal)
- Menulis ke mode Wilayah Anda (campuran primer)

Menulis ke mode Region (tidak ada primer)

Modus menulis ke setiap Region write sepenuhnya aktif-aktif dan tidak memberlakukan pembatasan di mana operasi menulis dapat terjadi. Wilayah mana pun dapat menerima permintaan tulis kapan saja. Ini adalah mode yang paling sederhana; Namun, ini hanya dapat digunakan dengan beberapa jenis aplikasi. Ini cocok ketika semua operasi tulis idempoten. Idempoten berarti bahwa mereka dapat diulang dengan aman sehingga operasi penulisan bersamaan atau berulang di seluruh Wilayah tidak bertentangan — misalnya, ketika pengguna memperbarui data kontak mereka. Ini juga bekerja dengan baik untuk dataset tambahan saja di mana semua operasi tulis adalah sisipan unik di bawah kunci primer deterministik, yang merupakan kasus khusus menjadi idempoten. Terakhir, mode ini cocok di mana risiko operasi tulis yang bertentangan dapat diterima.



The write to any Region mode adalah arsitektur yang paling mudah untuk diterapkan. Routing lebih mudah karena setiap Wilayah dapat menjadi target menulis setiap saat. Failover lebih mudah, karena setiap operasi menulis baru-baru ini dapat diputar ulang beberapa kali untuk setiap Region sekunder. Jika memungkinkan, Anda harus merancang untuk mode tulis ini.

Misalnya, beberapa layanan streaming video menggunakan tabel global untuk melacak bookmark, ulasan, tanda status menonton, dan sebagainya. Penyebaran ini dapat menggunakan write ke mode Region apa pun selama mereka memastikan bahwa setiap operasi penulisan idempoten. Ini akan terjadi jika setiap pembaruan—misalnya, menyetel kode waktu terbaru baru, menetapkan ulasan baru, atau menyetel status jam tangan baru—menetapkan status baru pengguna secara langsung, dan nilai berikutnya yang benar untuk item tidak bergantung pada nilainya saat ini. Jika, secara

kebetulan, permintaan tulis pengguna dialihkan ke Wilayah yang berbeda, operasi penulisan terakhir akan bertahan dan status global akan menyelesaikan sesuai dengan tugas terakhir. Operasi baca dalam mode ini pada akhirnya akan menjadi konsisten, tertunda oleh `ReplicationLatency` nilai terbaru.

Dalam contoh lain, sebuah perusahaan jasa keuangan menggunakan tabel global sebagai bagian dari sistem untuk mempertahankan penghitungan menjalankan pembelian kartu debit untuk setiap pelanggan, untuk menghitung bahwa hadiah uang kembali pelanggan. Transaksi baru mengalir dari seluruh dunia dan pergi ke beberapa Wilayah. Perusahaan ini dapat menggunakan write ke mode Region apa pun dengan desain ulang yang cermat. Sketsa desain awal mempertahankan satu `RunningBalance` item per pelanggan. Tindakan pelanggan memperbarui saldo dengan `ADD` ekspresi, yang tidak idempoten (karena nilai baru yang benar tergantung pada nilai saat ini), dan saldo tidak sinkron jika ada dua operasi tulis ke saldo yang sama pada waktu yang sama di Wilayah yang berbeda. Desain ulang menggunakan streaming acara, yang berfungsi seperti buku besar dengan alur kerja khusus tambahan. Setiap tindakan pelanggan menambahkan item baru ke koleksi item yang dipelihara untuk pelanggan tersebut. (Koleksi item adalah kumpulan item yang berbagi kunci primer tetapi memiliki kunci semacam yang berbeda.) Setiap operasi tulis adalah sisipan idempoten yang menggunakan ID pelanggan sebagai kunci partisi dan ID transaksi sebagai kunci sortir. Desain ini membuat perhitungan keseimbangan lebih sulit karena membutuhkan `Query` untuk menarik item diikuti oleh beberapa matematika sisi klien, tetapi itu membuat semua operasi tulis idempoten dan mencapai penyederhanaan yang signifikan dalam routing dan failover. (Ini dibahas lebih detail dalam panduan ini.)

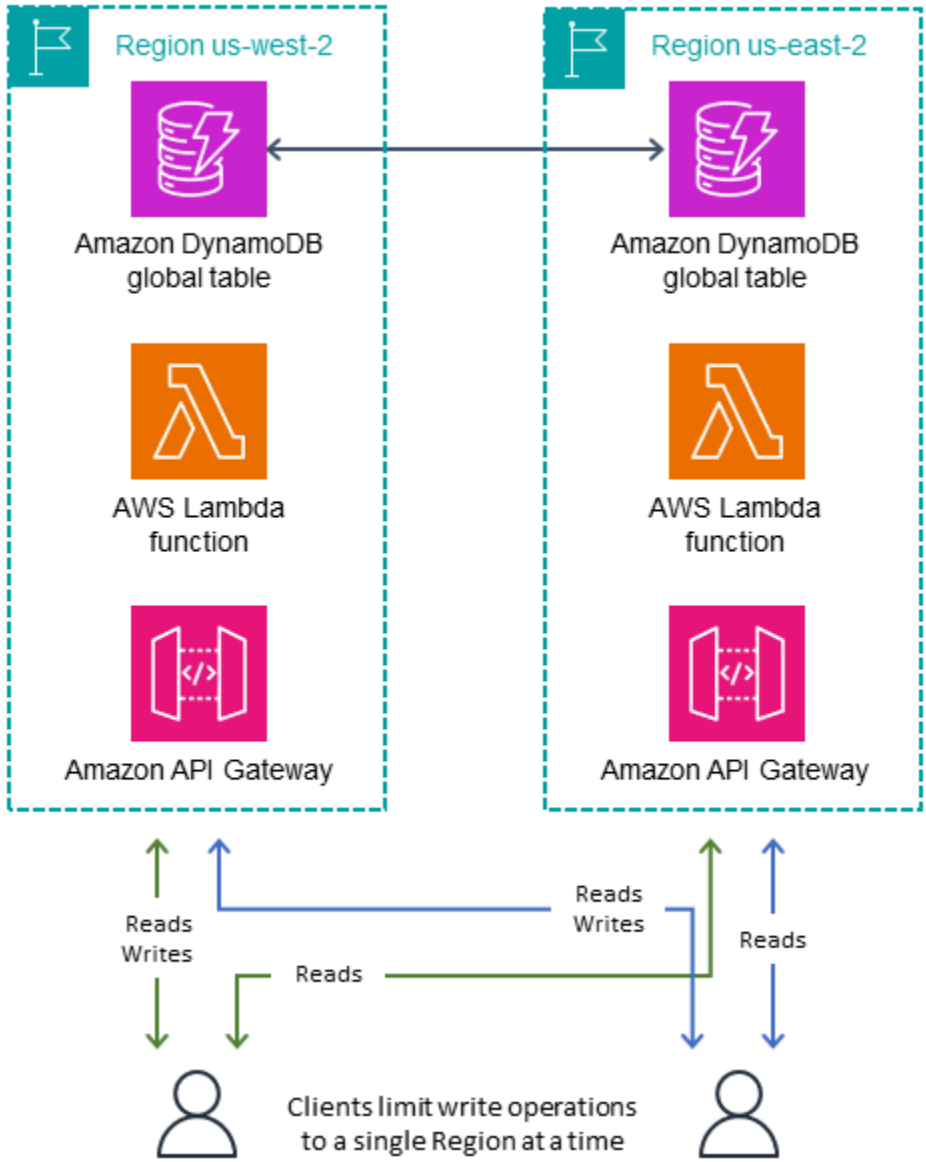
Contoh ketiga melibatkan perusahaan yang menyediakan layanan penempatan iklan online. Perusahaan ini memutuskan bahwa risiko kehilangan data yang rendah akan dapat diterima untuk mencapai penyederhanaan desain penulisan ke mode Wilayah apa pun. Saat menayangkan iklan, mereka hanya memiliki beberapa milidetik untuk mengambil metadata yang cukup untuk menentukan iklan mana yang akan ditampilkan, lalu merekam tayangan iklan sehingga tidak segera mengulangi iklan yang sama. Mereka menggunakan tabel global untuk mendapatkan operasi baca latensi rendah untuk pengguna akhir di seluruh dunia dan operasi penulisan latensi rendah. Mereka merekam semua tayangan iklan untuk pengguna dalam satu item, yang direpresentasikan sebagai daftar yang terus berkembang. Mereka menggunakan satu item alih-alih menambahkan koleksi item, sehingga mereka dapat menghapus tayangan iklan yang lebih lama sebagai bagian dari setiap operasi penulisan tanpa membayar operasi penghapusan. Operasi penulisan ini tidak idempoten; jika pengguna akhir yang sama melihat iklan yang ditayangkan dari beberapa Wilayah pada waktu yang kira-kira bersamaan, ada kemungkinan operasi penulisan untuk tayangan iklan dapat menimpa

yang lain. Risikonya adalah pengguna mungkin melihat iklan berulang sesekali. Mereka memutuskan bahwa ini dapat diterima.

