Optimalkan biaya untuk beban kerja Microsoft AWS

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Optimalkan biaya untuk beban kerja Microsoft AWS

Table of Contents

Optimalkan biaya untuk beban kerja Microsoft AWS

Bill Pfeiffer, Chase Lindeman, dan Kevin Sookhan, Amazon Web Services ()AWS

Oktober 2025 (sejarah dokumen)

Gambaran umum

Panduan ini memberikan rekomendasi, praktik terbaik, dan strategi untuk membantu mengoptimalkan biaya beban kerja Microsoft Anda. AWS Panduan ini juga mencakup AWS pengetahuan dasar, teknik pengoptimalan biaya, dan arsitektur referensi untuk membantu Anda membangun dan mengotomatiskan beban kerja berkinerja tinggi yang hemat biaya yang memenuhi tujuan bisnis Anda. Secara kolektif, panduan ini disebut sebagai Microsoft on AWS Cost Optimization (MACO). Panduan MACO dikembangkan oleh pakar industri dan didasarkan pada skenario dunia nyata.

Panduan ini mencakup beban kerja Microsoft berikut:

- Windows di Amazon Elastic Compute Cloud (Amazon EC2)
- SQL Server
- Wadah
- Penyimpanan
- Active Directory
- .NET

Audiens

Panduan ini ditujukan untuk arsitek, insinyur, administrator, direktur, pembuat keputusan teknis CTOs, dan AWS Mitra. Sangat membantu tetapi tidak perlu memiliki pengalaman sebelumnya dan pemahaman dasar tentang AWS penagihan, teknologi Microsoft, dan administrasi AWS sistem.

Cara menggunakan panduan ini

Anda dapat menggunakan panduan ini untuk merencanakan dan mengimplementasikan perjalanan MACO Anda ke cloud. Kami menyarankan Anda membaca panduan ini dari awal hingga akhir untuk mendapatkan pemahaman yang komprehensif tentang opsi dan pendekatan untuk mengoptimalkan

biaya beban kerja Microsoft Anda. AWS Anda dapat meninjau bagian beban kerja berikut berdasarkan kebutuhan organisasi Anda:

- Windows di Amazon EC2
- SQL Server
- Wadah
- Penyimpanan
- Direktori Aktif
- .NET

> **Important**
>
> Sampel kode yang disediakan dalam panduan ini hanya untuk tujuan demonstrasi. Ini adalah praktik terbaik untuk menguji semua kode di lingkungan pengembangan sebelum menggunakannya di lingkungan produksi. Sebelum Anda menerapkan kode apa pun, kami sarankan Anda menguji kode Anda dalam batch kecil dan kemudian meninjau perubahan biaya yang dihasilkan dari kode dengan menggunakan AWS Cost Explorer. Ini dapat membantu Anda memecahkan masalah kasus tepi dan masalah lain yang dapat menjadi masalah di kemudian hari.

> **Important**
>
> Contoh harga dalam panduan ini didasarkan pada harga pada saat publikasi. Harga dapat berubah sewaktu-waktu. Selain itu, biaya Anda dapat bervariasi tergantung pada Anda Wilayah AWS, Layanan AWS kuota, dan faktor lain yang terkait dengan lingkungan cloud Anda.

Hasil bisnis yang ditargetkan

Panduan ini dapat membantu Anda dan organisasi Anda mencapai hasil bisnis berikut:

- Pelajari cara menggunakan AWS Optimization and Licensing Assessment (AWS OLA) untuk menilai dan mengoptimalkan lingkungan lokal dan cloud Anda saat ini, berdasarkan pemanfaatan sumber daya, lisensi pihak ketiga, dan dependensi aplikasi.
- Kembangkan kasus bisnis untuk pengoptimalan biaya dengan menggunakan Kalkulator AWS Modernisasi untuk Beban Kerja Microsoft.
- Optimalkan biaya untuk beban kerja Microsoft spesifik Anda, termasuk beban kerja untuk Windows di Amazon Elastic Compute Cloud (Amazon EC2), SQL Server, container, penyimpanan, Active Directory, dan .NET.

Perjalanan pengoptimalan biaya

Ruang lingkup, waktu, dan jalur spesifik perjalanan migrasi cloud Anda bergantung pada tujuan bisnis Anda, persyaratan teknis, dan faktor lainnya. Bagian ini memberikan contoh perjalanan migrasi cloud yang berfokus pada Cloud Financial Management dengan AWS dan mematuhi rekomendasi dan praktik terbaik MACO. Anda dapat menggunakan contoh ini untuk mendapatkan pemahaman tentang cara merancang perjalanan migrasi cloud untuk beban kerja Microsoft.

Tugas tingkat tinggi berikut menggambarkan pendekatan yang dapat diambil organisasi untuk menerapkan rekomendasi dan praktik terbaik MACO:

- Tetapkan strategi penandaan dan aktifkan tag alokasi biaya yang ditentukan pengguna. Untuk informasi selengkapnya, lihat AWS Whitepaper Best Practices for AWS Tagging Resources.
- Tentukan anggaran berdasarkan aplikasi, tim, atau departemen. Untuk informasi selengkapnya, lihat Mengelola biaya Anda AWS Budgets di Panduan Pengguna AWS Billing and Cost Management.
- Lakukan AWS Optimalisasi dan Penilaian Lisensi (AWS OLA) untuk mempercepat penghematan. Untuk informasi selengkapnya, lihat AWS Optimasi dan Penilaian Lisensi dalam AWS dokumentasi.
- Bawa Lisensi Anda Sendiri (BYOL) untuk beban kerja Windows dan SQL Server dengan menggunakan. Amazon Elastic Compute Cloud Dedicated Hosts Untuk informasi lebih lanjut, lihat Bawa lisensi untuk beban kerja Windows dan SQL Server bagian panduan ini.
- Optimalkan lisensi SQL Server Anda pada. AWS Untuk informasi lebih lanjut, lihat Memahami lisensi SQL Server bagian panduan ini.
- Pilih jenis instans yang tepat untuk beban kerja Windows. Untuk informasi lebih lanjut, lihat Pilih jenis instans yang tepat untuk beban kerja Windows bagian panduan ini.
- Pilih jenis instance yang tepat untuk beban kerja SQL. Untuk informasi lebih lanjut, lihat Pilih instans EC2 yang tepat untuk beban kerja SQL Server bagian panduan ini.
- Migrasikan Amazon Elastic Block Store (Amazon EBS) dari gp2 ke gp3. Untuk informasi lebih lanjut, lihat Migrasikan volume Amazon EBS dari gp2 ke gp3 bagian panduan ini.
- Kontrol beban kerja dengan Penjadwal EC2 Instance aktif. AWS Untuk informasi lebih lanjut, lihat Mengotomatiskan jadwal berhenti dan mulai bagian panduan ini.
- Hapus biaya SQL Server untuk beban kerja non-produksi dengan menggunakan SQL Server Developer Edition. Untuk informasi lebih lanjut, lihat Evaluasi edisi Pengembang SQL Server bagian panduan ini.

- Gunakan Availability Zone tunggal untuk Amazon FSx untuk Windows File Server untuk pengembangan dan pengujian beban kerja. Untuk informasi lebih lanjut, lihat Gunakan satu Availability Zone bagian panduan ini.
- Rightsize beban kerja Windows Anda dengan menggunakan. AWS Compute Optimizer Untuk informasi lebih lanjut, lihat Beban kerja Windows ukuran yang tepat bagian panduan ini.
- Optimalkan pengeluaran di Windows di Amazon EC2 dengan menggunakan Savings Plans. Untuk informasi lebih lanjut, lihat Optimalkan pengeluaran untuk Windows di Amazon EC2 bagian panduan ini.
- Aktifkan deduplikasi data FSx untuk Windows File Server. Untuk informasi lebih lanjut, lihat Aktifkan deduplikasi data di Amazon FSx bagian panduan ini.
- Gunakan sharding data untuk sistem file aktif FSx untuk Windows File Server. Untuk informasi lebih lanjut, lihat Memahami sharding data FSx untuk Windows File Server bagian panduan ini.
- Optimalkan strategi pencadangan SQL Server Anda. Untuk informasi lebih lanjut, lihat Optimalkan strategi pencadangan SQL Server bagian panduan ini.
- Buat aplikasi framework .NET statis mendukung penskalaan dinamis. Untuk informasi lebih lanjut, lihat panduan ini. Mendukung penskalaan dinamis untuk aplikasi.NET Framework statis
- Gunakan layanan mikro .NET tanpa server. Untuk informasi lebih lanjut, lihat Pertimbangkan .NET tanpa server bagian panduan ini.
- Pindahkan aplikasi Windows Anda ke wadah. Untuk informasi lebih lanjut, lihat Kontainerisasi aplikasi.NET bagian panduan ini.
- Gunakan AWS Compute Optimizeruntuk ukuran kanan wadah Windows yang berjalan untuk AWS Fargate Amazon Elastic Container Service (Amazon ECS). Untuk informasi lebih lanjut, lihat Aktifkan Compute Optimizer bagian panduan ini.
- Refactor ke .NET modern dan pindah ke Linux. Untuk informasi lebih lanjut, lihat Refactor ke .NET modern dan pindah ke Linux bagian panduan ini.
- Manfaatkan instance dan wadah Graviton. Untuk informasi lebih lanjut, lihat Gunakan instance dan wadah Graviton bagian panduan ini.
- Memodernisasi database SQL Server. Untuk informasi lebih lanjut, lihat Memodernisasi database SQL Server bagian panduan ini.
- Desain infrastruktur Direktori Aktif. Untuk informasi lebih lanjut, lihat Active Directory bagian panduan ini.

Untuk informasi selengkapnya tentang perjalanan pelanggan yang berfokus pada Cloud Financial Management dengan AWS, lihat AWS Whitepaper kemampuan Cloud Financial Management.

Rekomendasi teratas untuk mengoptimalkan biaya

Gambaran umum

Optimalisasi biaya adalah salah satu pilar dari AWS Well-Architected Framework dan memainkan peran penting dalam rencana migrasi cloud Anda. Anda akan menemukan rekomendasi untuk pengoptimalan biaya di seluruh panduan ini, tetapi bagian ini menyebutkan rekomendasi dengan dampak tertinggi. Anda dapat menerapkan rekomendasi ini dengan cepat dan mereka akan memiliki dampak yang signifikan pada organisasi Anda. Rekomendasi ini dapat membantu meletakkan dasar bagi seluruh upaya pengoptimalan biaya Anda.

Rekomendasi teratas

Tabel berikut mencantumkan rekomendasi teratas untuk pengoptimalan biaya berdampak tertinggi. Kolom “Sulit untuk diterapkan” menilai setiap pengoptimalan berdasarkan skala apa yang paling mudah diterapkan (1) ke apa yang paling sulit untuk diterapkan (5). Kolom “Estimasi penghematan” menunjukkan perkiraan berbasis persentase berapa banyak organisasi Anda dapat menabung untuk setiap pengoptimalan yang direkomendasikan.

Pengoptimalan	Kesulitan untuk menerapkan	Perkiraan penghematan
Beban kerja Windows ukuran yang tepat	3	25%
Bawa lisensi untuk beban kerja Windows dan SQL Server	3	30%
Evaluasi edisi Pengembang SQL Server	2	20%
Memahami lisensi SQL Server	2	Hingga 50%
Mengotomatiskan jadwal berhenti dan mulai	3	Hingga 40%

Pengoptimalan	Kesulitan untuk menerapkan	Perkiraan penghematan
Pilih jenis instans yang tepat untuk beban kerja Windows	1	10— 30%
Refactor ke .NET modern dan pindah ke Linux	5	10— 20%
Optimalkan pengeluaran untuk Windows di Amazon EC2	3	Hingga 20— 40%
Migrasikan volume Amazon EBS dari gp2 ke gp3	4	Hingga 20%

Important

Perkiraan penghematan pada tabel sebelumnya berlaku untuk setiap domain teknis individu, bukan AWS pengeluaran keseluruhan dalam akun. Misalnya, Anda dapat menerapkan Penjadwal Instance dalam berbagai jenis dan ukuran lingkungan yang dapat mengubah potensi penghematan. Perkiraan berlaku khusus untuk biaya EC2 instans Amazon dan tidak menyiratkan penghematan keseluruhan untuk yang lain Layanan AWS. Perkiraan ini disediakan sebagai pengukur, bukan jaminan.

Pakar MACO tersedia untuk berbicara tentang pengoptimalan biaya secara lebih mendalam. Untuk menyiapkan rapat untuk menyelami kasus penggunaan Anda, hubungi tim akun Anda atau email optimize-microsoft@amazon.com.

AWS Optimalisasi dan Penilaian Perizinan

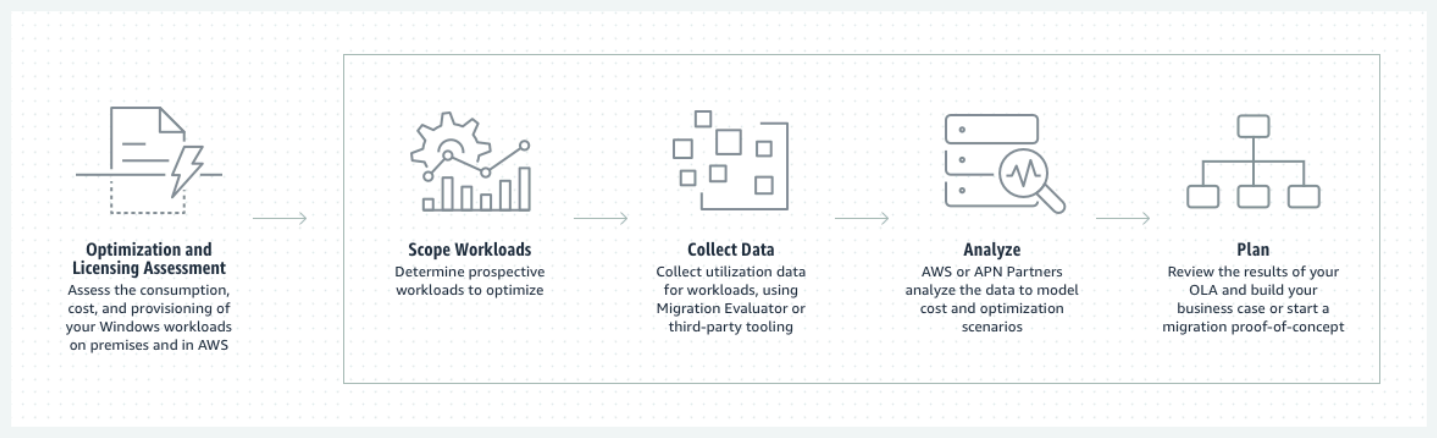
Gambaran umum

Penilaian AWS Pengoptimalan dan Lisensi (AWS OLA) dapat membantu Anda menilai dan mengoptimalkan lingkungan cloud lokal dan yang ada saat ini, berdasarkan pemanfaatan sumber daya, lisensi pihak ketiga, dan dependensi aplikasi. Anda dapat menggunakan AWS OLA untuk membantu organisasi Anda membangun strategi migrasi dan lisensi yang membuka penghematan biaya saat Anda bermigrasi ke AWS atau menilai beban kerja Microsoft yang ada. AWS AWS OLA juga dapat membantu Anda mencapai hal-hal berikut:

- Memahami penerapan, kinerja aplikasi, dan kontrak yang ada.
- Ukuran sumber daya Anda yang tepat.
- Kembangkan peta jalan ke. AWS Cloud
- Mengurangi atau menghilangkan biaya dengan menggunakan investasi yang ada dan hanya membayar untuk apa yang Anda gunakan.