Menulis ke satu mode Region (primer tunggal)

Modus write to one Region write aktif-pasif dan merutekan semua operasi tulis tabel ke Region aktif tunggal. (DynamoDB tidak memiliki gagasan tentang Region aktif tunggal; lapisan di luar DynamoDB mengelola ini.) Mode write to one Region menghindari konflik penulisan dengan memastikan bahwa operasi penulisan hanya mengalir ke satu Wilayah pada satu waktu. Mode tulis ini membantu ketika Anda ingin menggunakan ekspresi atau transaksi bersyarat. Ekspresi ini tidak memungkinkan kecuali Anda tahu bahwa Anda bertindak terhadap data terbaru, sehingga mereka memerlukan pengiriman semua permintaan tulis ke satu Wilayah yang memiliki data terbaru.

Akhirnya operasi baca yang konsisten dapat pergi ke salah satu Replika Regions untuk mencapai latensi yang lebih rendah. Operasi baca yang sangat konsisten harus masuk ke Wilayah primer tunggal.



Kadang-kadang diperlukan untuk mengubah Wilayah aktif dalam menanggapi kegagalan Regional, seperti yang dibahas nanti. Beberapa pengguna mengubah Wilayah yang sedang aktif pada jadwal reguler, seperti menerapkan follow-the-sunpenerapan. Ini menempatkan Wilayah aktif di dekat geografi yang memiliki aktivitas terbanyak (biasanya di mana siang hari, dengan demikian namanya), yang menghasilkan operasi baca dan tulis latensi terendah. Ini juga memiliki manfaat samping memanggil kode perubahan Wilayah setiap hari, dan memastikan bahwa itu diuji dengan baik sebelum pemulihan bencana.

Wilayah pasif mungkin menyimpan infrastruktur downscaled di sekitar DynamoDB yang akan dibangun hanya jika menjadi Wilayah aktif. Panduan ini tidak mencakup lampu pilot dan desain siaga

hangat. Untuk informasi lebih lanjut, Anda dapat membaca posting blog Disaster Recovery (DR) Arsitektur padaAWS, Bagian III: Pilot Light dan Warm Standby.

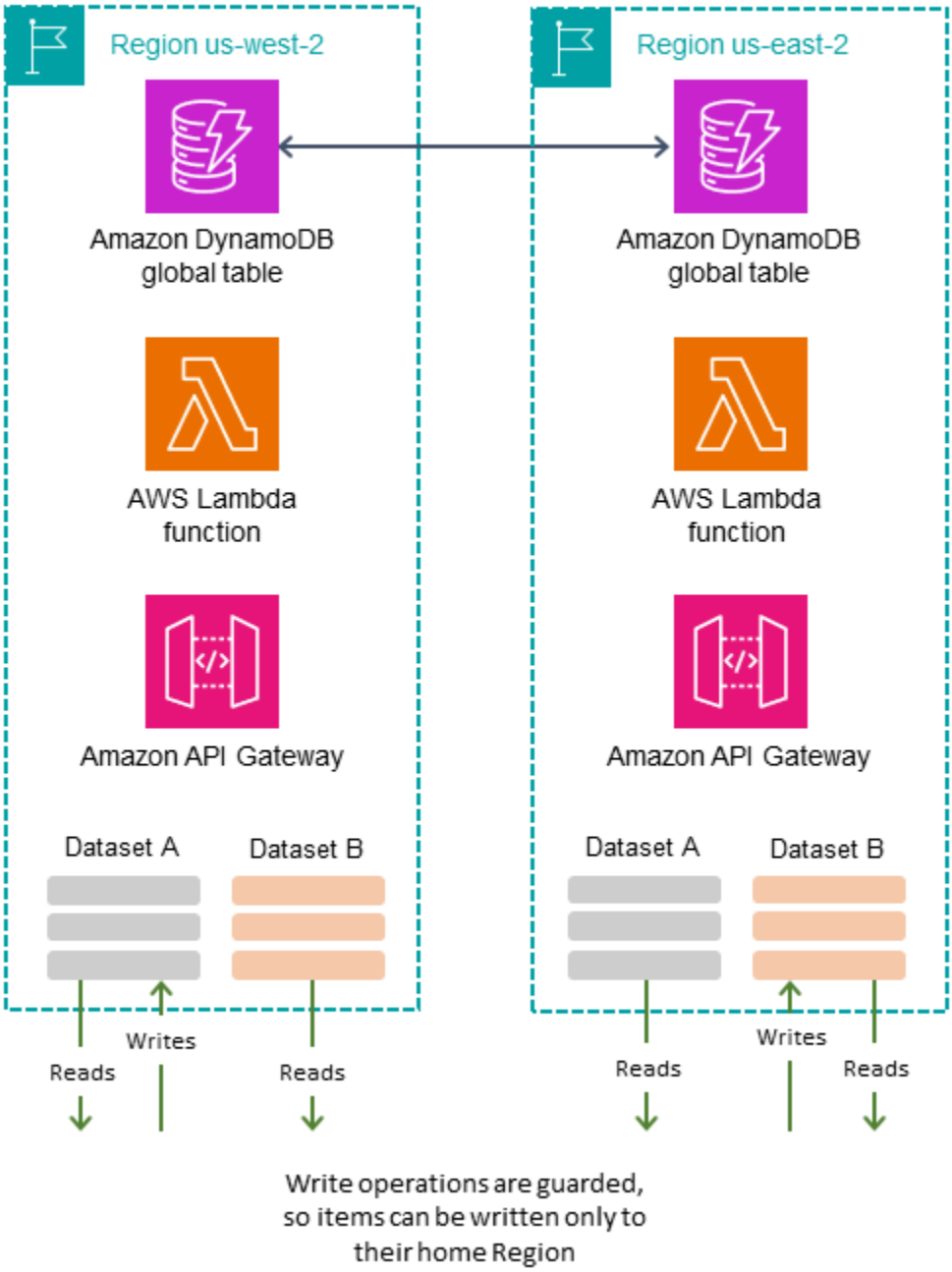
Menggunakan mode write to one Region berfungsi dengan baik saat Anda menggunakan tabel global untuk operasi baca dengan latensi rendah dan terdistribusi secara global. Contohnya adalah perusahaan media sosial besar yang perlu memiliki data referensi yang sama yang tersedia di setiap Wilayah di seluruh dunia. Mereka tidak sering memperbarui data, tetapi ketika mereka melakukannya, mereka hanya menulis ke satu Wilayah untuk menghindari potensi konflik penulisan. Operasi baca selalu diizinkan dari Wilayah mana pun.

Sebagai contoh lain, pertimbangkan perusahaan jasa keuangan yang dibahas sebelumnya yang menerapkan perhitungan cash-back harian. Mereka menggunakan write to any Region mode untuk menghitung saldo tetapi menulis ke satu mode Region untuk melacak pembayaran cash-back. Jika mereka ingin memberi hadiah sepeser pun untuk setiap $10 yang dihabiskan, mereka harus `Query` untuk semua transaksi dari hari sebelumnya, menghitung total yang dihabiskan, menulis keputusan uang kembali ke tabel baru, menghapus kumpulan item yang ditanyakan untuk menandainya sebagai dikonsumsi, dan menggantinya dengan item tunggal yang menyimpan sisa yang harus masuk ke perhitungan hari berikutnya. Pekerjaan ini membutuhkan transaksi, sehingga bekerja lebih baik dengan menulis ke satu mode Region. Sebuah aplikasi dapat mencampur mode tulis, bahkan pada tabel yang sama, selama beban kerja tidak memiliki peluang tumpang tindih.

Menulis ke mode Wilayah Anda (campuran primer)

Tulis ke mode tulis Wilayah Anda menetapkan subset data yang berbeda ke Wilayah rumah yang berbeda dan memungkinkan operasi penulisan ke item hanya melalui Wilayah asalnya. Mode ini aktif-pasif tetapi menetapkan Wilayah aktif berdasarkan item. Setiap Wilayah adalah yang utama untuk dataset yang tidak tumpang tindih, dan operasi penulisan harus dijaga untuk memastikan lokalitas yang tepat.

Mode ini mirip dengan menulis ke satu Wilayah kecuali bahwa ia memungkinkan operasi tulis latensi rendah, karena data yang terkait dengan setiap pengguna dapat ditempatkan di dekat jaringan yang lebih dekat dengan pengguna tersebut. Ini juga menyebarkan infrastruktur sekitarnya secara lebih merata antar Wilayah dan membutuhkan lebih sedikit pekerjaan untuk membangun infrastruktur selama skenario failover, karena semua Wilayah memiliki sebagian infrastruktur mereka yang sudah aktif.



Anda dapat menentukan Wilayah rumah untuk item dalam beberapa cara:

- Intrinsik: Beberapa aspek data, seperti atribut khusus atau nilai yang disematkan dalam kunci partisi, membuat Wilayah rumahnya jelas. Teknik ini dijelaskan dalam posting blog Gunakan Wilayah menyematkan untuk mengatur Wilayah rumah untuk item dalam tabel global Amazon DynamoDB.

- Negosiasi: Wilayah asal dari setiap kumpulan data dinegosiasikan dengan cara eksternal, seperti dengan layanan global terpisah yang mengelola tugas. Tugas mungkin memiliki durasi terbatas setelah itu tunduk pada negosiasi ulang.
- Berorientasi pada tabel: Alih-alih membuat tabel global yang mereplikasi tunggal, Anda membuat jumlah tabel global yang sama dengan mereplikasi Regions. Setiap nama tabel menunjukkan Wilayah asalnya. Dalam operasi standar, semua data ditulis ke Wilayah asal sementara Wilayah lain menyimpan salinan hanya-baca. Selama failover, Region lain sementara mengadopsi tugas tulis untuk tabel itu.

Misalnya, bayangkan Anda bekerja untuk perusahaan game. Anda memerlukan operasi baca dan tulis latensi rendah untuk semua gamer di seluruh dunia. Anda menetapkan setiap gamer ke Wilayah yang paling dekat dengan mereka. Wilayah itu mengambil semua operasi baca dan tulis mereka, memastikan read-after-write konsistensi yang kuat. Namun, ketika seorang gamer bepergian atau jika Wilayah asal mereka mengalami pemadaman, salinan lengkap data mereka tersedia di Wilayah alternatif, dan pemain dapat ditugaskan ke Wilayah rumah yang berbeda.

Sebagai contoh lain, bayangkan Anda bekerja di perusahaan konferensi video. Setiap metadata panggilan konferensi ditugaskan ke Wilayah tertentu. Pemanggil dapat menggunakan Wilayah yang paling dekat dengan mereka untuk latensi terendah. Jika terjadi pemadaman Wilayah, penggunaan tabel global memungkinkan pemulihan cepat karena sistem dapat memindahkan pemrosesan panggilan ke Wilayah yang berbeda di mana salinan data yang direplikasi sudah ada.

Strategi perutean untuk tabel global

Mungkin bagian paling rumit dari deployment tabel global adalah mengelola perutean permintaan. Permintaan harus dikirim dari pengguna akhir ke Wilayah yang dipilih terlebih dahulu, lalu dirutekan dengan cara tertentu. Permintaan menemukan beberapa tumpukan layanan di Wilayah itu, termasuk lapisan komputasi yang mungkin terdiri dari penyeimbang beban yang didukung oleh AWS Lambda fungsi, wadah, atau node Amazon Elastic Compute Cloud (AmazonEC2), dan mungkin layanan lain, termasuk mungkin database lain. Lapisan komputasi itu berkomunikasi dengan DynamoDB. Itu harus dilakukan dengan menggunakan titik akhir lokal untuk Wilayah itu. Data dalam tabel global direplikasi ke semua Wilayah lain yang berpartisipasi, dan setiap Wilayah memiliki tumpukan layanan serupa di sekitar tabel DynamoDB-nya.

Tabel global menyediakan salinan lokal dari data yang sama untuk setiap tumpukan di berbagai Wilayah. Sebaiknya Anda merancang tumpukan tunggal dalam satu Wilayah dan mengantisipasi melakukan panggilan jarak jauh ke titik akhir DynamoDB Wilayah sekunder jika tabel DynamoDB lokal mengalami masalah. Ini bukan praktik terbaik. Latensi yang terkait dengan lintas Wilayah mungkin 100 kali lebih tinggi dibandingkan akses lokal. back-and-forth Serangkaian 5 permintaan mungkin membutuhkan milidetik saat dilakukan secara lokal tetapi detik saat melintasi dunia. Sebaiknya rutekan pengguna akhir ke Wilayah lain untuk diproses. Untuk memastikan ketahanan, Anda memerlukan replikasi di beberapa Wilayah: replikasi lapisan komputasi serta lapisan data.

Ada banyak teknik untuk merutekan permintaan pengguna akhir ke Wilayah untuk diproses. Pilihan yang tepat tergantung pada mode tulis Anda dan pertimbangan failover Anda. Bagian ini membahas empat opsi: client-driven, compute-layer, Amazon Route 53, dan. AWS Global Accelerator

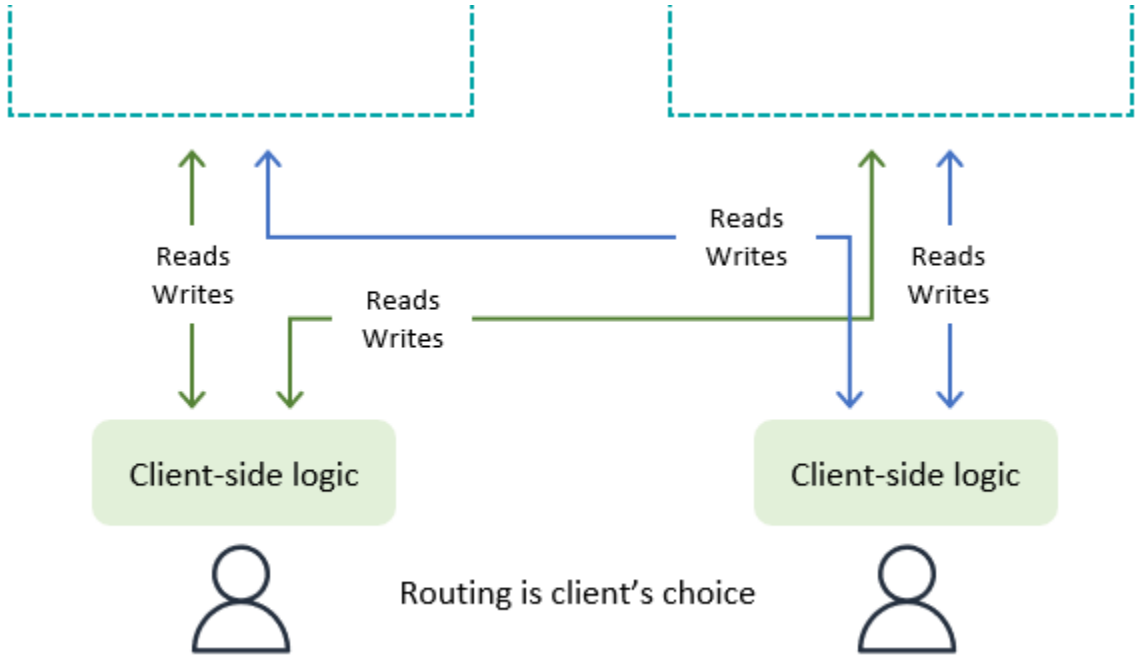
Topik

- Perutean permintaan berbasis klien
- Perutean permintaan lapisan komputasi
- Perutean permintaan Route 53
- Perutean permintaan Global Accelerator

Perutean permintaan berbasis klien

Dengan routing permintaan berbasis klien, klien pengguna akhir (aplikasi, halaman web dengan JavaScript, atau klien lain) melacak titik akhir aplikasi yang valid (misalnya, titik akhir Amazon API

Gateway daripada titik akhir DynamoDB literal) dan menggunakan logika tertanam sendiri untuk memilih Wilayah yang akan berkomunikasi. Ini mungkin memilih berdasarkan seleksi acak, latensi terendah yang diamati, pengukuran bandwidth tertinggi yang diamati, atau pemeriksaan kesehatan yang dilakukan secara lokal.



Sebagai keuntungan, routing permintaan berbasis klien dapat beradaptasi dengan hal-hal seperti kondisi lalu lintas internet publik dunia nyata untuk beralih Wilayah jika melihat kinerja yang menurun. Klien harus mengetahui semua titik akhir potensial, tetapi peluncuran titik akhir Regional baru bukanlah hal yang sering terjadi.

Dengan menulis ke mode Wilayah mana pun, klien dapat secara sepihak memilih titik akhir yang disukai. Jika aksesnya ke satu Wilayah terganggu, klien dapat merutekan ke titik akhir lain.