Kami menyarankan Anda membuat AWS OLA langkah pertama dalam perjalanan pengoptimalan biaya Anda. Anda dapat bekerja dengan AWS Partner Network untuk menyelesaikan AWS OLA. Mereka akan membantu Anda mengumpulkan data penilaian dan memberi Anda rekomendasi untuk mengoptimalkan biaya lisensi dan instans Anda.

Diagram berikut memberikan gambaran umum tentang proses penilaian.



Opsi penilaian

Anda dapat memilih dari dua opsi AWS OLA untuk beban kerja Microsoft Anda di AWS:

- Versi Lite — Dalam kasus penggunaan ini, semua beban kerja Anda aktif. VMware Anda dapat AWS memberikan output dari RVTools. Kemudian, AWS dapat menawarkan waktu penyelesaian 1-5 hari. Pendekatan ini menggunakan point-in-time informasi yang ditarik langsung dari VMware vCenter untuk mengembangkan rekomendasi ukuran dan menawarkan opsi harga sesuai permintaan.
- Versi lengkap — Dalam kasus penggunaan ini, Anda memiliki lingkungan campuran yang berjalan di berbagai penyedia cloud, server fisik, dan server virtual. AWS menggunakan agen sistem operasi untuk mengumpulkan data penggunaan dari 14 hingga 30 hari. Hal ini memungkinkan AWS untuk membuat keputusan ukuran instans berdasarkan pola penggunaan aplikasi Anda. AWS menggunakan beberapa alat pihak ketiga, seperti Cloudamize, untuk menyelesaikan analisis. AWS bekerja dengannya AWS Partner Network untuk membantu memberikan penilaian total biaya kepemilikan (TCO) akhir dengan beberapa opsi harga yang memperhitungkan model penetapan harga dan arsitektur yang berbeda.

Penilaian penuh

Penilaian AWS OLA lengkap dimulai dengan panggilan telepon satu jam. Selama panggilan ini, AWS membantu Anda menentukan AWS infrastruktur yang paling optimal untuk mendukung migrasi, memilih metode pengumpulan data, dan menetapkan garis waktu penyelesaian. Menerapkan alat penemuan di organisasi Anda bergantung pada metode pengumpulan data, ukuran organisasi Anda, dan alat yang digunakan organisasi Anda untuk mengelola armada servernya. Biasanya dibutuhkan waktu dua minggu untuk mengumpulkan data penggunaan.

Proses AWS OLA penuh memakan waktu antara 30-45 hari dan terdiri dari fase-fase berikut:

- Beban kerja lingkup
- Kumpulkan data
- Menganalisis data
- Rencanakan langkah selanjutnya

Beban kerja lingkup

Pertama, AWS bekerja dengan Anda dan tim Anda untuk menentukan ruang lingkup penilaian. Ini biasanya dipecah berdasarkan jenis lingkungan (misalnya, non-produksi dan produksi). Ruang lingkup mencakup lokasi beban kerja. Ini bisa berupa beban kerja tempat Anda bermigrasi AWS, beban kerja yang sudah berjalan AWS (misalnya, OLA AWS untuk Amazon EC2), atau beban kerja yang berjalan di penyedia cloud lainnya.

Kumpulkan data

Selanjutnya, AWS gunakan perkakas untuk membantu penemuan sumber daya dan mengumpulkan data kinerja dari server Anda. Perkakas ini hadir dalam empat opsi penerapan:

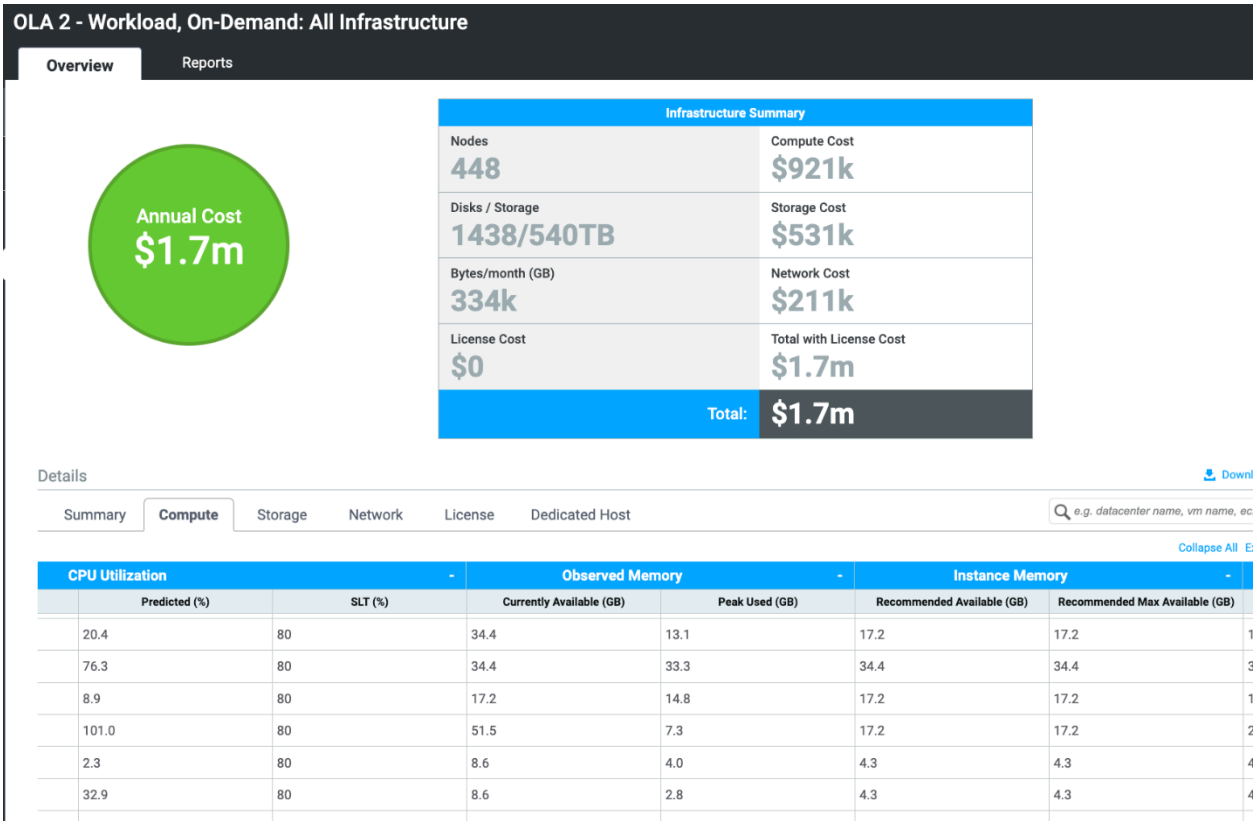
- Alat yang dapat menanyakan hypervisor (hanya membutuhkan kredenal VMware vCenter atau Hyper-V)
- Agen yang dapat digunakan pada mesin fisik atau virtual
- Penemuan tanpa agen dengan menggunakan SSH, Windows Remote Management (WinRM), atau Windows Management Instrumentation (WMI) tergantung pada lingkungan dan sistem operasi Anda
- Pengumpulan dan analisis data file datar

Untuk penerapan perkakas, Anda dapat mencampur dan mencocokkan setiap opsi dan mengkonsolidasikan hasil. Sangat penting untuk memastikan bahwa opsi apa pun yang Anda pilih tidak membebani sumber daya TI Anda. AWS berusaha untuk membuat proses penilaian sebagai turnkey mungkin. Selain panggilan telepon singkat untuk membantu penyiapan, tim AWS OLA dan arsitek solusi spesialis Microsoft akan menyiapkan analisis biaya kepemilikan total (TCO) dan rekomendasi untuk ditinjau.

Pengumpulan data biasanya memakan waktu dua hingga tiga minggu ketika pemanfaatan CPU, pemanfaatan RAM, throughput penyimpanan, IOPS, dan throughput jaringan dianalisis. Idealnya, pengumpulan ini berlangsung selama waktu puncak bulan bisnis Anda (misalnya, selama pelaporan end-of-month keuangan). AWS ingin menangkap penggunaan puncak karena ini memberikan sampel statistik yang baik untuk AWS instance berukuran tepat, sambil tetap menjamin kinerjanya dapat melebihi apa yang tersedia di tempat. AWS menggabungkan metrik pemanfaatan dengan heuristik kinerja dari berbagai generasi prosesor untuk menargetkan dengan tepat berapa banyak CPU dan RAM yang dibutuhkan oleh beban kerja tertentu. Target ini biasanya kurang dari apa yang

dialokasikan di tempat. Ini tidak hanya mengurangi biaya komputasi untuk ukuran instance tetapi juga mengoptimalkan biaya lisensi.

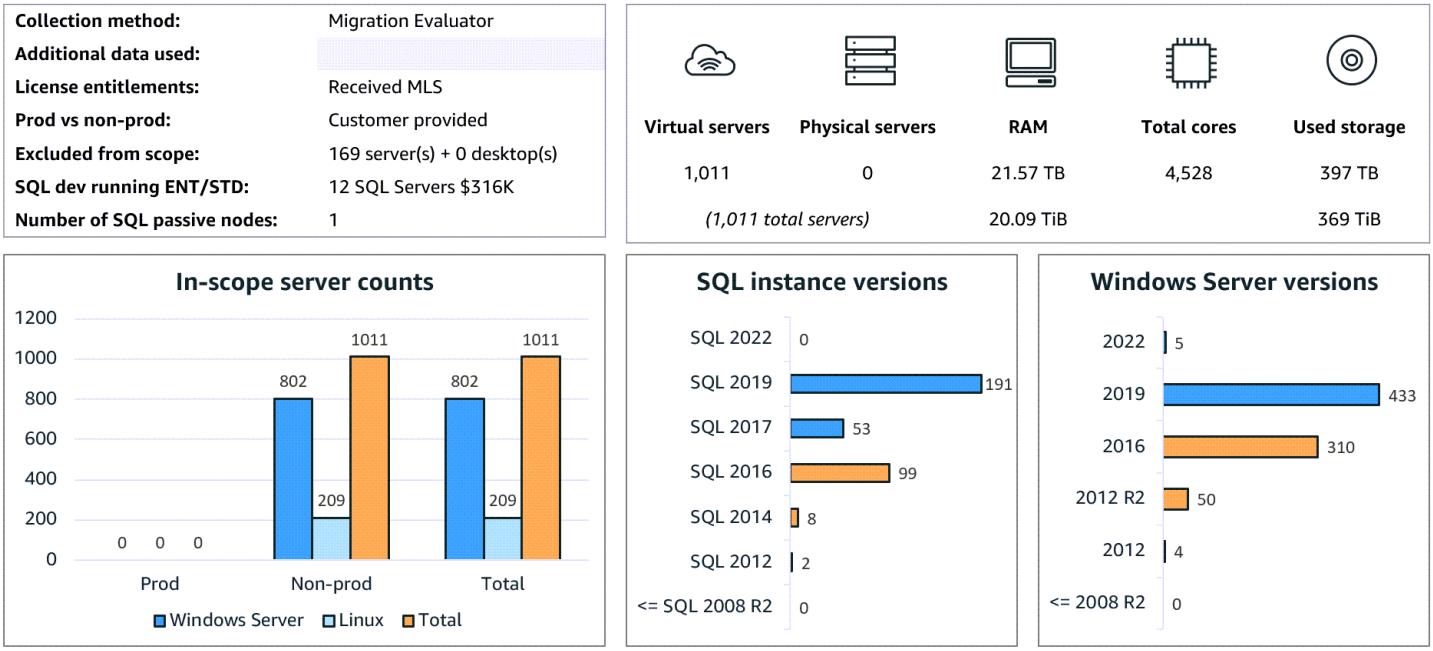
Tampilan dasbor berikut menunjukkan contoh biaya infrastruktur yang dapat ditangkap oleh penilaian.



Menganalisis data

AWS memberikan presentasi tanya jawab setelah pengumpulan data selesai. AWS meninjau data, merangkum temuan, dan kemudian membuat rekomendasi untuk penggunaan lokal dan migrasi cloud. Anda dapat mengurangi biaya komputasi dan lisensi dengan memeriksa peluang konsolidasi, peningkatan elastisitas (di mana beban kerja dapat dimatikan atau disesuaikan secara musiman), peluang SKU kanan (misalnya, edisi SQL Server Enterprise sedang digunakan, tetapi persyaratan sumber daya dan penggunaan fitur menyarankan SQL Server edisi Standar cocok). Untuk produk seperti SQL Server yang dilisensikan oleh inti, seringkali masuk akal secara finansial untuk menempatkan beban kerja dalam contoh komputasi yang lebih mahal. Artinya, jika profil CPU dan rasio RAM terhadap vCPU menyajikan efek bersih untuk mengurangi jumlah inti berlisensi untuk kasus penggunaan yang disertakan lisensi dan Bring Your Own License (BYOL).