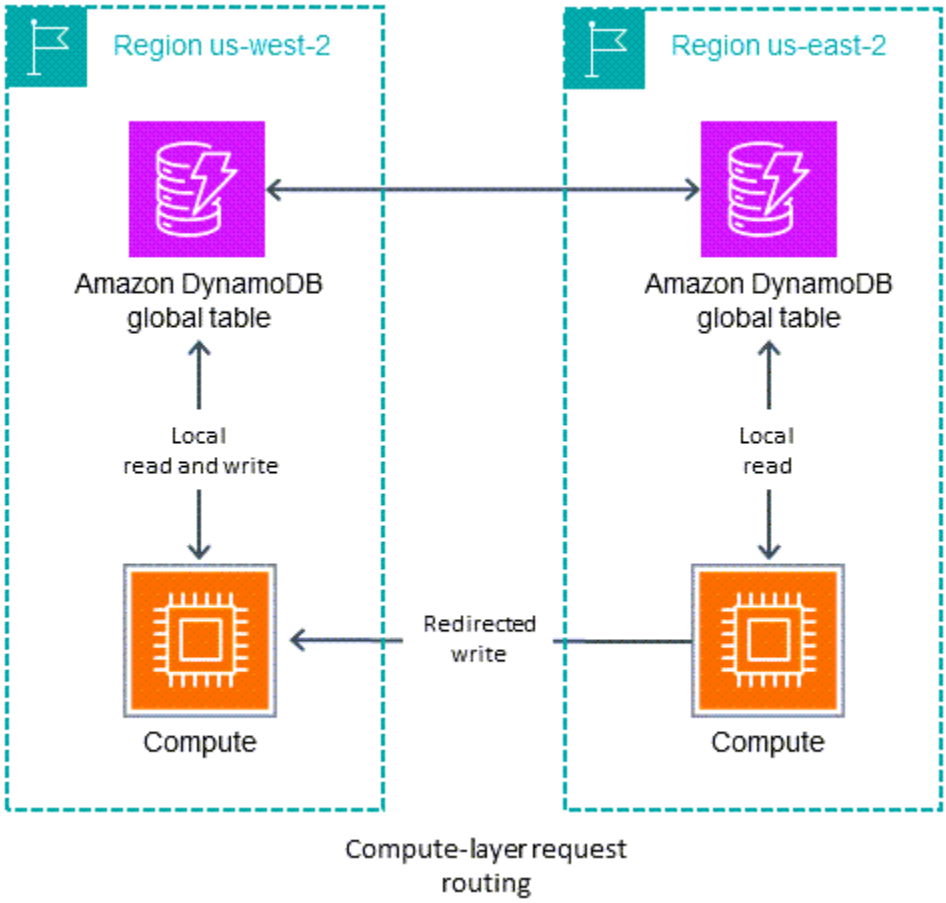
Dengan mode write to one Region, klien memerlukan mekanisme untuk merutekan permintaan penulisannya ke Region yang saat ini aktif. Ini bisa menjadi mekanisme dasar, seperti menguji secara empiris Wilayah mana yang saat ini menerima permintaan tulis (mencatat penolakan penulisan dan kembali ke alternatif). Atau dapat berupa mekanisme yang kompleks, seperti menggunakan koordinator global untuk menanyakan status aplikasi saat ini (mungkin dibangun di atas kontrol perutean Amazon Application Recovery Controller (ARCARC) (), yang menyediakan lima wilayah, sistem berbasis kuorum untuk mempertahankan status global untuk kebutuhan seperti ini). Klien dapat memutuskan apakah permintaan baca dapat masuk ke Wilayah mana pun untuk konsistensi akhirnya atau harus diarahkan ke Wilayah aktif untuk konsistensi yang kuat.

Dengan mode tulis ke Wilayah Anda, klien perlu menentukan Wilayah beranda untuk kumpulan data yang digunakannya. Misalnya, jika klien berhubungan dengan akun pengguna dan setiap akun pengguna di-homed ke Wilayah, klien dapat meminta penetapan titik akhir yang sesuai untuk digunakan dengan kredensialnya dari sistem login global.

Misalnya, perusahaan jasa keuangan yang membantu pengguna mengelola keuangan bisnis mereka melalui web menggunakan tabel global dengan menulis ke mode Wilayah Anda. Setiap pengguna harus masuk ke layanan pusat. Layanan tersebut mengembalikan kredensil serta titik akhir untuk Wilayah tempat kredensil tersebut akan berfungsi. Wilayah yang dikembalikan didasarkan pada tempat dataset pengguna saat ini di-homed. Kredensialnya valid untuk waktu yang singkat. Setelah itu, halaman web secara otomatis menegosiasikan login baru, yang memberikan kesempatan untuk berpotensi mengarahkan aktivitas pengguna ke Wilayah baru.

Perutean permintaan lapisan komputasi

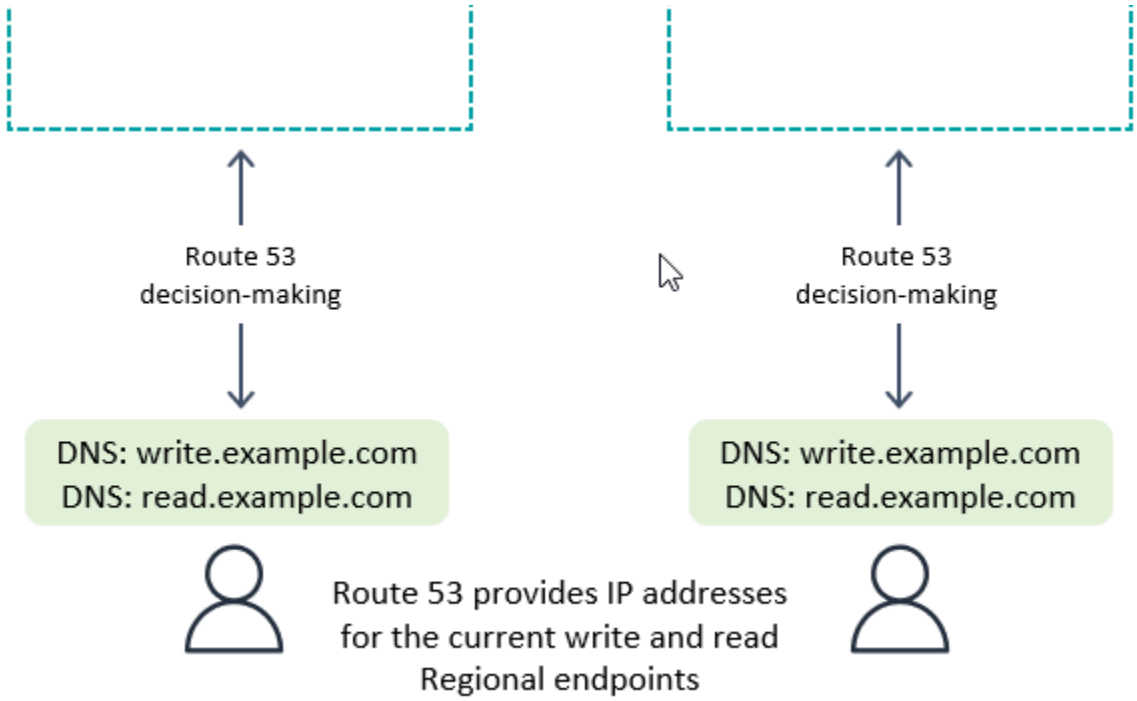
Dengan routing permintaan compute-layer, kode yang berjalan di lapisan komputasi menentukan apakah akan memproses permintaan secara lokal atau meneruskannya ke salinan dirinya sendiri yang berjalan di Wilayah lain. Saat Anda menggunakan mode tulis ke satu Wilayah, lapisan komputasi mungkin mendeteksi bahwa itu bukan Region aktif dan mengizinkan operasi baca lokal sambil meneruskan semua operasi tulis ke Wilayah lain. Kode lapisan komputasi ini harus mengetahui topologi data dan aturan perutean, dan menegakkannya dengan andal, berdasarkan pengaturan terbaru yang menentukan Wilayah mana yang aktif untuk data mana. Tumpukan perangkat lunak luar di dalam Wilayah tidak harus mengetahui bagaimana permintaan baca dan tulis dirutekan oleh layanan mikro. Dalam desain yang kuat, Wilayah penerima memvalidasi apakah Wilayah tersebut merupakan wilayah primer saat ini untuk operasi tulis. Jika tidak, maka akan muncul kesalahan yang menunjukkan bahwa status global perlu diperbaiki. Wilayah penerima mungkin juga melakukan buffering pada operasi tulis untuk sementara waktu jika Wilayah utama sedang dalam proses perubahan. Dalam semua kasus, tumpukan komputasi di suatu Wilayah hanya menulis ke titik akhir DynamoDB lokalnya, tetapi tumpukan komputasi tersebut mungkin berkomunikasi satu sama lain.



Vanguard Group menggunakan sistem yang disebut Global Orchestration and Status Tool (GOaST) dan perpustakaan bernama Global Multi-Region library (GMRlib) untuk proses routing ini, seperti yang disajikan di re:Invent 2022. Mereka menggunakan model primer follow-the-sun tunggal. GOaSTmempertahankan keadaan global, mirip dengan kontrol ARC routing yang dibahas di bagian sebelumnya. Ini menggunakan tabel global untuk melacak Wilayah mana yang merupakan Wilayah utama dan kapan sakelar utama berikutnya dijadwalkan. Semua operasi baca dan tulis dilaluiGMRlib, yang berkoordinasi denganGOaST. GMRlibmemungkinkan operasi baca dilakukan secara lokal, pada latensi rendah. Untuk operasi tulis, GMRlib periksa apakah Wilayah lokal adalah Wilayah utama saat ini. Jika ya, operasi tulis selesai secara langsung. Jika tidak, GMRlib teruskan tugas menulis ke GMRlib dalam Wilayah utama. Pustaka penerima tersebut mengonfirmasi bahwa ia juga menganggap dirinya sebagai Wilayah utama dan menimbulkan kesalahan jika bukan, yang menunjukkan penundaan penyebaran dengan keadaan global. Pendekatan ini memberikan manfaat validasi dengan tidak menulis langsung ke titik akhir DynamoDB jarak jauh.