Berikut ini menunjukkan contoh analisis berdasarkan data yang dikumpulkan oleh penilaian.



Skenario pengoptimalan umum termasuk mengidentifikasi peluang pengoptimalan AWS sumber daya dan penghematan lisensi pihak ketiga.

Contoh peluang optimasi AWS sumber daya:

- Hindari penyediaan berlebihan untuk penggunaan puncak.
- Hindari sumber daya yang terlalu menentukan dan kurang menggunakan.
- Ukuran instans Anda dengan tepat dan bermigrasi ke generasi instans terbaru. EC2
- Menghemat biaya operasi dengan pindah ke database terkelola.

Contoh penghematan lisensi pihak ketiga:

- Kurangi inti yang diperlukan untuk menjalankan beban kerja yang sama.
- Singkirkan paket edisi dan add-on SQL Server Enterprise yang tidak perlu.
- Hapus server zombie dan ganti perangkat keras yang sudah ketinggalan zaman.
- Gunakan opsi BYOL dan termasuk lisensi untuk mengurangi perjanjian komersial masa depan.
- Modernisasi menjadi solusi open-source dan cloud-native.

Rencanakan langkah selanjutnya

Terakhir, AWS gunakan data kinerja yang dikumpulkan untuk memperkirakan ukuran dan biaya beban kerja tertentu. AWS juga dapat melihat secara agregat pada lingkungan cakupan Anda dan memberikan analisis kuantitatif. Ini dapat membantu Anda menentukan apakah opsi terbaik adalah penyegaran lokal atau migrasi ke AWS. Anda dapat membangun kasus bisnis ekonomi cloud dengan menggunakan ringkasan analisis TCO (seperti yang ditunjukkan pada contoh berikut) yang disediakan di akhir AWS OLA.

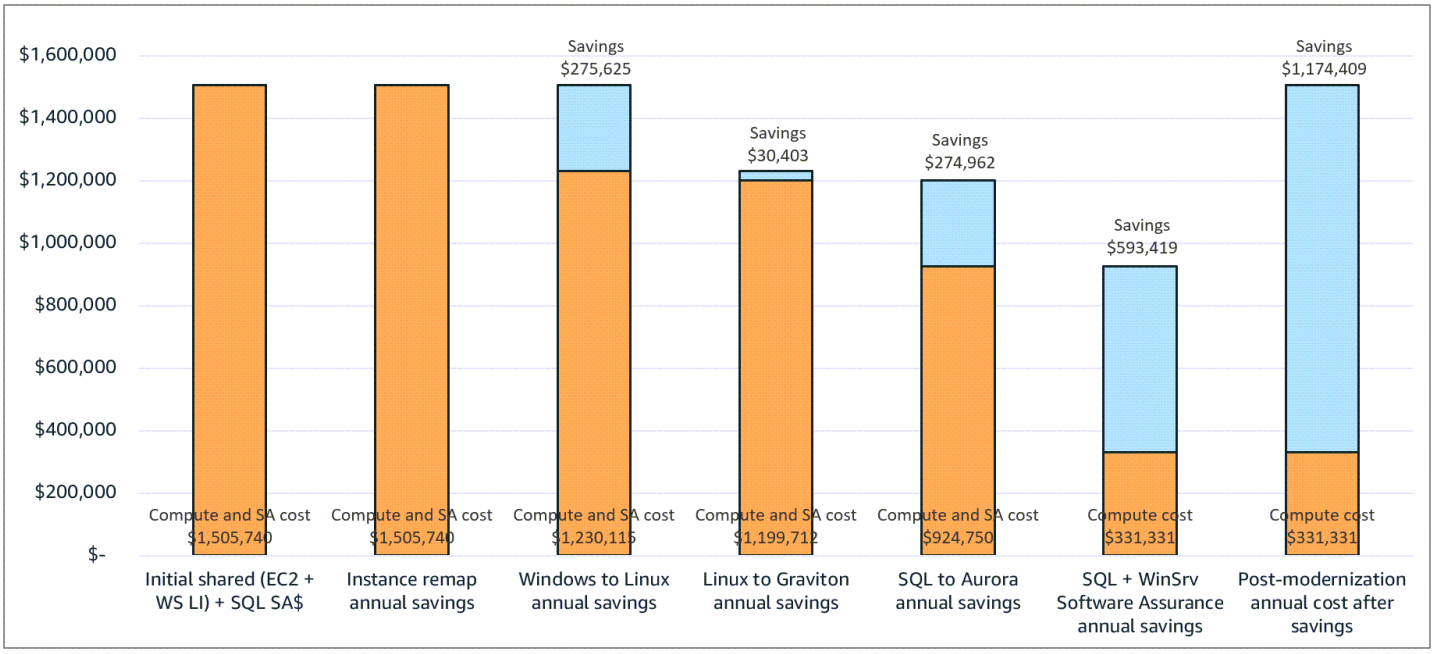
	Option 1: Amazon EC2 shared	**Option 1a: Amazon EC2 shared + power management**	**Option 2: Amazon EC2 mixed**	**Option 2a: Amazon EC2 mixed + power management**
Option details: compute	*100% Reserved Instances (RIs)*	*RIs + on-demand power management*	*100% Ris*	*RIs + on-demand power management*
Option details: Microsoft licenses	*WS LI and SQL BYOL*	*WS LI and SQL BYOL*	*WS BYOL or LI+SQL BYOL*	*WS BYOL or LI+SQL BYOL*
Compute costs[1]				
Year 1 compute cost	$414,546	$482,623	$504,019	$513,941
Year 1 vendor license included cost	$392,858	$244,415	$9,804	$4,783
	$807,404	**$727,038**	**$513,823**	**$518,724**
Total compute savings in year 1, compared to Option 1	—	*10% ($80,366)*	*36% ($293,581)*	*36% ($288,680)*
Storage and networking costs[2]				
Annual estimated storage cost	$336,494	$336,494	$336,494	$336,494
Annual estimated networking cost	$41,455	$41,455	$41,455	$41,455
	$377,949	**$377,949**	**$377,949**	**$377,949**
Microsoft license costs**				
WS/CIS annual Software Assurance (SA) + current SPLA/Subs cost	$0	$0	$0	$0
WS/CIS license + SA + SPLA/Subs true-up cost	$0	$0	$0	$0
SQL annual SA + current SPLA/Subs cost	$0	$0	$0	$0
SQL license SA + current SPLA/Subs true-up cost	$0	$0	$0	$0
	$0	**$0**	**$0**	**$0**
Total estimated costs	**$1,185,353**	**$1,104,987**	**$891,772**	**$896,673**
Annual TCO savings in year 1, compared to Option 1	—	*7% ($80,366)*	*25% ($293,581)*	*24% ($288,680)*

[1] Pricing model used: 3-year, no upfront RI
[2] Software Assurance and true-up costs provided by Microsoft

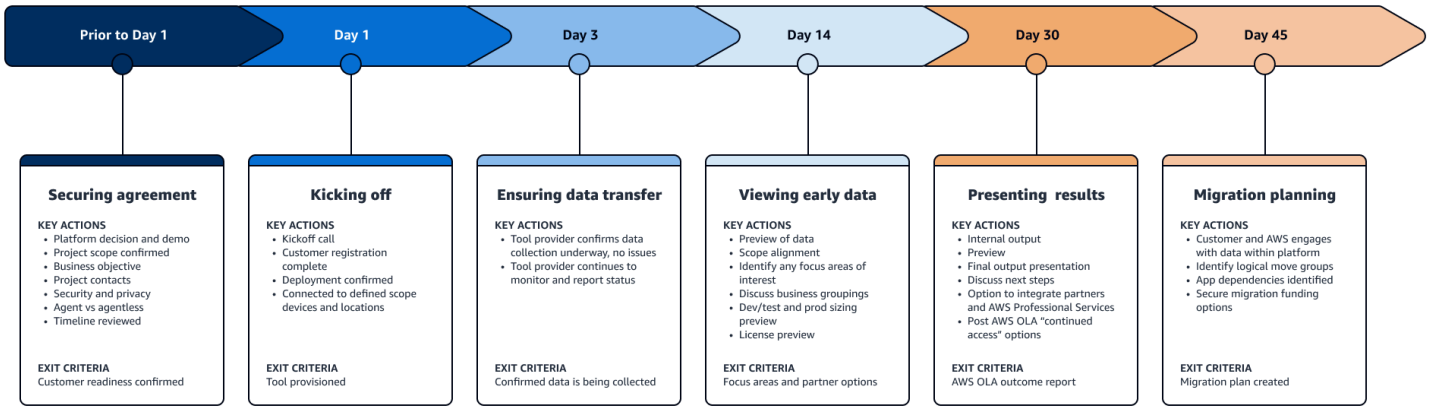
AWS OLA juga memberikan wawasan tentang dampak modernisasi terhadap beban kerja Anda yang ada dengan membuat saran seperti berikut:

- Pindah ke sistem operasi Linux.
- Tambahkan dukungan aplikasi untuk prosesor ARM (AWS Graviton).
- Pindahkan beban kerja SQL Server ke Amazon Aurora.
- Hapus jaminan perangkat lunak dengan memindahkan beban kerja Windows dan SQL Server ke teknologi sumber terbuka.

Diagram berikut menunjukkan penghematan biaya yang dapat dicapai melalui teknik modernisasi seperti pindah dari Windows ke Linux atau dari SQL Server ke Aurora.



Proses AWS OLA penuh memakan waktu sekitar 45 hari dari awal hingga akhir. Diagram berikut menunjukkan contoh timeline.



Jika Anda memiliki VMware lingkungan yang murni dan dapat memberikan output dari RVTools, maka Anda dapat mengurangi timeline ini menjadi satu minggu bisnis. Selain itu, AWS dapat menganalisis file datar yang mencakup data aset dan pemanfaatan, seperti rata-rata CPU, puncak CPU, rata-rata RAM, dan puncak RAM.

Dampak penilaian

Rata-rata pelanggan biasanya mengalami pengurangan biaya 20-30 persen dari upaya ukuran yang tepat. Ukuran yang tepat cocok dengan beban kerja sumber dengan AWS instance berukuran terbaik berdasarkan data penggunaan. Penyesuaian ukuran yang tepat ini tidak hanya mengurangi biaya

bulanan AWS lingkungan, tetapi sering menghasilkan penghematan di tempat lain dalam organisasi. Misalnya, keuntungan 20-30 persen dari lisensi Windows atau SQL Server dapat mengurangi true-up berikutnya dengan Microsoft, atau membebaskan lisensi untuk aplikasi tambahan. line-of-business Konsolidasi dan ukuran yang tepat dari beban kerja SQL Server umumnya di mana keuntungan finansial paling dramatis direalisasikan.

AWS dapat membantu Anda mengkategorikan sistem ke dalam ember modernisasi. Beberapa sistem adalah warisan dan tidak layak secara finansial untuk disentuh, sementara yang lain dapat dimodernisasi menjadi wadah atau aplikasi tanpa server di mana penghematan paling signifikan direalisasikan. Percakapan dengan AWS tim Anda beralih dari topik umum tentang apa yang dimungkinkan cloud ke diskusi yang lebih spesifik tentang bagaimana dan mengapa beban kerja tertentu harus dimodernisasi. AWS juga membantu Anda mengeksplorasi peluang inovasi potensial.

Langkah berikutnya

Jika Anda memulai perjalanan pengoptimalan biaya untuk beban kerja Microsoft yang berjalan di lingkungan lokal atau di tempat AWS, berinteraksi dengan tim AWS akun Anda dan minta OLA. AWS AWS anggota tim dapat menjawab pertanyaan Anda dan membantu Anda memutuskan apakah AWS OLA pada akhirnya merupakan pilihan yang tepat untuk Anda dan organisasi Anda. Atau, Anda dapat meminta AWS OLA secara online.

Sumber daya tambahan

- AWS Optimalisasi dan Penilaian Lisensi (AWS dokumentasi)
- AWS re:invent 2022 - Cara menghemat biaya dan mengoptimalkan beban kerja Microsoft pada AWS (05) () ENT2 YouTube

Windows di Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) adalah platform komputasi awan yang sangat fleksibel dan terukur yang ideal untuk menjalankan beban kerja Windows Anda. Anda dapat menggunakan Amazon EC2 untuk menyebarkan, mengelola, dan menskalakan beban kerja Windows Server Anda pada infrastruktur yang aman, andal, sangat tersedia, dan dapat disesuaikan. AWS Cloud Pertimbangkan manfaat utama berikut menjalankan beban kerja Windows di Amazon EC2:

- Skalabilitas - Amazon EC2 memungkinkan Anda untuk dengan mudah menskalakan beban kerja Windows Anda untuk mengakomodasi perubahan persyaratan. Anda dapat dengan cepat membuat instans EC2 baru untuk menangani peningkatan permintaan, dan dengan mudah menghentikan instans saat tidak lagi diperlukan. Anda hanya membayar untuk sumber daya yang benar-benar Anda gunakan.
- Fleksibilitas — Windows di Amazon EC2 mendukung berbagai jenis instans yang dirancang untuk memenuhi berbagai persyaratan beban kerja, mulai dari instans tujuan umum hingga memori atau instans yang dioptimalkan untuk komputasi. Fleksibilitas ini memastikan bahwa Anda dapat memilih jenis instans terbaik untuk aplikasi berbasis Windows spesifik Anda, memaksimalkan kinerja dan meminimalkan biaya.
- Keamanan — AWS menyediakan beberapa lapisan keamanan untuk beban kerja Windows Anda, termasuk firewall jaringan, enkripsi data, dan kontrol akses aman. Ini berarti Anda dapat mempercayai bahwa aplikasi dan data Anda dilindungi, sambil tetap memiliki kontrol penuh atas pengaturan dan konfigurasi keamanan Anda.
- Efisiensi biaya — Model pay-as-you-go penetapan harga memungkinkan Anda membayar hanya untuk sumber daya yang Anda gunakan—menghilangkan kebutuhan untuk investasi di muka dalam perangkat keras dan perangkat lunak. Model ini juga memungkinkan Anda untuk mengoptimalkan biaya, mengurangi pengeluaran modal, dan meningkatkan efisiensi operasional. Ini adalah model harga yang ideal untuk bisnis dari semua ukuran.

Bagian panduan ini mencakup topik-topik berikut:

- Mengotomatiskan jadwal berhenti dan mulai
- Beban kerja Windows ukuran yang tepat
- Pilih jenis instans yang tepat untuk beban kerja Windows
- Bawa lisensi untuk beban kerja Windows dan SQL Server
- Optimalkan pengeluaran untuk Windows di Amazon EC2

- Pantau biaya menggunakan AWS alat

Mengotomatiskan jadwal berhenti dan mulai

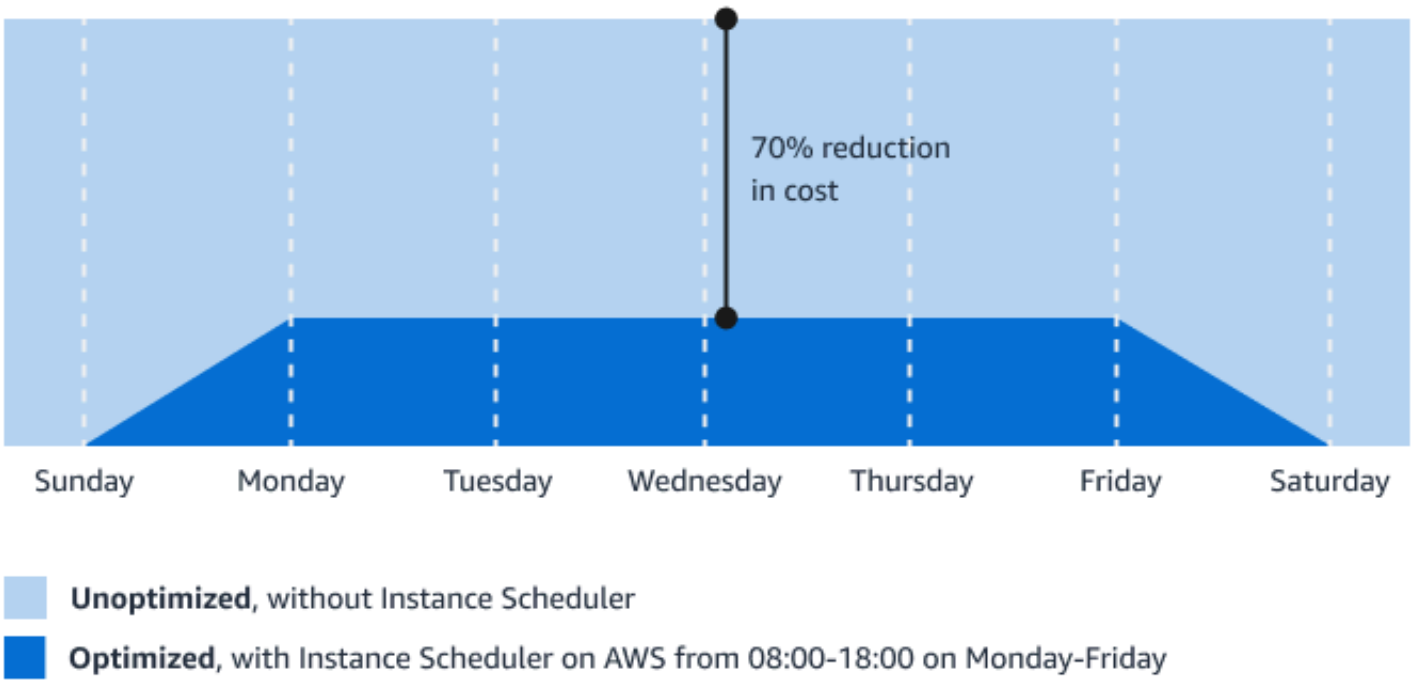
Ikhtisar

Penjadwal Instans aktif AWS dapat membantu Anda mengurangi biaya operasional dengan mengotomatiskan memulai dan menghentikan instans Amazon EC2 dan Amazon Relational Database Service (Amazon RDS). Jika Anda membiarkan semua instans Anda berjalan pada pemanfaatan penuh terus-menerus, Anda bisa berakhir membayar untuk sumber daya yang tidak digunakan. Penjadwal Instance aktif AWS memungkinkan Anda untuk mematikan instance selama waktu ketika tidak diperlukan, seperti selama jam non-bisnis, akhir pekan, atau periode lain saat penggunaan rendah. Hal ini dapat menyebabkan penghematan biaya yang signifikan dari waktu ke waktu.

Instance Scheduler on AWS juga menawarkan penjadwalan instans lintas akun, penandaan otomatis, dan kemampuan untuk mengonfigurasi jadwal atau periode dengan menggunakan antarmuka baris perintah atau jendela pemeliharaan. AWS Systems Manager Fitur-fitur ini dapat membantu Anda mengelola instans dengan lebih efektif dan akurat melacak dan mengalokasikan biaya di berbagai proyek atau tim.

Studi kasus

Pertimbangkan contoh perusahaan yang menggunakan Penjadwal Instance AWS di lingkungan produksi untuk secara otomatis menghentikan instance di luar jam kerja setiap hari. Jika perusahaan ini membiarkan semua instansinya berjalan dengan pemanfaatan penuh, mereka dapat mencapai penghematan biaya hingga 70 persen untuk kasus-kasus yang hanya diperlukan selama jam kerja reguler. Bagan berikut menunjukkan bagaimana pemanfaatan mingguan dikurangi dari 168 jam menjadi 50 jam.

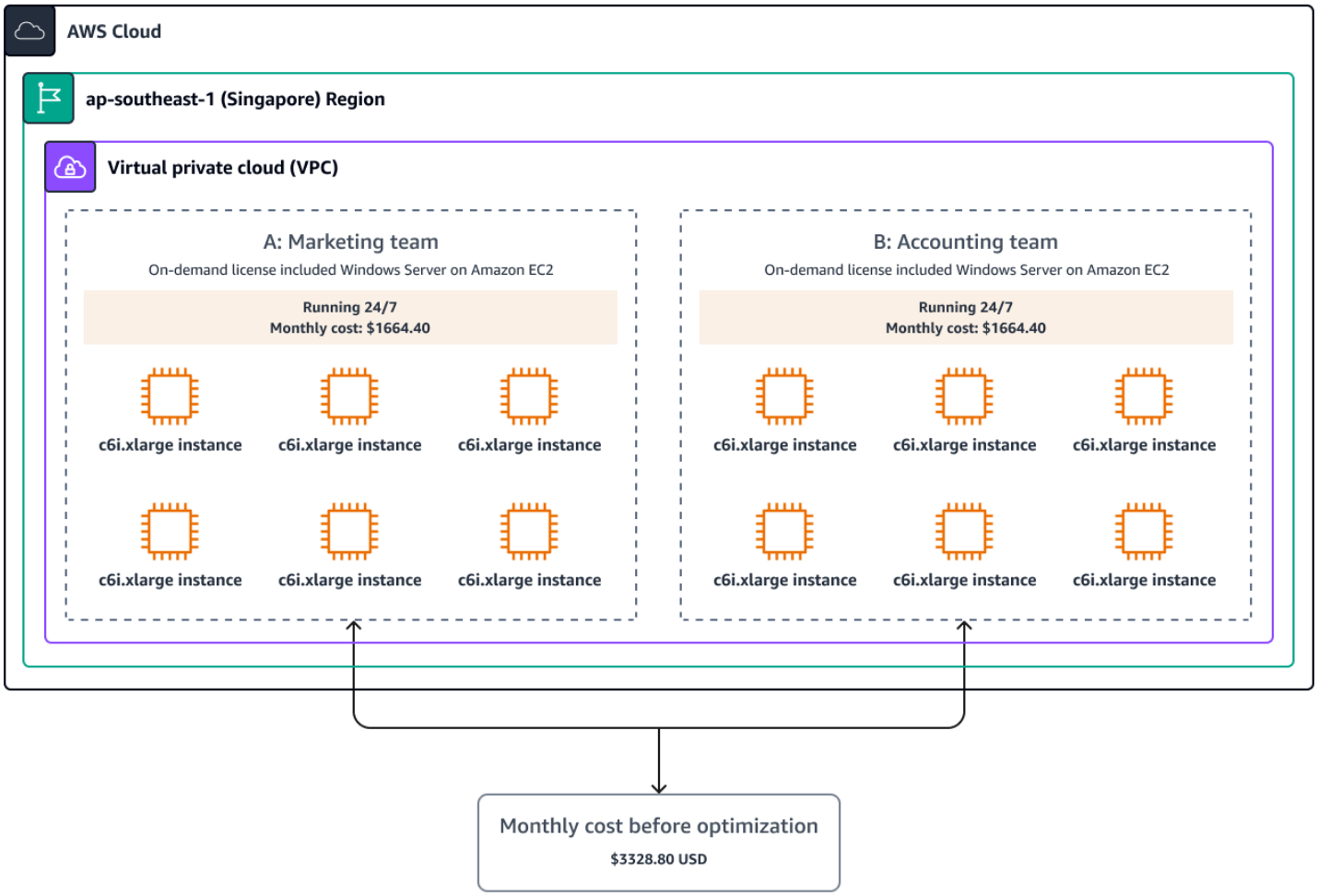


Pertimbangkan contoh lain. Perusahaan utilitas listrik Jamaika Public Service Company Limited (JPS) memigrasikan database-nya ke Amazon RDS. Sekarang, JPS menggunakan Amazon EC2 untuk meng-host layanan API dan menjalankan aplikasi lain. Untuk JPS, Instance Scheduler AWS menjadi alat utama untuk mengelola lingkungan non-produksi. JPS menggunakan Penjadwal Instance AWS untuk mengurangi biaya pengembangan dan mengelola instans EC2 berdasarkan kebutuhan tim dan jadwal kerja. Ini membantu JPS mengurangi biaya hingga 40 persen. Untuk informasi selengkapnya, lihat studi AWS kasus Layanan Publik Jamaika Bermigrasi Secara Efisien ke Cloud, Mengurangi Biaya sebesar 40% Menggunakan Penjadwal Instance. AWS

Skenario pengoptimalan biaya

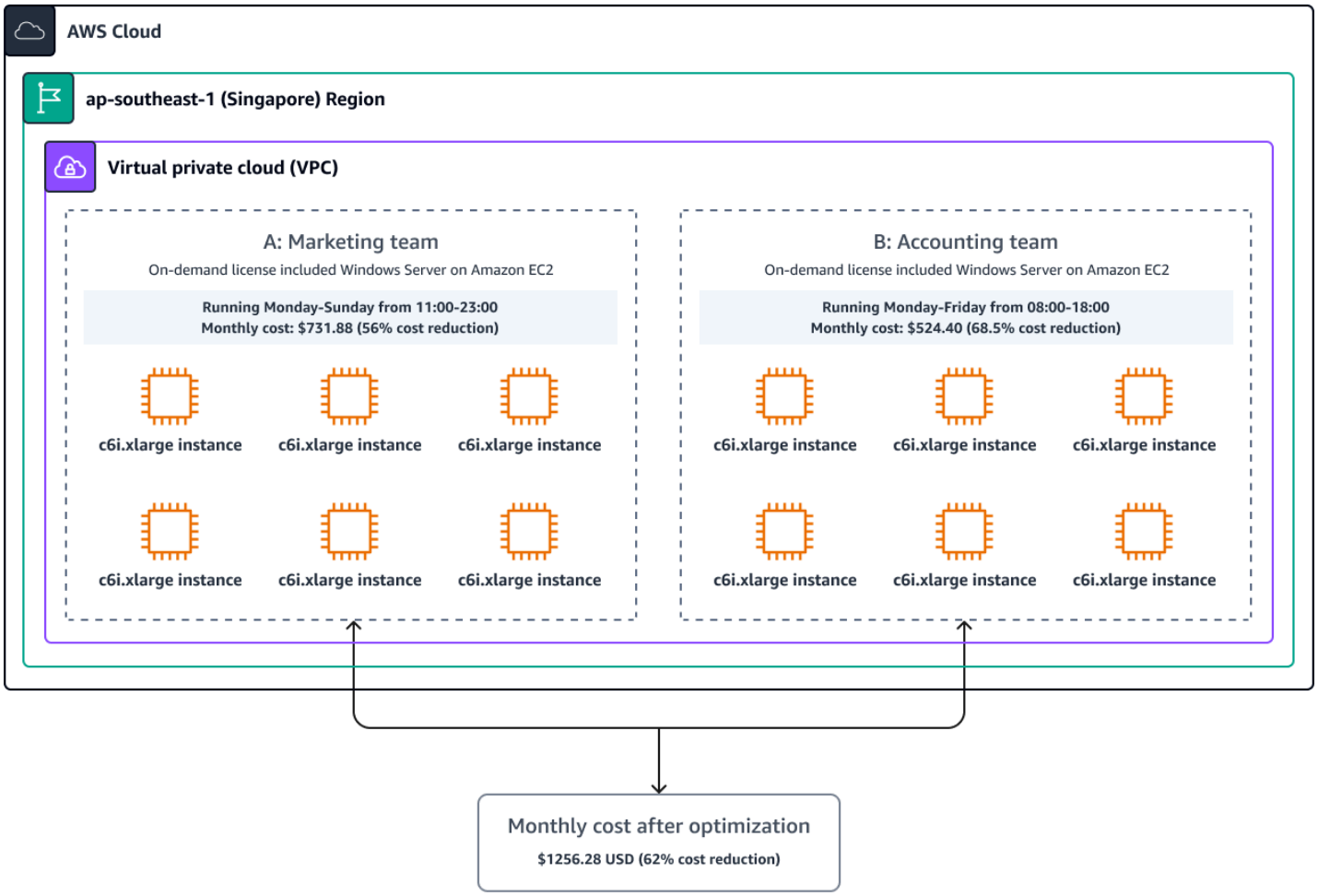
Contoh skenario berikut membantu menggambarkan keuntungan biaya menggunakan Instance Scheduler pada. AWS Dalam skenario ini, sebuah perusahaan ritel besar di Singapura menyebarkan dua lingkungan Windows di Amazon EC2. Lingkungan pertama, yang dikenal sebagai beban kerja A, digunakan oleh tim pemasaran untuk menganalisis transaksi di dalam toko waktu nyata saat toko buka. Lingkungan kedua, yang dikenal sebagai beban kerja B, disediakan untuk tim akuntansi, yang hanya bekerja selama jam kerja reguler. Jadwal operasi kedua lingkungan saat ini (24/7) tidak ideal mengingat pola penggunaan saat ini dan memerlukan pengoptimalan untuk mengurangi biaya operasional perusahaan.