Perutean permintaan Route 53

Amazon Route 53 adalah teknologi Domain Name Service (DNS). Dengan Route 53, klien meminta titik akhir dengan mencari nama DNS domain yang terkenal, dan Route 53 mengembalikan alamat IP yang sesuai dengan titik akhir Regional yang ditentukannya paling tepat. Route 53 memiliki daftar panjang kebijakan perutean yang digunakan untuk menentukan Wilayah yang sesuai. Itu juga dapat melakukan routing failover untuk merutekan lalu lintas dari Wilayah yang gagal pemeriksaan kesehatan.



Dengan menulis ke mode Wilayah mana pun, atau jika digabungkan dengan perutean permintaan lapisan komputasi di backend, Route 53 dapat diberikan kebebasan penuh untuk mengembalikan Wilayah berdasarkan aturan internal yang kompleks, seperti memilih Wilayah di jaringan terdekat atau kedekatan geografis, atau pilihan lainnya.

Dengan menulis ke satu mode Wilayah, Anda dapat mengonfigurasi Route 53 untuk mengembalikan Wilayah yang saat ini aktif (dengan menggunakanARC). Jika klien ingin terhubung ke Wilayah pasif (misalnya, untuk operasi baca), itu bisa mencari DNS nama yang berbeda.

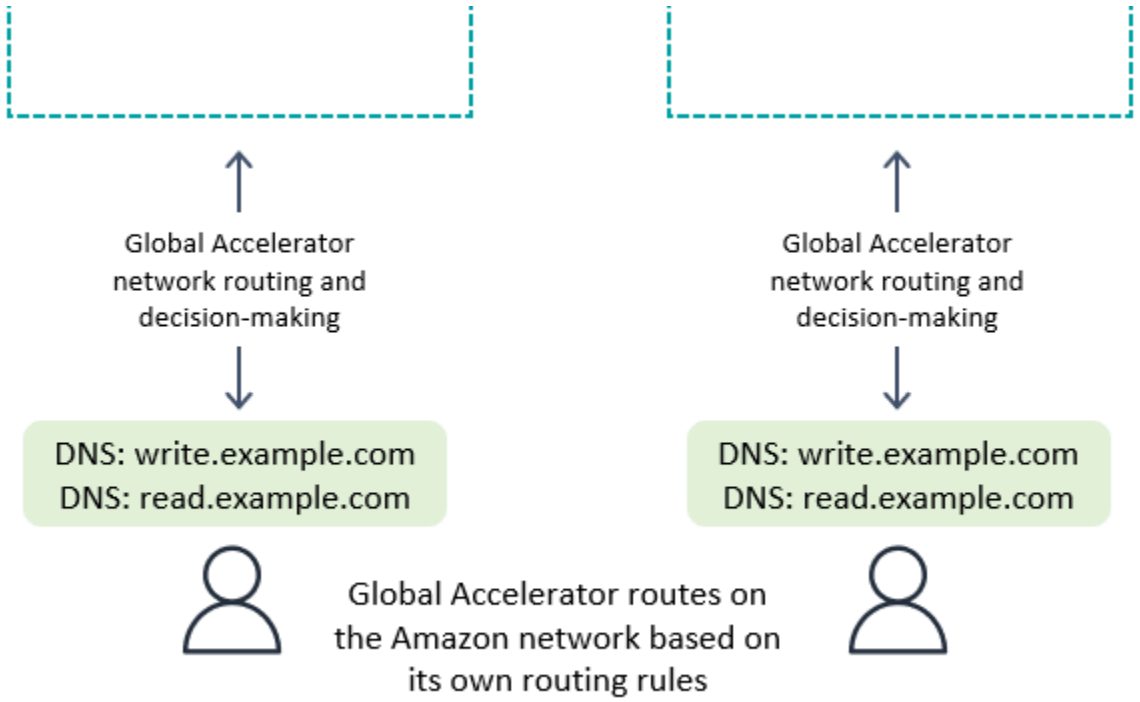
> **Note**
>
> Klien menyimpan alamat IP dalam respons dari Route 53 untuk waktu yang ditunjukkan oleh pengaturan time to live (TTL) pada nama domain. Yang lebih lama TTL memperpanjang

tujuan waktu pemulihan (RTO) bagi semua klien untuk mengenali titik akhir yang baru. Nilai 60 detik adalah tipikal untuk penggunaan failover. Tidak semua perangkat lunak dengan sempurna mematuhi DNS TTL kedaluwarsa, dan mungkin ada beberapa tingkat DNS caching, seperti pada sistem operasi, mesin virtual, dan aplikasi.

Dengan menulis ke mode Wilayah Anda, sebaiknya hindari Route 53 kecuali Anda juga menggunakan perutean permintaan lapisan komputasi.

Perutean permintaan Global Accelerator

AWS Global AcceleratorDengan klien mencari nama domain terkenal di Route 53. Namun, alih-alih mendapatkan kembali alamat IP yang sesuai dengan titik akhir Regional, klien mendapatkan kembali alamat IP statis anycast yang merutekan ke lokasi AWS tepi terdekat. Mulai dari lokasi tepi itu, semua lalu lintas dirutekan pada AWS jaringan pribadi ke beberapa titik akhir (Network Load Balancers, Application Load Balancers, EC2 instance, atau alamat IP elastis) di Wilayah yang dipilih oleh aturan routing yang dipertahankan dalam Global Accelerator. Dibandingkan dengan perutean berdasarkan aturan Route 53, perutean permintaan Global Accelerator memiliki latensi yang lebih rendah karena mengurangi jumlah lalu lintas di internet publik. Selain itu, karena Global Accelerator tidak bergantung pada DNS TTL kedaluwarsa untuk mengubah aturan perutean, ia dapat menyesuaikan perutean lebih cepat.



Dengan menulis ke mode Wilayah apa pun, atau jika dikombinasikan dengan perutean permintaan lapisan komputasi di backend, Global Accelerator bekerja dengan mulus. Klien terhubung ke lokasi tepi terdekat dan tidak perlu khawatir tentang Wilayah mana yang menerima permintaan.

Dengan mode write to one Region, aturan perutean Global Accelerator harus mengirim permintaan ke Region yang sedang aktif. Anda dapat menggunakan pemeriksaan kondisi yang secara artifisial melaporkan kegagalan pada Wilayah yang tidak diperhitungkan sebagai Wilayah aktif oleh sistem global Anda. Seperti halnyaDNS, dimungkinkan untuk menggunakan nama DNS domain alternatif untuk merutekan permintaan baca, jika permintaan dapat berasal dari Wilayah mana pun.

Dengan menulis ke mode Wilayah Anda, sebaiknya hindari Global Accelerator kecuali Anda juga menggunakan perutean permintaan lapisan komputasi.

Proses evakuasi untuk tabel global

Mengevakuasi suatu Wilayah adalah proses migrasi aktivitas—biasanya aktivitas menulis, mungkin aktivitas membaca — jauh dari Wilayah itu.

Mengevakuasi Wilayah langsung

Anda dapat memutuskan untuk mengevakuasi Wilayah langsung karena sejumlah alasan: sebagai bagian dari aktivitas bisnis biasa (misalnya, jika Anda menggunakan, tulis ke satu mode Wilayah)follow-the-sun, karena keputusan bisnis untuk mengubah Wilayah yang saat ini aktif, sebagai respons terhadap kegagalan dalam tumpukan perangkat lunak di luar DynamoDB, atau karena Anda menghadapi masalah umum seperti latensi yang lebih tinggi dari biasanya di Wilayah.

Dengan menulis ke mode Wilayah apa pun, mengevakuasi Wilayah langsung sangatlah mudah. Anda dapat mengarahkan lalu lintas ke Wilayah alternatif dengan menggunakan sistem perutean apa pun, dan membiarkan operasi penulisan yang telah terjadi di Wilayah yang dievakuasi mereplikasi seperti biasa.

Dengan menulis ke satu Wilayah dan menulis ke mode Wilayah Anda, Anda harus memastikan bahwa semua operasi penulisan ke Wilayah aktif telah direkam sepenuhnya, diproses streaming, dan disebarkan secara global sebelum memulai operasi penulisan di Wilayah aktif yang baru, untuk memastikan bahwa operasi penulisan di future diproses terhadap versi data terbaru.

Katakanlah bahwa Wilayah A aktif dan Wilayah B pasif (baik untuk tabel penuh atau untuk item yang homed di Wilayah A). Mekanisme khas untuk melakukan evakuasi adalah dengan menghentikan operasi penulisan ke A, menunggu cukup lama hingga operasi tersebut disebarkan sepenuhnya ke B, memperbarui tumpukan arsitektur untuk mengenali B sebagai aktif, dan kemudian melanjutkan operasi tulis ke B. tidak ada metrik untuk menunjukkan dengan kepastian mutlak bahwa Wilayah A telah sepenuhnya mereplikasi datanya ke Wilayah B. jika Wilayah A sehat, menghentikan operasi tulis ke Wilayah A dan menunggu 10 kali nilai maksimum `ReplicationLatency` metrik baru-baru ini biasanya akan cukup untuk menentukan bahwa replikasi selesai. Jika Wilayah A tidak sehat dan menunjukkan area latensi lain yang meningkat, Anda akan memilih kelipatan yang lebih besar untuk waktu tunggu.