Diagram berikut menunjukkan biaya bulanan sebelum optimasi.



Misalnya, ada 31 hari di bulan Maret, di mana 23 di antaranya adalah hari kerja. Jika tim pemasaran menggunakan Penjadwal Instance AWS dan mengoperasikan instans mereka hanya bila diperlukan (yaitu, untuk 321 jam per bulan, bukan 730 jam per bulan), mereka berpotensi menghemat $932,52 setiap bulan. Ini berarti pengurangan 56 persen dalam biaya operasi. Tim akuntansi dapat mengalami keuntungan yang signifikan juga, dengan waktu penggunaan instans mereka turun dari 730 jam per bulan menjadi 230 jam. Ini menghasilkan pengurangan $1.140, atau 68,5 persen. Perusahaan dapat menghemat total gabungan $2.072.52 per bulan (sama dengan pengurangan 62 persen), atau $24.870.24 per tahun.

Diagram berikut menunjukkan biaya bulanan setelah optimasi.



> **Note**
>
> Harga untuk contoh ini ditentukan dengan menggunakan AWS Kalkulator Hargapada Maret 2023.

Rekomendasi optimisasi biaya

Bagian ini menjelaskan cara menerapkan dan mengonfigurasi Penjadwal Instance AWS berdasarkan skenario contoh yang tercakup dalam bagian skenario pengoptimalan Biaya sebelumnya. Kami menyarankan Anda mengambil langkah-langkah berikut untuk mengoptimalkan biaya Anda dengan menggunakan Penjadwal Instance pada AWS:

1. Luncurkan tumpukan Penjadwal Instance
2. Konfigurasikan periode

3. Konfigurasikan jadwal
4. Contoh tag

Diagram arsitektur berikut menunjukkan apa yang dibuat dalam AWS Cloud oleh stack Scheduler Instance.

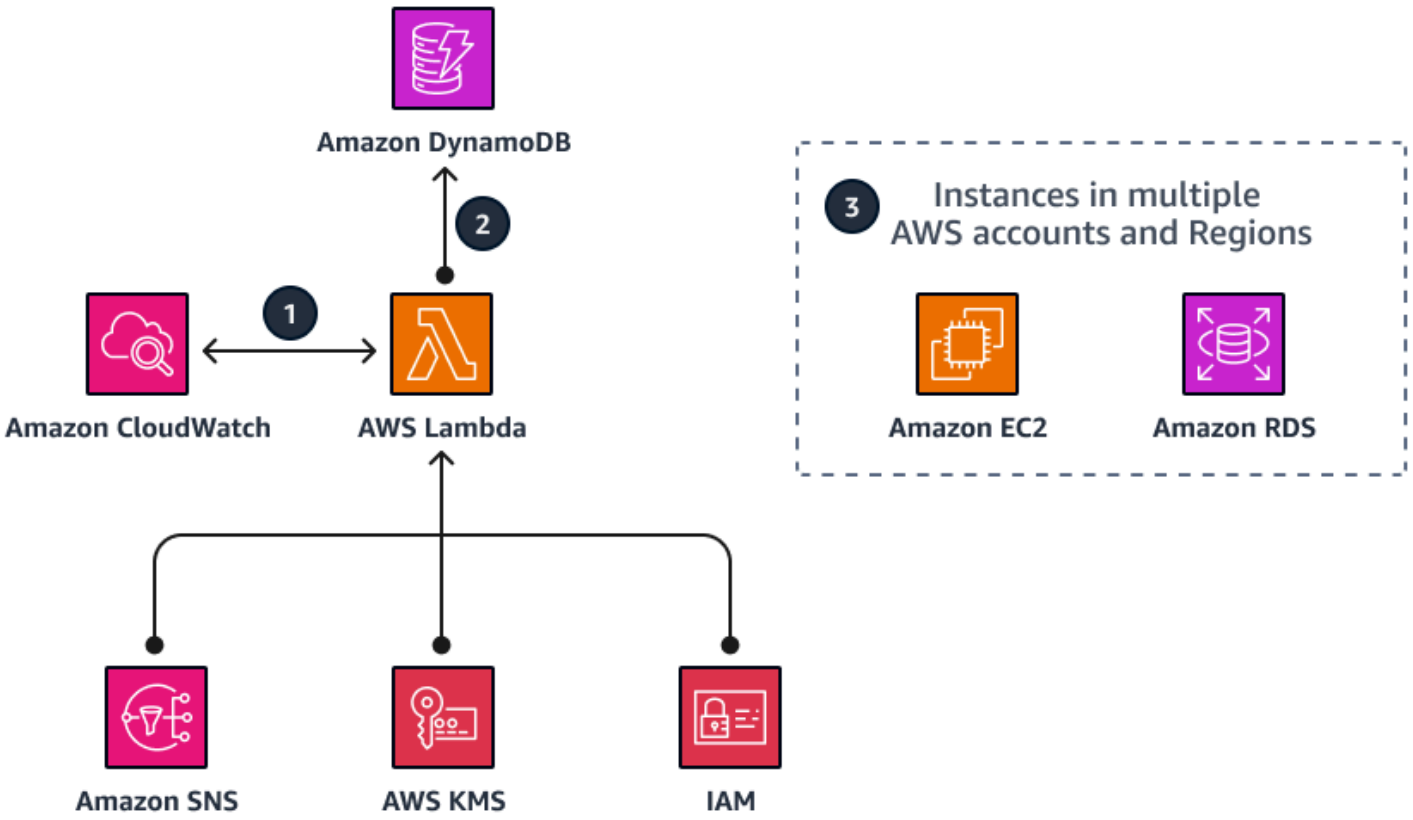


Diagram menunjukkan langkah-langkah alur kerja berikut:

1. AWS CloudFormation Template mengatur CloudWatch acara Amazon pada interval yang Anda tentukan. Acara ini memanggil AWS Lambda fungsi. Selama konfigurasi, Anda menentukan Wilayah AWS dan akun. Anda juga menentukan tag kustom yang AWS digunakan Penjadwal Instance untuk mengaitkan jadwal dengan instans Amazon EC2 yang berlaku, instans Amazon RDS, dan cluster.
2. Nilai konfigurasi jadwal disimpan di Amazon DynamoDB, dan fungsi Lambda mengambilnya setiap kali dijalankan. Anda kemudian dapat menerapkan tag kustom ke instance yang berlaku.
3. Selama konfigurasi awal Penjadwal Instans, Anda menentukan kunci tag untuk mengidentifikasi instans Amazon EC2 dan Amazon RDS yang berlaku. Saat Anda membuat jadwal, nama yang

Anda tentukan akan digunakan sebagai nilai tag yang mengidentifikasi jadwal yang ingin Anda terapkan ke sumber daya yang ditandai.

Luncurkan tumpukan Penjadwal Instance

Bagian ini menunjukkan cara meluncurkan CloudFormation tumpukan untuk Penjadwal Instance. AWS

> **Note**
>
> Anda bertanggung jawab atas biaya yang Layanan AWS digunakan saat menjalankan Penjadwal Instance pada AWS. Mulai Januari 2023, biaya untuk menjalankan solusi ini dengan pengaturan default di `us-east-1` Wilayah adalah sekitar $9,90 per bulan untuk biaya Lambda, atau kurang jika Anda memiliki kredit penggunaan bulanan tingkat gratis Lambda. Untuk informasi selengkapnya, lihat bagian Biaya Penjadwal Instans pada Panduan AWS Implementasi di Perpustakaan AWS Solusi.

Untuk meluncurkan stack scheduler instance, selesaikan langkah-langkah berikut.

1. Masuk ke Konsol Manajemen AWSdan pilih Luncurkan solusi (templat yang dapat diunduh) untuk meluncurkan `instance-scheduler-on-aws.template` CloudFormation templat.

   > **Note**
   >
   > Anda juga dapat mengunduh template sebagai titik awal untuk implementasi Anda sendiri.

2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan Penjadwal Instance di Wilayah lain, gunakan pemilih Wilayah di bilah navigasi konsol.

   > **Note**
   >
   > Contoh ini menggunakan Wilayah Asia Pasifik (Singapura).

3. Pada halaman Buat Tumpukan, di bagian Prasyarat - Siapkan templat, verifikasi bahwa opsi Template siap dipilih. Di bagian Sumber templat, verifikasi bahwa opsi URL Amazon S3 dipilih.
4. Verifikasi bahwa URL templat yang benar ada di kotak teks URL Amazon S3, lalu pilih Berikutnya.

5. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda. Untuk informasi tentang batasan penamaan karakter, lihat Batas IAM dan STS dalam dokumentasi AWS Identity and Access Management (IAM). Nama tumpukan untuk contoh dalam panduan ini disebut`MyInstanceScheduler`.

> **Note**
>
> Nama stack tidak dapat berisi lebih dari 28 karakter.

6. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi seperlunya.
7. Pilih Berikutnya. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Pilih kotak yang mengakui bahwa template akan membuat sumber daya IAM.
9. Pilih Buat untuk men-deploy tumpukan.

Konfigurasikan periode

Setelah Anda menerapkan CloudFormation template, solusi akan membuat tabel DynamoDB yang berisi aturan periode sampel dan jadwal yang dapat Anda gunakan sebagai referensi untuk membuat aturan dan jadwal periode kustom Anda sendiri. Untuk contoh konfigurasi periode, lihat Jadwal sampel dalam Penjadwal Instance pada AWS dokumentasi.

Untuk menyelesaikan langkah skenario ini, Anda harus menghasilkan periode yang sesuai dengan setiap beban kerja dan memenuhi kebutuhan spesifiknya. Contoh:

```
Period 1 (Workload A):
    Name: retail-hours
    Days: Monday to Sunday
    Hours: 1100 - 2300
Period 2 (Workload B):
    Name: office-hours
    Days: Monday to Friday
    Hours: 0800 - 1800
```

Untuk mengonfigurasi periode, selesaikan langkah-langkah berikut:

1. Masuk ke konsol DynamoDB dan pastikan Anda berada di Wilayah yang sama tempat Anda meluncurkan template untuk CloudFormation Penjadwal Instance. AWS

2. Di panel navigasi, pilih Tabel, lalu pilih tabel bernama ConfigTable.
3. Pilih Jelajahi item tabel.
4. Untuk membuat periode jam kantor, pilih periode untuk item jam kantor.
5. Pada halaman Edit item, ubah nilai waktu mulai menjadi 0800 dan waktu akhir menjadi 1800. Biarkan nilai default di tempat untuk hari kerja.

> **Note**
>
> Nilai waktu mulai dan akhir menentukan kapan instance harus dimulai dan dihentikan, sedangkan nilai hari kerja menentukan hari dalam seminggu jadwal ini berlaku (Senin hingga Jumat untuk contoh ini).

6. Pilih Simpan perubahan.
7. Untuk menduplikasi periode jam kantor dan menggunakannya untuk membuat periode baru untuk jam ritel, pilih periode untuk item jam kantor. Kemudian, dari menu Tindakan, pilih item Duplikat.
8. Ubah atribut agar sesuai dengan kebutuhan Anda. Atribut berikut digunakan untuk memenuhi persyaratan skenario contoh:

```
type: period
name: retail-hours
begintime: 11:00
description: Retail hours
endtime: 23:00
weekdays: mon-sun
```

9. Pilih Buat item.
10. Di ConfigTableDynamoDB, identifikasi dua periode yang baru saja Anda buat tercantum dalam daftar item.

Konfigurasikan jadwal

Dalam konteks Penjadwal Instance on AWS, jadwal mengacu pada penerapan satu atau lebih periode dan zona waktu yang relevan. Jadwal ini kemudian ditetapkan ke instance Anda sebagai tag. Bagian ini menunjukkan cara membuat dua jadwal (ditunjukkan di bawah) untuk mengakomodasi pola waktu yang bervariasi dari dua contoh beban kerja, dan kemudian mengaitkan jadwal dengan periode yang Anda buat di bagian sebelumnya.

```
Schedule 1:
    Name: singapore-office-hours
    Period: office-hours
    Timezone: Asia/Singapore
Schedule 2:
    Name: singapore-retail-hours
    Period: retail-hours
    Timezone: Asia/Singapore
```

Untuk membuat dan mengkonfigurasi jadwal, selesaikan langkah-langkah berikut:

1. Masuk ke konsol DynamoDB dan pastikan Anda berada di Wilayah yang sama tempat Anda meluncurkan template untuk CloudFormation Penjadwal Instance. AWS
2. Di panel navigasi, pilih Tabel, lalu pilih tabel bernama ConfigTable.
3. Pilih Jelajahi item tabel.
4. Untuk menduplikasi jadwal jam kantor Inggris dan menggunakannya untuk membuat jadwal baru untuk jam kantor Anda (jam kantor Singapura, untuk contoh ini), pilih jadwal untuk item tersebut uk-office-hours . Kemudian, dari menu Tindakan, pilih item Duplikat.
5. Ubah atribut agar sesuai dengan kebutuhan Anda. Atribut berikut digunakan untuk memenuhi persyaratan skenario contoh:

   ```
   type: schedule
   name: singapore-office-hours
   description: Office hours in Singapore
   periods: office-hours
   timezone: Asia/Singapore
   ```

6. Pilih Buat item.
7. Ulangi langkah 4—6 untuk membuat jadwal jam ritel Singapura menggunakan nilai atribut berikut:

   ```
   type: schedule
   name: singapore-retail-hours
   description: Retail hours in Singapore
   periods: retail-hours
   timezone: Asia/Singapore
   ```

8. Di ConfigTableDynamoDB, identifikasi dua jadwal dan dua periode yang Anda buat.

Contoh tag

Setelah Anda menetapkan jadwal Anda, Anda harus menggunakan tag untuk mengalokasikan jadwal ke instance tertentu yang ingin Anda gunakan. Anda dapat menggunakan editor tag di dalamnya AWS Resource Groupsuntuk menghasilkan dan menetapkan tag ke instans Amazon EC2 Anda.

1. Masuk ke Konsol Manajemen AWSdan pastikan Anda berada di Wilayah yang sama tempat Anda meluncurkan CloudFormation templat sebelumnya.
2. Buka konsol Resource Groups. Di panel navigasi, perluas Penandaan, lalu pilih Editor Tag.
3. Di bagian Temukan sumber daya untuk diberi tag, untuk Wilayah, pilih Wilayah Anda. Untuk jenis Sumber Daya, pilih Amazon EC2 atau Amazon RDS. Skenario ini berfokus pada instans Amazon EC2 dalam beban kerja A. Tim pemasaran menggunakan beban kerja A di Wilayah Singapura. Sumber daya untuk beban kerja ini sudah ditandai dengan kunci Departemen dan nilai Pemasaran. Anda dapat menggunakan tag ini untuk mencari instance.
4. Pilih Cari sumber daya.
5. Pilih instance yang ingin Anda sertakan dalam jadwal dari daftar hasil penelusuran, lalu pilih Kelola tag sumber daya yang dipilih.
6. Di bagian Edit tag dari semua sumber daya yang dipilih, pilih Tambahkan tag untuk menambahkan tag jadwal Penjadwal Instance ke instans EC2 Anda. Anda dapat menggunakan kunci tag dan nilai yang cocok schedulea (sebelumnya dibuat di DynamoDB).
7. Untuk kunci Tag, tambahkan Jadwal. Untuk nilai Tag, masukkan singapore-retail-hours.
8. Pilih Tinjau dan terapkan perubahan tanda.
9. Untuk menerapkan tag ke semua instans EC2 yang Anda pilih, pilih Terapkan perubahan ke semua yang dipilih.
10. Ulangi langkah 3-9 untuk jadwal tambahan yang ingin Anda terapkan.

Validasi hasil

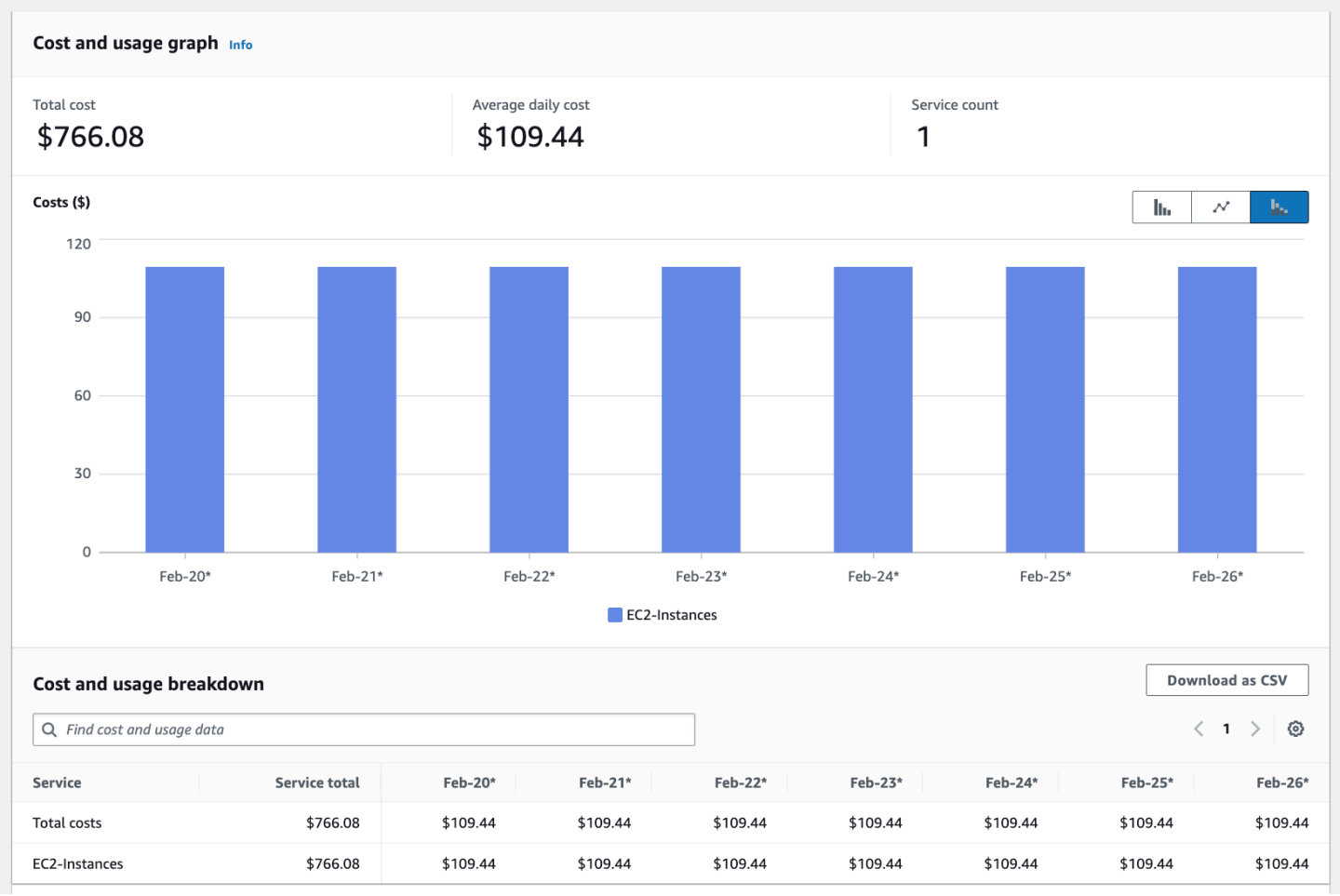
Kami menyarankan Anda menggunakan AWS Cost Exploreruntuk mengukur manfaat biaya menggunakan Penjadwal Instance pada AWS. Anda dapat menggunakan Cost Explorer untuk melakukan hal berikut:

- Lihat dan analisis biaya yang terkait dengan instans EC2 Anda, termasuk instans yang dikelola oleh Penjadwal Instance.

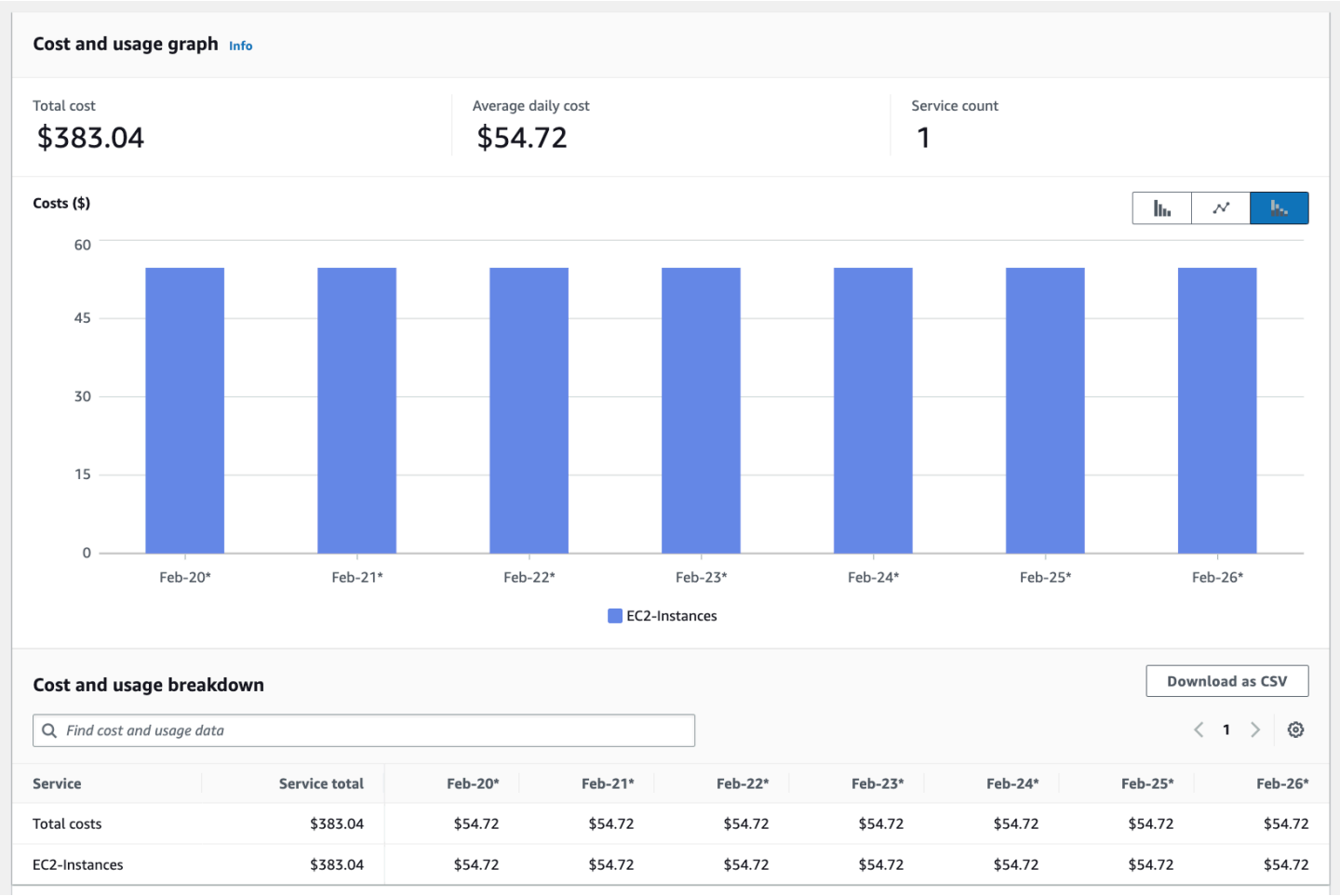
- Filter tampilan Cost Explorer berdasarkan tag sehingga Anda dapat fokus pada beban kerja tertentu dan mendapatkan tampilan terperinci tentang penghematan biaya yang dicapai dengan menggunakan Penjadwal Instance.
- Dapatkan wawasan tentang dampak finansial menggunakan Instance Scheduler.
- Identifikasi peluang untuk optimalisasi biaya lebih lanjut dan buat keputusan berbasis data untuk mengoptimalkan pengeluaran Anda AWS .

Bagan berikut menggambarkan biaya beban kerja operasi A dan beban kerja B selama periode tujuh hari (Senin-Minggu) sebelum optimasi dengan menggunakan Penjadwal Instance.

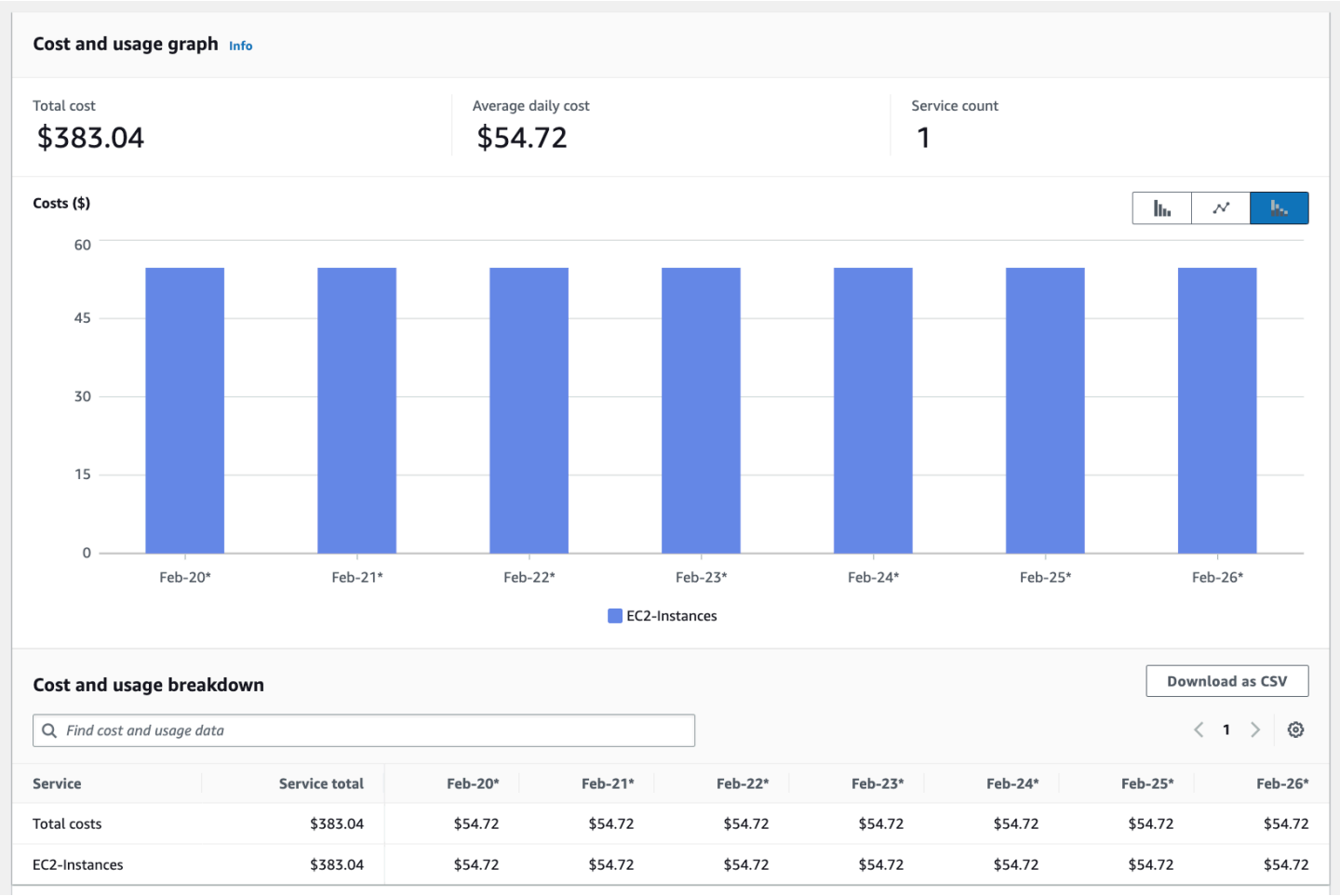
Gabungan total biaya beban kerja A dan B