Mengevakuasi offline

Ada kasus khusus yang perlu dipertimbangkan: Bagaimana jika Wilayah A sepenuhnya offline tanpa pemberitahuan? Ini sangat tidak mungkin tetapi harus dipertimbangkan. Jika ini terjadi, setiap operasi

penulisan di Wilayah A yang belum disebarkan akan diadakan dan disebarkan setelah Wilayah A kembali online. Operasi penulisan tidak hilang, tetapi propagasi mereka tertunda tanpa batas waktu.

Cara melanjutkan dalam acara ini adalah keputusan aplikasi. Untuk kelangsungan bisnis, operasi tulis mungkin perlu melanjutkan ke Wilayah B. primer baru Namun, jika item di Wilayah B menerima pembaruan sementara ada propagasi yang tertunda dari operasi tulis untuk item tersebut dari Wilayah A, propagasi ditekan di bawah model win penulis terakhir. Pembaruan apa pun di Wilayah B mungkin menekan permintaan penulisan yang masuk.

Dengan menulis ke mode Wilayah apa pun, operasi baca dan tulis dapat berlanjut di Wilayah B, percaya bahwa item di Wilayah A akan menyebar ke Wilayah B pada akhirnya dan mengenali potensi item yang hilang hingga Wilayah A kembali online. Bila memungkinkan, seperti dengan operasi tulis idempoten, Anda harus mempertimbangkan memutar ulang lalu lintas tulis baru-baru ini (misalnya, dengan menggunakan sumber peristiwa hulu) untuk mengisi celah dari setiap operasi tulis yang berpotensi hilang dan membiarkan penulis terakhir memenangkan resolusi konflik menekan propagasi akhirnya operasi tulis yang masuk.

Dengan mode tulis lainnya, Anda harus mempertimbangkan sejauh mana pekerjaan dapat dilanjutkan dengan sedikit out-of-date pandangan dunia. Beberapa durasi operasi tulis yang kecil, seperti yang dilacak oleh`ReplicationLatency`, akan hilang sampai Wilayah A kembali online. Bisakah bisnis bergerak maju? Dalam beberapa kasus penggunaan itu bisa, tetapi dalam kasus lain mungkin tidak tanpa mekanisme mitigasi tambahan.

Misalnya, bayangkan Anda harus menjaga saldo kredit yang tersedia tanpa gangguan bahkan setelah pemadaman penuh suatu Wilayah. Anda dapat membagi saldo menjadi dua item yang berbeda, satu homed di Wilayah A dan satu di Wilayah B, dan mulai masing-masing dengan setengah saldo yang tersedia. Ini akan menggunakan write ke mode Region Anda. Pembaruan transaksional yang diproses di setiap Wilayah akan menuliskan salinan saldo lokal. Jika Wilayah A berjalan sepenuhnya offline, pekerjaan masih dapat dilanjutkan dengan pemrosesan transaksi di Wilayah B, dan operasi tulis akan terbatas pada bagian saldo yang disimpan di Wilayah B. Memisahkan saldo seperti ini memperkenalkan kompleksitas ketika saldo menjadi rendah atau kredit harus diseimbangkan kembali, tetapi itu memberikan satu contoh pemulihan bisnis yang aman bahkan dengan operasi penulisan tertunda yang tidak pasti.

Sebagai contoh lain, bayangkan bahwa Anda menangkap data formulir web. Anda dapat menggunakan kontrol konkurensi optimis (OCC) untuk menetapkan versi ke item data dan menyematkan versi terbaru ke dalam formulir web sebagai bidang tersembunyi. Pada setiap kirimkan, operasi tulis hanya berhasil jika versi dalam database masih cocok dengan versi yang dibuat oleh formulir. Jika versi tidak cocok, formulir web dapat di-refresh (atau digabungkan dengan

hati-hati) berdasarkan versi saat ini dalam database, dan pengguna dapat melanjutkan lagi. Model OCC biasanya melindungi terhadap klien lain menimpa dan menghasilkan versi baru dari data, tetapi juga dapat membantu selama failover di mana klien mungkin menemukan versi data yang lebih lama. Mari kita bayangkan bahwa Anda menggunakan stempel waktu sebagai versi. Formulir ini pertama kali dibangun terhadap Wilayah A pada pukul 12:00 tetapi (setelah failover) mencoba menulis ke Wilayah B dan pemberitahuan bahwa versi terbaru dalam database adalah 11:59. Dalam skenario ini, klien dapat menunggu versi 12:00 untuk menyebarkan ke Wilayah B dan kemudian menulis di atas versi itu, atau membangun pada 11:59 dan membuat versi 12:01 baru (yang, setelah menulis, akan menekan versi masuk setelah Region A pulih).

Sebagai contoh ketiga, perusahaan jasa keuangan menyimpan data tentang akun pelanggan dan transaksi keuangan mereka dalam database DynamoDB. Jika terjadi pemadaman Wilayah A yang lengkap, mereka ingin memastikan bahwa aktivitas penulisan apa pun yang terkait dengan akun mereka sepenuhnya tersedia di Wilayah B, atau mereka ingin mengkarantina akun mereka sebagai sebagian yang diketahui sampai Wilayah A kembali online. Alih-alih menghentikan semua bisnis, mereka memutuskan untuk menghentikan bisnis hanya untuk sebagian kecil dari akun yang mereka tentukan memiliki transaksi yang tidak disebarkan. Untuk mencapai hal ini, mereka menggunakan Wilayah ketiga, yang akan kita sebut Wilayah C. Sebelum mereka memproses operasi tulis apa pun di Wilayah A, mereka menempatkan ringkasan singkat dari operasi yang tertunda (misalnya, jumlah transaksi baru untuk akun) di Wilayah C. ringkasan ini cukup untuk Wilayah B untuk menentukan apakah pandangannya sepenuhnya mutakhir. Tindakan ini secara efektif mengunci akun sejak penulisan di Wilayah C hingga Wilayah A menerima operasi penulisan dan Wilayah B menerimanya. Data di Wilayah C tidak digunakan kecuali sebagai bagian dari proses failover, setelah itu Wilayah B dapat memeriksa ulang datanya dengan Wilayah C untuk memeriksa apakah ada akunnya yang kedaluwarsa. Akun-akun tersebut akan ditandai sebagai karantina sampai pemulihan Wilayah A menyebarkan data sebagian ke Wilayah B. Jika Wilayah C gagal, Wilayah D baru dapat dipintal untuk digunakan sebagai gantinya. Data di Wilayah C sangat sementara, dan setelah beberapa menit Wilayah D akan memiliki up-to-date catatan yang cukup tentang operasi penulisan dalam penerbangan agar sepenuhnya berguna. Jika Wilayah B gagal, Wilayah A dapat terus menerima permintaan tulis bekerjasama dengan Wilayah C. perusahaan ini bersedia menerima penulisan latensi yang lebih tinggi (ke dua Wilayah: C dan kemudian A) dan beruntung memiliki model data di mana keadaan akun dapat diringkas secara ringkas.

Perencanaan kapasitas throughput untuk tabel global

Migrasi lalu lintas dari satu Wilayah ke Wilayah lain memerlukan pertimbangan cermat terhadap pengaturan tabel DynamoDB terkait kapasitas.

Berikut adalah beberapa pertimbangan untuk mengelola kapasitas menulis:

- Tabel global harus berada dalam mode sesuai permintaan atau disediakan dengan penskalaan otomatis yang diaktifkan.
- Jika disediakan dengan penskalaan otomatis, pengaturan tulis (pemanfaatan minimum, maksimum, dan target) direplikasi di seluruh Wilayah. Meskipun pengaturan penskalaan otomatis disinkronkan, kapasitas tulis aktual yang disediakan mungkin mengambang secara independen di antara Wilayah.
- Salah satu alasan Anda mungkin melihat kapasitas penulisan yang disediakan berbeda adalah karena fitur time to live (TTL). Saat Anda mengaktifkan TTL di DynamoDB, Anda dapat menentukan nama atribut yang nilainya menunjukkan waktu kedaluwarsa item tersebut, dalam format waktu epoch Unix dalam hitungan detik. Setelah itu, DynamoDB dapat menghapus item tanpa menimbulkan biaya tulis. Dengan tabel global, Anda dapat mengonfigurasi TTL di Wilayah mana pun, dan pengaturan secara otomatis direplikasi ke Wilayah lain yang terkait dengan tabel global. Ketika item memenuhi syarat untuk dihapus melalui TTL aturan, pekerjaan itu dapat dilakukan di Wilayah mana pun. Operasi penghapusan dilakukan tanpa menggunakan unit tulis pada tabel sumber, tetapi tabel replika akan mendapatkan penulisan yang direplikasi dari operasi penghapusan itu dan akan menimbulkan biaya unit tulis yang direplikasi.
- Jika Anda menggunakan penskalaan otomatis, pastikan pengaturan kapasitas tulis maksimum yang disediakan cukup tinggi untuk menangani semua operasi penulisan serta semua operasi penghapusan potensialTTL. Penskalaan otomatis menyesuaikan setiap Wilayah sesuai dengan konsumsi tulisnya. Tabel sesuai permintaan tidak memiliki pengaturan kapasitas tulis maksimum yang disediakan, tetapi batas throughput tulis maksimum tingkat tabel menentukan kapasitas tulis berkelanjutan maksimum yang diperbolehkan oleh tabel sesuai permintaan. Batas defaultnya adalah 40.000, tetapi dapat disesuaikan. Kami menyarankan Anda mengaturnya cukup tinggi untuk menangani semua operasi tulis (termasuk operasi TTL tulis) yang mungkin dibutuhkan tabel sesuai permintaan. Nilai ini harus sama di seluruh Wilayah yang berpartisipasi saat Anda menyiapkan tabel global.