Beban Kerja A biaya

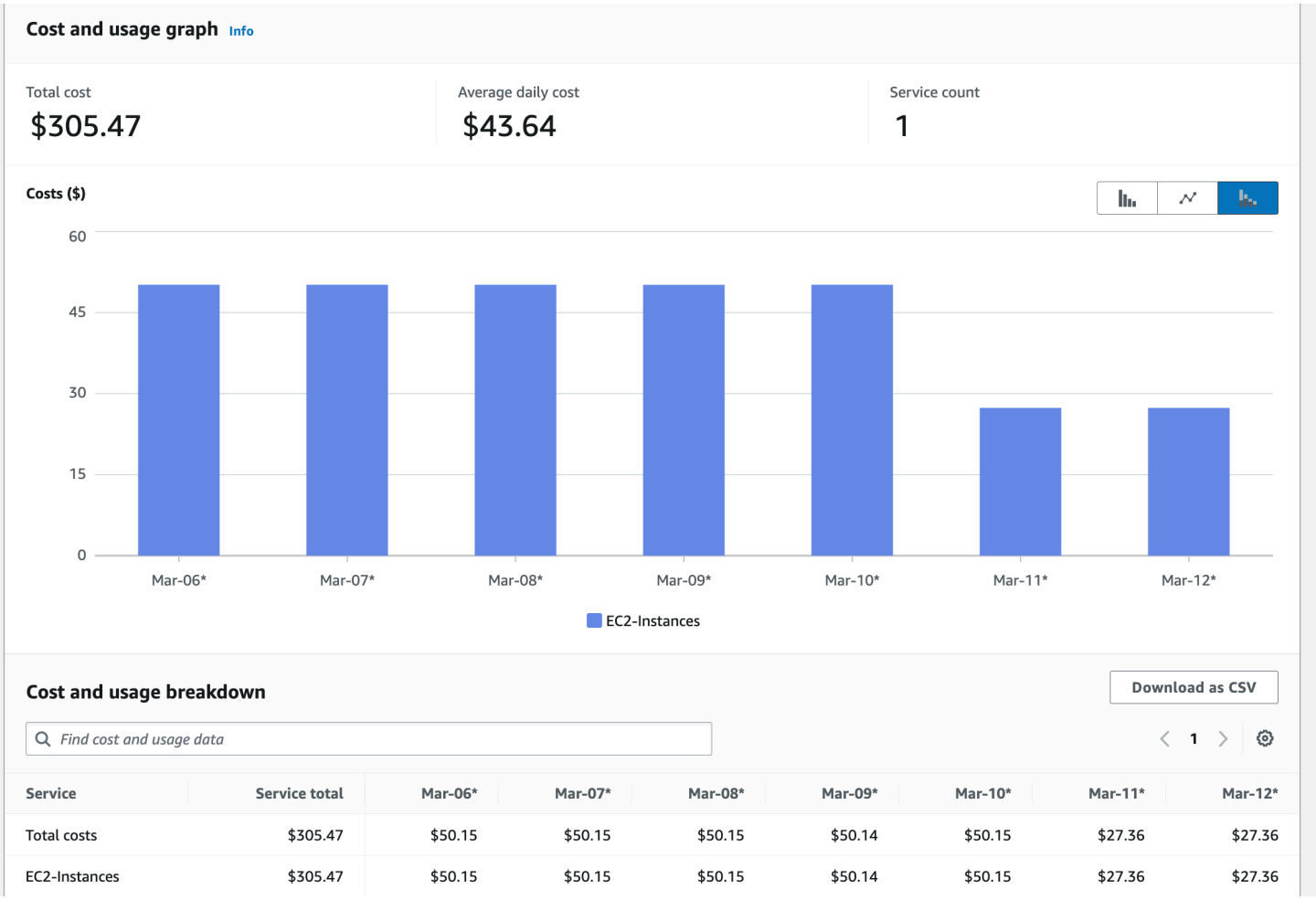


Beban kerja B biaya



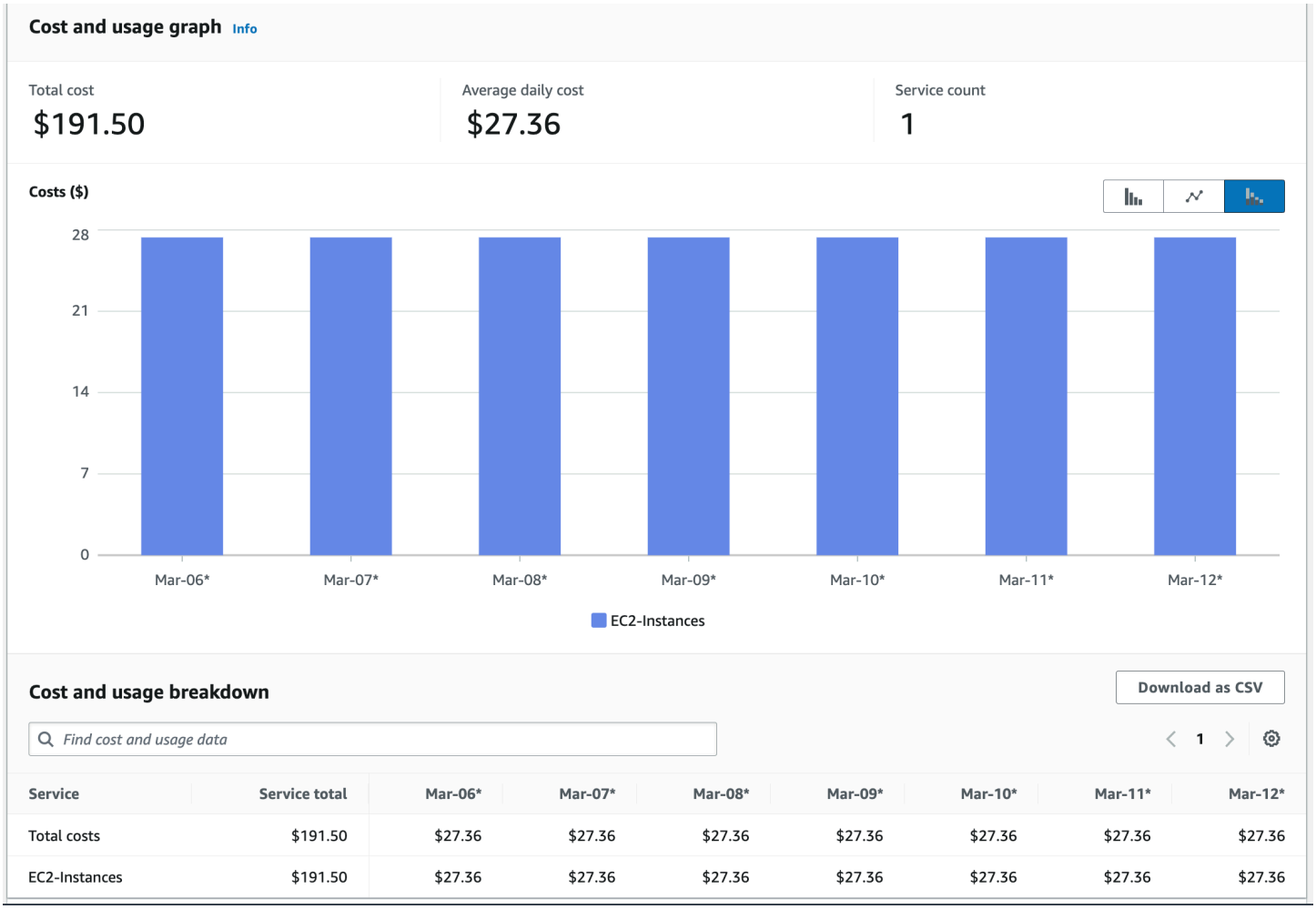
Dalam skenario ini, Cost Explorer menunjukkan pengurangan biaya yang dihasilkan dari penerapan Penjadwal Instance pada. AWS Bagan berikut menggambarkan biaya operasional beban kerja A dan beban kerja B untuk jangka waktu tujuh hari (Senin-Minggu) pasca-optimasi.

Gabungan total biaya beban kerja A dan B



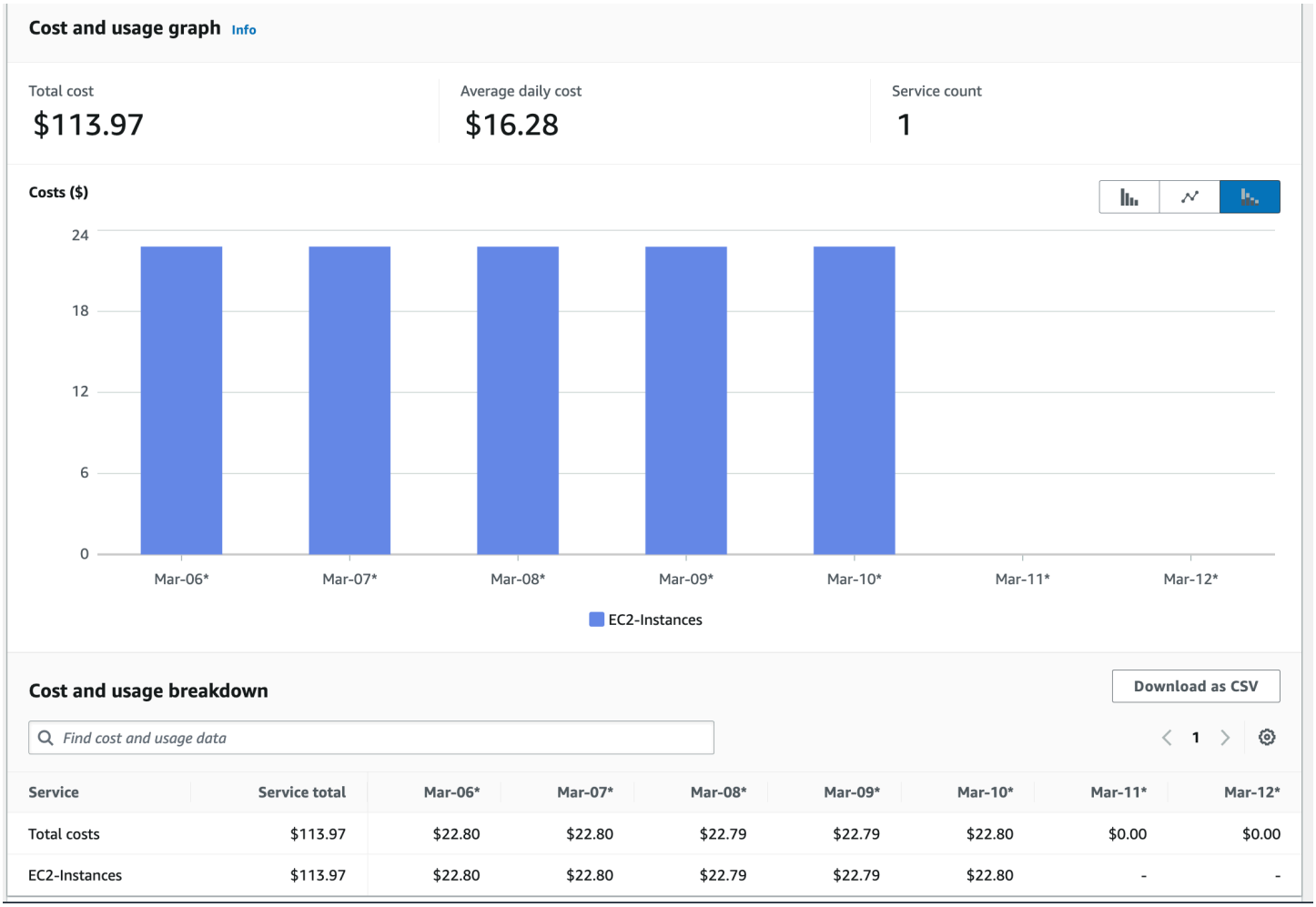
Service	Service total	Mar-06*	Mar-07*	Mar-08*	Mar-09*	Mar-10*	Mar-11*	Mar-12*
Total costs	$305.47	$50.15	$50.15	$50.15	$50.14	$50.15	$27.36	$27.36
EC2-Instances	$305.47	$50.15	$50.15	$50.15	$50.14	$50.15	$27.36	$27.36

Beban Kerja A biaya



Service	Service total	Mar-06*	Mar-07*	Mar-08*	Mar-09*	Mar-10*	Mar-11*	Mar-12*
Total costs	$191.50	$27.36	$27.36	$27.36	$27.36	$27.36	$27.36	$27.36
EC2-Instances	$191.50	$27.36	$27.36	$27.36	$27.36	$27.36	$27.36	$27.36

Beban kerja B biaya



Service	Service total	Mar-06*	Mar-07*	Mar-08*	Mar-09*	Mar-10*	Mar-11*	Mar-12*
Total costs	$113.97	$22.80	$22.80	$22.79	$22.79	$22.80	$0.00	$0.00
EC2-Instances	$113.97	$22.80	$22.80	$22.79	$22.79	$22.80	-	-

Sumber daya tambahan

- Mengotomatiskan memulai dan menghentikan AWS instance (Penjadwal Instance pada dokumentasi) AWS
- Kembali ke Dasar: Menggunakan Penjadwal Instans untuk Mengontrol Biaya Sumber Daya Amazon EC2 dan Amazon RDS () YouTube
- Menandai AWS sumber daya Anda (Panduan Pengguna Tagging AWS Resources)
- Menganalisis biaya Anda dengan AWS Cost Explorer (AWS Manajemen Penagihan dan Biaya dokumentasi)

Beban kerja Windows ukuran yang tepat

Ikhtisar

Ukuran yang tepat adalah salah satu alat hemat biaya yang paling ampuh. AWS menawarkan berbagai metode untuk mengumpulkan informasi ukuran yang tepat, mulai dari meninjau beban kerja potensial dengan menggunakan AWS Optimization and Licensing Assessment (AWS OLA) hingga meninjau beban kerja yang ada dengan menggunakan. AWS Cost Explorer