Berikut adalah beberapa pertimbangan untuk mengelola kapasitas baca:

- Pengaturan manajemen kapasitas baca boleh berbeda di antara Wilayah karena diasumsikan bahwa Wilayah yang berbeda mungkin memiliki pola baca yang independen. Saat Anda pertama kali menambahkan replika global ke tabel, kapasitas Wilayah sumber disebarkan. Setelah pembuatan, Anda dapat menyesuaikan pengaturan kapasitas baca, yang tidak ditransfer ke sisi lain.
- Saat Anda menggunakan penskalaan otomatis DynamoDB, pastikan pengaturan kapasitas baca maksimum yang disediakan cukup tinggi untuk menangani semua operasi baca di seluruh Wilayah. Selama operasi standar, kapasitas baca mungkin akan disebarkan di seluruh Wilayah, tetapi tabel selama failover harus mampu beradaptasi secara otomatis dengan peningkatan beban kerja baca. Tabel sesuai permintaan tidak memiliki pengaturan kapasitas baca maksimum yang disediakan, tetapi batas throughput baca maksimum tingkat tabel menentukan kapasitas baca berkelanjutan maksimum yang diperbolehkan oleh tabel sesuai permintaan. Batas defaultnya adalah 40.000, tetapi dapat disesuaikan. Sebaiknya Anda mengaturnya cukup tinggi untuk menangani semua operasi baca yang mungkin diperlukan tabel jika semua operasi baca dirutekan ke Wilayah tunggal ini.
- Jika tabel di satu Wilayah biasanya tidak menerima lalu lintas baca tetapi mungkin harus menyerap sejumlah besar lalu lintas baca setelah failover, Anda dapat meningkatkan kapasitas baca tabel yang disediakan, menunggu tabel selesai diperbarui, lalu menyediakan tabelnya lagi. Anda dapat membiarkan tabel dalam mode yang disediakan atau mengalihkannya ke mode sesuai permintaan. Ini akan menghangatkan tabel untuk menerima tingkat lalu lintas baca yang lebih tinggi.

ARCmemiliki pemeriksaan kesiapan yang dapat berguna dalam mengonfirmasi bahwa Wilayah DynamoDB memiliki pengaturan tabel dan kuota akun yang serupa, apakah Anda menggunakan Route 53 untuk merutekan permintaan atau tidak. Pemeriksaan kesiapan ini juga membantu Anda menyesuaikan kuota tingkat akun agar cocok.

Daftar periksa persiapan untuk tabel global

Gunakan daftar periksa berikut untuk keputusan dan tugas saat Anda men-deploy tabel global.

- Tentukan jumlah dan Wilayah yang akan berpartisipasi dalam tabel global.
- Tentukan mode tulis aplikasi Anda.
- Rencanakan strategi routing Anda, berdasarkan mode tulis Anda.
- Tentukan rencana evakuasi Anda, berdasarkan mode tulis dan strategi perutean Anda.
- Tangkap metrik tentang kesehatan, latensi, dan kesalahan di setiap Wilayah. Untuk daftar metrik DynamoDB, lihat AWS posting blog Memantau Amazon DynamoDB untuk kesadaran operasional. Anda juga harus menggunakan kenari sintetis (permintaan buatan yang dirancang untuk mendeteksi kegagalan) serta pengamatan langsung lalu lintas pelanggan. Tidak semua masalah muncul di metrik DynamoDB.
- Atur alarm untuk peningkatan `ReplicationLatency` yang berkelanjutan. Peningkatan mungkin menunjukkan kesalahan konfigurasi yang tidak disengaja, yaitu tabel global memiliki pengaturan tulis berbeda di Wilayah berbeda, yang menyebabkan kegagalan permintaan yang direplikasi dan peningkatan latensi. Hal ini juga dapat mengindikasikan adanya gangguan regional. Contoh yang baik adalah menghasilkan peringatan jika rata-rata terkini melebihi 180.000 milidetik. Anda mungkin juga memperhatikan `ReplicationLatency` penurunan ke 0, yang menunjukkan replikasi terhenti.
- Tetapkan pengaturan baca dan tulis maksimum yang memadai untuk setiap tabel global.
- Identifikasi kondisi di mana Anda akan mengevakuasi suatu Wilayah. Jika keputusan melibatkan penilaian manusia, dokumentasikan semua pertimbangannya. Pekerjaan ini harus dilakukan dengan hati-hati sebelumnya, bukan di bawah tekanan.
- Simpan runbook untuk setiap tindakan yang harus dilakukan saat Anda mengevakuasi suatu Wilayah. Biasanya pekerjaan yang dilakukan untuk tabel global sangat sedikit, tetapi memindahkan sisa tumpukan dapat menjadi pekerjaan yang rumit.

> **Note**
>
> Dengan prosedur failover, praktik terbaik adalah hanya mengandalkan operasi pesawat data dan bukan pada operasi pesawat kontrol, karena beberapa operasi pesawat kontrol mungkin terdegradasi selama kegagalan Wilayah. Untuk informasi selengkapnya, lihat

posting AWS blog Membangun aplikasi tangguh dengan tabel global Amazon DynamoDB: Bagian 4.

- Uji semua aspek runbook secara berkala, termasuk evakuasi Wilayah. Runbook yang belum teruji adalah runbook yang tidak dapat diandalkan.
- Pertimbangkan AWS Resilience Hubuntuk menggunakan untuk mengevaluasi ketahanan seluruh aplikasi Anda (termasuk tabel global). Layanan ini memberikan pandangan komprehensif tentang status ketahanan portofolio aplikasi Anda melalui dasbornya.
- Pertimbangkan untuk menggunakan pemeriksaan ARCkesiapan untuk mengevaluasi konfigurasi aplikasi Anda saat ini dan melacak penyimpangan dari praktik terbaik.
- Saat Anda menulis pemeriksaan kesehatan untuk digunakan dengan Route 53 atau Global Accelerator, buat serangkaian panggilan yang mencakup alur database lengkap. Jika Anda membatasi pemeriksaan untuk mengonfirmasi hanya bahwa titik akhir DynamoDB sudah aktif, Anda tidak akan dapat mencakup banyak mode kegagalan AWS Identity and Access Management seperti IAM () kesalahan konfigurasi, masalah penerapan kode, kegagalan dalam tumpukan di luar DynamoDB, latensi baca atau tulis yang lebih tinggi dari rata-rata, dan sebagainya.

Tabel global FAQ

Bagian ini memberikan jawaban atas pertanyaan umum tentang tabel global DynamoDB.

Berapa harga untuk tabel global?

- Operasi tulis dalam tabel DynamoDB tradisional dihargai dalam unit kapasitas tulis (WCU) untuk tabel yang disediakan atau unit permintaan tulis (WRU) untuk tabel sesuai permintaan. Jika Anda menulis item 5 KB, itu dikenakan biaya 5 unit. Tulis ke tabel global diberi harga dalam unit kapasitas tulis yang direplikasi (RWCU) untuk tabel yang disediakan atau unit permintaan tulis yang direplikasi (RWRU) untuk tabel sesuai permintaan.
- RWCU dan RWRU termasuk biaya infrastruktur streaming yang diperlukan untuk mengelola replikasi. Dengan demikian, mereka dihargai 50 persen lebih tinggi dari WCU dan WRU. Biaya transfer data Lintas Wilayah berlaku.
- Biaya RWCu dan RwRU dikenakan di setiap Wilayah di mana item ditulis secara langsung atau ditulis melalui replikasi.
- Menulis ke indeks sekunder global (GSI) dianggap sebagai operasi penulisan lokal dan menggunakan unit tulis reguler.
- Tidak ada kapasitas cadangan yang tersedia untuk RWCU saat ini. Membeli kapasitas cadangan untuk WCU mungkin masih bermanfaat untuk tabel di mana GSI mengkonsumsi unit tulis.
- Saat Anda menambahkan Wilayah baru ke tabel global, DynamoDB bootstrap Wilayah baru secara otomatis dan menagih Anda seolah-olah itu adalah pemulihan tabel, berdasarkan ukuran GB tabel. Ini juga membebankan biaya transfer data lintas wilayah.

Wilayah mana yang didukung tabel global?

Tabel global mendukung semuaWilayah AWS.

Bagaimana GSI ditangani dengan tabel global?

Dalam tabel global (saat ini, versi 2019), saat Anda membuat GSI di satu Wilayah, secara otomatis dibuat di Wilayah lain yang berpartisipasi dan secara otomatis diisi ulang.

Bagaimana cara menghentikan replikasi tabel global?

Anda dapat menghapus tabel replika seperti Anda menghapus tabel lainnya. Menghapus tabel global akan menghentikan replikasi ke Wilayah tersebut dan menghapus salinan tabel yang disimpan di Wilayah tersebut. Namun, Anda tidak dapat menghentikan replikasi sambil menyimpan salinan tabel sebagai entitas independen, dan Anda juga tidak dapat menjeda replikasi.

Bagaimana Amazon DynamoDB Streams berinteraksi dengan tabel global?

Setiap tabel global menghasilkan aliran independen berdasarkan semua operasi penulisannya, dari mana pun mereka memulai. Anda dapat menggunakan aliran DynamoDB di satu Wilayah atau di semua Wilayah (secara independen). Jika ingin memproses operasi tulis lokal tetapi tidak direplikasi, Anda dapat menambahkan atribut Wilayah Anda sendiri ke setiap item untuk mengidentifikasi Wilayah penulisan. Anda kemudian dapat menggunakan filter AWS Lambda peristiwa untuk memanggil fungsi Lambda hanya untuk operasi tulis di Wilayah lokal. Ini membantu dengan menyisipkan dan memperbarui operasi, tetapi tidak menghapus operasi.

Bagaimana tabel global menangani transaksi?

Operasi transaksional memberikan jaminan atomisitas, konsistensi, isolasi, daya tahan (ACID) hanya di Wilayah tempat operasi penulisan awalnya terjadi. Transaksi tidak didukung di seluruh Wilayah dalam tabel global. Misalnya, jika Anda memiliki tabel global dengan replika di Wilayah AS Timur (Ohio) dan AS Barat (Oregon) serta melakukan operasi `TransactWriteItems` di Wilayah AS Timur (Ohio), Anda mungkin melihat transaksi yang selesai sebagian di Wilayah AS Barat (Oregon) seiring perubahan direplikasi. Perubahan direplikasi ke Wilayah lain hanya setelah perubahan tersebut dibuat di Wilayah sumber.

Bagaimana tabel global berinteraksi dengan cache DynamoDB Accelerator (DAX)?

Tabel global melewati DAX dengan memperbarui DynamoDB secara langsung, sehingga DAX tidak mengetahui jika menyimpan data yang sudah usang. Cache DAX disegarkan hanya ketika TTL cache kedaluwarsa.

Apakah tag pada tabel disebarkan?

Tidak, tag tidak disebarkan secara otomatis.

Haruskah saya membuat cadangan tabel di semua Wilayah atau hanya satu?

Jawabannya tergantung pada tujuan pencadangan.

- Jika Anda ingin memastikan ketahanan data, DynamoDB sudah menyediakan perlindungan itu. Layanan ini memastikan ketahanan.
- Jika Anda ingin menyimpan snapshot untuk catatan historis (misalnya, untuk memenuhi persyaratan peraturan), membuat cadangan di satu Wilayah sudah cukup. Anda dapat menyalin cadangan ke Wilayah tambahan menggunakan AWS Backup.
- Jika Anda ingin memulihkan data yang dihapus atau dimodifikasi secara keliru, gunakan DynamoDB point-in-time recovery (PITR) di satu Wilayah.

Bagaimana cara menerapkan tabel global dengan menggunakan? AWS CloudFormation

- CloudFormation merupakan tabel DynamoDB dan tabel global sebagai dua sumber daya terpisah: dan. `AWS::DynamoDB::Table` `AWS::DynamoDB::GlobalTable` Salah satu pendekatannya adalah membuat semua tabel yang berpotensi bersifat global dengan menggunakan `GlobalTable` konstruksi, menyimpannya sebagai tabel mandiri pada awalnya, dan menambahkan Wilayah nanti, jika perlu.
- Dalam CloudFormation, setiap tabel global dikendalikan oleh satu tumpukan, dalam satu Wilayah, terlepas dari jumlah replika. Saat Anda menerapkan template Anda, CloudFormation membuat dan memperbarui semua replika sebagai bagian dari operasi tumpukan tunggal. Jangan men-deploy sumber daya AWS::DynamoDB::GlobalTable yang sama di beberapa Wilayah. Tindakan tersebut tidak didukung dan akan mengakibatkan kesalahan. Jika men-deploy templat aplikasi di beberapa Wilayah, Anda dapat menggunakan ketentuan untuk membuat sumber daya `AWS::DynamoDB::GlobalTable` di satu Wilayah. Alternatifnya, Anda dapat memilih untuk menentukan sumber daya `AWS::DynamoDB::GlobalTable` Anda dalam tumpukan yang terpisah dari tumpukan aplikasi Anda, dan memastikan bahwa sumber daya tersebut disebarkan ke satu Wilayah.

- Jika Anda memiliki tabel reguler dan ingin mengonversinya menjadi tabel global sambil menjaganya tetap dikelola oleh CloudFormation: Setel kebijakan penghapusan ke`Retain`, hapus tabel dari tumpukan, ubah tabel menjadi tabel global di konsol, lalu impor tabel global sebagai sumber daya baru ke tumpukan. Untuk informasi lebih lanjut, lihat AWS GitHub repositori amazon-dynamodb-table-to -. global-table-cdk
- Replikasi lintas akun tidak didukung saat ini.

Kesimpulan dan sumber daya

Tabel global DynamoDB memiliki kontrol yang sangat sedikit tetapi masih memerlukan pertimbangan yang cermat. Anda harus menentukan mode tulis, model perutean, dan proses evakuasi. Anda harus melengkapi aplikasi Anda di setiap Wilayah dan siap menyesuaikan rute atau melakukan evakuasi untuk menjaga kesehatan global. Hadiahnya adalah memiliki kumpulan data yang didistribusikan secara global dengan operasi baca dan tulis latensi rendah yang dirancang untuk ketersediaan 99,999%.

Untuk informasi selengkapnya tentang tabel global DynamoDB, lihat sumber daya berikut:

- Dokumentasi Amazon DynamoDB
- Pengendali Pemulihan Aplikasi Amazon Route 53
- ARCpemeriksaan kesiapan (AWS dokumentasi)
- Kebijakan perutean rute 53 (AWS dokumentasi)
- AWS Global Accelerator
- Perjanjian tingkat layanan DynamoDB
- AWS Dasar-dasar Multi-Wilayah (whitepaper)AWS
- Pola desain ketahanan data dengan AWS (AWS re:Invent 2022 presentasi)
- Bagaimana Fidelity Investments dan Reltio dimodernisasi dengan Amazon DynamoDB (presentasi Re:invent 2022)AWS
- Pola desain Multi-Wilayah dan praktik terbaik (AWS re:Invent 2022 presentasi)
- Arsitektur Pemulihan Bencana (DR) pada AWS, BagianIII: Pilot Light dan Warm Standby (posting AWS blog)
- Gunakan pin Wilayah untuk mengatur Wilayah beranda untuk item dalam tabel global AWS Amazon DynamoDB (posting blog)
- Memantau Amazon DynamoDB untuk kesadaran AWS operasional (posting blog)
- Penskalaan DynamoDB: Bagaimana partisi, tombol pintas, dan split untuk kinerja dampak panas (posting blog)AWS

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
AWS Global Accelerator Informasi yang diperbarui	Memperbaiki titik akhir untuk perutean permintaan Global Accelerator.	Maret 14, 2024
Informasi Wilayah AWS dukungan yang diperbarui	Memperbarui FAQ untuk menunjukkan bahwa tabel global sekarang mendukung semua Wilayah AWS.	15 November 2023
Publikasi awal	—	19 Mei 2023

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

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

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

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

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

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

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

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 Perusahaan.

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. 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 penyimpanan atau kelas yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, AWS Panorama menawarkan perangkat yang menambahkan CV ke jaringan kamera lokal, dan Amazon SageMaker menyediakan algoritme pemrosesan gambar untuk CV.

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD 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.

CV

Lihat visi komputer.

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data 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 di tempat. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

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

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi 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 pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

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).

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

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

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

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 lebih 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.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan 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.

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi 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 berbeda Wilayah 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 selengkapnya, lihat Apa itu IoT?

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi 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

Sebuah 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.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

mekanisme

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

akun anggota

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

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

Layanan kecil dan independen yang berkomunikasi 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. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat pembelajaran mesin.

modernisasi

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

penilaian kesiapan modernisasi

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

aplikasi monolitik (monolit)

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

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

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

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

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

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat panduan integrasi operasi.

jejak organisasi

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

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi,

dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan 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.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

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

persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka. 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 pengoptimalan kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

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

principal

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

Privasi oleh Desain

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

zona host 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.

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

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

pseudonimisasi

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

terbitkan/berlangganan (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

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 Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke 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.

PENIPUAN

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh

tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations 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 file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

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

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

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

SPOF

Lihat satu titik kegagalan.

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat 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.