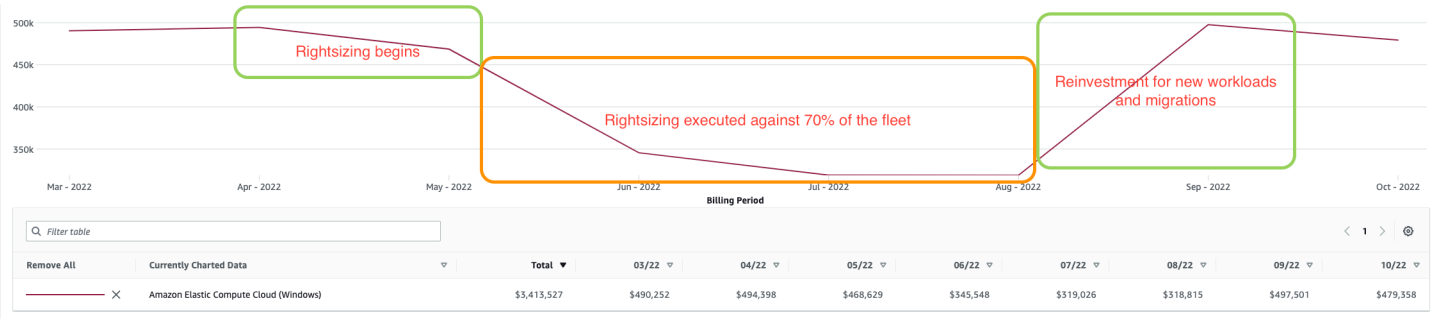
Bagian ini menunjukkan kepada Anda cara menggunakan AWS Compute Optimizeruntuk mengidentifikasi peluang ukuran yang tepat Amazon EC2. Compute Optimizer membantu mencegah penyediaan berlebih dan kekurangan penyediaan untuk jenis sumber daya berikut: AWS

- Jenis instans Amazon Elastic Compute Cloud (Amazon EC2)
- Volume Amazon Elastic Block Store (Amazon EBS)
- Layanan Amazon Elastic Container Service (Amazon ECS) di AWS Fargate
- AWS Lambdafungsi berdasarkan data pemanfaatan yang disediakan oleh Amazon CloudWatch

Skenario pengoptimalan biaya

Mengukur efektivitas ukuran yang tepat dapat menjadi tantangan, karena upaya ukuran yang tepat dapat diarahkan ke aplikasi, tim, atau seluruh organisasi tertentu. Misalnya, pertimbangkan sebuah organisasi yang memigrasikan beberapa ribu instance ke AWS, dengan 90 persen armada mereka terdiri dari beban kerja Windows. Organisasi dapat menggunakan Compute Optimizer untuk menganalisis armada mereka dan menemukan penyediaan berlebihan yang signifikan di seluruh akun mereka dan. Wilayah AWS Kemudian, mereka dapat menggunakan AWS Systems Manager Otomasi untuk mengukur ukuran armada mereka melalui beberapa jendela pemeliharaan. Akibatnya, organisasi berhasil menyesuaikan jenis instans berukuran tepat untuk 70 persen armada mereka dan mencapai penghematan biaya 35 persen.

Dasbor berikut menggambarkan penghematan yang dicapai selama beberapa bulan karena organisasi contoh ini secara strategis menerapkan rekomendasi ukuran yang tepat dari Compute Optimizer. Tujuan mereka adalah untuk mengoperasikan beban kerja mereka yang ada seefisien mungkin untuk melanjutkan migrasi yang macet dari pusat data kolokasi mendekati akhir kontraknya.



Rekomendasi optimisasi biaya

Kami menyarankan Anda mengambil langkah-langkah berikut untuk mengoptimalkan biaya Anda dengan menggunakan Compute Optimizer:

- Aktifkan Compute Optimizer
- Aktifkan pengumpulan metrik Memori untuk node Windows
- Konsumsi rekomendasi Compute Optimizer
- Tandai contoh untuk ukuran yang tepat
- Aktifkan tag alokasi biaya untuk bekerja dengan alat AWS penagihan
- Terapkan rekomendasi ukuran yang tepat dengan AWS Systems Manager Otomasi
- Pertimbangkan metode pengubahan ukuran alternatif
- Tinjau biaya sebelum dan sesudah di Cost Explorer

Aktifkan Compute Optimizer

Anda dapat mengaktifkan Compute Optimizer di tingkat organisasi atau akun tunggal. AWS Organizations Konfigurasi seluruh organisasi menyediakan laporan berkelanjutan untuk instans baru dan yang sudah ada di seluruh armada Anda untuk semua akun anggota. Ini memungkinkan ukuran yang tepat menjadi aktivitas berulang, bukan aktivitas. point-in-time

Tingkat organisasi

Untuk sebagian besar organisasi, cara paling efisien untuk menggunakan Compute Optimizer adalah di tingkat organisasi. Ini memberikan visibilitas multi-akun dan Multi-wilayah ke organisasi Anda dan memusatkan data menjadi satu sumber untuk ditinjau. Untuk mengaktifkan ini di tingkat organisasi, lakukan hal berikut:

1. Masuk ke akun manajemen Organizations Anda dengan peran yang memiliki izin yang diperlukan dan pilih untuk memilih semua akun dalam organisasi ini. Organisasi Anda harus mengaktifkan semua fitur.
2. Setelah mengaktifkan akun manajemen, Anda dapat masuk ke akun, melihat semua akun anggota lainnya, dan menelusuri rekomendasi mereka.

> **Note**
>
> Ini adalah praktik terbaik untuk mengonfigurasi akun administrator yang didelegasikan untuk Compute Optimizer. Ini memungkinkan Anda untuk menggunakan prinsip hak istimewa yang paling sedikit. Dengan begitu, Anda dapat meminimalkan akses ke akun manajemen organisasi sambil tetap menyediakan akses ke layanan di seluruh organisasi.

Tingkat akun tunggal

Jika Anda menargetkan akun dengan biaya tinggi tetapi tidak memiliki akses AWS Organizations, Anda masih dapat mengaktifkan Compute Optimizer untuk akun dan Wilayah tersebut. Untuk mempelajari proses keikutsertaan, lihat Memulai AWS Compute Optimizer dalam dokumentasi Compute Optimizer.

Aktifkan pengumpulan metrik memori untuk node Windows

Metrik memori menyediakan Compute Optimizer dengan metrik penting yang diperlukan untuk membuat rekomendasi ukuran tepat yang terinformasi dengan baik di organisasi Anda. Ini karena analisis CPU, memori, jaringan, dan penyimpanan yang dilakukan sebelum menawarkan rekomendasi.

Untuk meneruskan metrik memori dari instans Windows EC2 ke Compute Optimizer, Anda harus mengaktifkan agen dan mengonfigurasi metrik memori CloudWatch yang akan dikumpulkan setiap 60 detik. Tidak ada biaya tambahan untuk menggunakan metrik memori dengan CloudWatch.

Aktifkan CloudWatch agen dan konfigurasikan metrik memori

Unduh ComputeOptimizefile.yl. Anda dapat menggunakan file ini untuk mengaktifkan pengumpulan memori untuk semua instance di akun Anda. File template menghasilkan komponen-komponen berikut:

- AWS Systems Manager Parameter Store — Ini menyimpan konfigurasi untuk CloudWatch agen yang diperlukan untuk mengumpulkan metrik memori.
- AWS Identity and Access Management Peran (IAM) dengan kebijakan AWS terkelola untuk AWS Systems Manager dilampirkan — Ini untuk dokumen Otomasi Systems Manager.
- AWS Systems Manager dokumen — Ini menginstal dan mengkonfigurasi CloudWatch agen (menggantikan CloudWatch konfigurasi yang ada).
- AWS Systems Manager Asosiasi Manajer Negara - Ini memungkinkan dokumen Systems Manager berjalan di semua instance di akun Anda.

> **Important**
>
> Menjalankan template ini menimpa CloudWatch konfigurasi yang ada pada instance.

Selanjutnya, lakukan hal berikut:

1. Masuk ke Konsol Manajemen AWS dan buka CloudFormation konsol.
2. Di panel navigasi, pilih Stacks (Tumpukan).
3. Pilih Buat tumpukan, lalu pilih Dengan sumber daya yang ada (sumber daya impor).
4. Pilih Berikutnya.
5. Untuk sumber Template, pilih Upload file template.
6. Pilih file, lalu unggah `ComputeOptimize.yml` file.
7. Pilih Berikutnya.
8. Pada halaman Tentukan detail tumpukan, untuk nama Stack, masukkan nama untuk tumpukan Anda, lalu pilih Berikutnya.
9. Pada halaman Identifikasi sumber daya, masukkan nilai pengenal untuk sumber daya yang Anda impor.
10. Pilih Impor sumber daya.
11. Setelah tumpukan digunakan, pilih tab Output untuk menemukan kunci, nilai, dan deskripsi untuk asosiasi Anda.

Pantau kemajuan asosiasi

1. Setelah penyebaran CloudFormation tumpukan selesai, buka konsol Systems Manager.

2. Di panel navigasi, di bagian Manajemen Node, pilih Manajer Negara.
3. Pada halaman Asosiasi, pilih ID asosiasi asosiasi Anda.
4. Pilih tab Riwayat eksekusi.
5. Di kolom Id eksekusi, pilih ID eksekusi asosiasi Anda. Statusnya harus sukses.

Lihat metrik di CloudWatch

Kami menyarankan Anda menunggu setidaknya lima menit hingga metrik terisi CloudWatch.

1. Buka konsol CloudWatch.
2. Di panel navigasi, perluas bagian Metrik, lalu pilih Semua metrik.
3. Konfirmasikan bahwa metrik muncul di bawah CWAgentnamespace.

> **Note**
>
> Untuk menerapkan pengaturan ke instance baru, jalankan kembali asosiasi.

Konsumsi rekomendasi Compute Optimizer

Pertimbangkan contoh yang berfokus pada membuat perubahan ukuran yang tepat dalam satu akun dan satu Wilayah. Dalam contoh ini, Compute Optimizer diaktifkan di tingkat organisasi di semua akun. Perlu diingat bahwa ukuran yang tepat adalah proses yang mengganggu yang dalam banyak kasus dilakukan dengan presisi oleh pemilik aplikasi selama jendela pemeliharaan terjadwal selama beberapa minggu.

Jika Anda menavigasi ke Compute Optimizer dari dalam akun manajemen organisasi (seperti yang ditunjukkan pada langkah-langkah berikut), Anda dapat memilih akun yang ingin Anda selidiki. Dalam contoh ini, ada enam contoh yang berjalan dalam satu akun di `us-east-1` Wilayah. Keenam contoh disediakan secara berlebihan. Tujuannya adalah untuk mengubah ukuran instance berdasarkan rekomendasi dari Compute Optimizer.

Identifikasi instans yang disediakan secara berlebihan dan detail rekomendasi ekspor

1. Masuk ke Konsol Manajemen AWS dan buka konsol Compute Optimizer.
2. Di panel navigasi, pilih Dasbor.

3. Di kotak pencarian di halaman Dashboard, masukkan Region=US East (Virginia N.). Kemudian, masukkan findings=Over-provisioned. Filter ini memungkinkan Anda melihat semua instans yang disediakan secara berlebihan di Wilayah. `us-east-1`
4. Untuk meninjau rekomendasi terperinci untuk instans EC2 yang disediakan secara berlebihan, gulir ke bawah ke kartu instans EC2, lalu pilih Lihat rekomendasi.
5. Pilih Ekspor dan simpan file untuk digunakan di masa mendatang.
6. Untuk bucket S3, masukkan nama bucket Amazon S3 yang Anda inginkan menjadi tujuan untuk file ekspor.

> **Note**
>
> Untuk menyimpan rekomendasi untuk tinjauan masa depan, Anda harus memiliki bucket S3 yang tersedia untuk Compute Optimizer untuk ditulis di setiap Wilayah. Untuk informasi selengkapnya, lihat kebijakan bucket Amazon S3 untuk dokumentasi AWS Compute Optimizer Compute Optimizer.

7. Di bagian Ekspor filter, pilih kotak centang Sertakan rekomendasi untuk semua akun anggota di organisasi.
8. Untuk jenis Sumber Daya, pilih instans EC2.
9. Di bagian Kolom untuk disertakan, pilih kotak centang Pilih semua.
10. Pilih Ekspor.

Pilih instans berdasarkan rekomendasi

Rekomendasi instans didasarkan pada metrik kinerja yang dikumpulkan dan dianalisis oleh Compute Optimizer. Penting untuk mengetahui beban kerja yang berjalan pada instance untuk memastikan bahwa Anda memilih contoh terbaik. Contoh ini mengasumsikan bahwa Anda dapat memilih dari instans Amazon EC2 R6i, R5, dan T3 generasi terbaru. Instans T3 burstable dan memiliki kemampuan bandwidth jaringan yang lebih rendah. Instans R5 dan R6 memiliki biaya per jam yang sama dan hampir identik. Namun, instans R6 memiliki kapasitas bandwidth jaringan yang lebih tinggi, fitur prosesor Intel generasi terbaru, dan menawarkan jejak komputasi yang sama dengan R5. Dalam contoh ini, R6 adalah opsi terbaik untuk memilih untuk mengubah ukuran.

1. Di konsol Compute Optimizer, pilih Rekomendasi untuk instans EC2 dari bilah navigasi. Halaman ini menunjukkan perbandingan jenis instans saat ini dengan opsi yang disarankan untuk menggantinya.

2. Untuk mendapatkan ID instance yang ingin Anda sesuaikan ukurannya, buka konsol Amazon S3 dari akun manajemen. AWS Organizations
3. Di panel navigasi, pilih Bucket, lalu pilih bucket yang Anda gunakan untuk menyimpan hasil ekspor Anda.
4. Pada tab Objek, pilih file ekspor Anda dari daftar objek, lalu pilih Unduh.
5. Untuk mengekstrak informasi instance dari file, Anda dapat menggunakan tombol Teks ke Kolom pada tab Data di Microsoft Excel.

> **Note**
>
> Instance IDs direpresentasikan sebagai Amazon Resource Names (ARNs). Pastikan untuk mengatur pembatas ke "/" dan ekstrak ID instance. Atau, Anda dapat menulis skrip atau menggunakan lingkungan pengembangan terintegrasi (IDE) untuk memangkas ARN.

6. Di Excel, filter kolom pencarian untuk hanya menampilkan instance OVER_PROVISIONED. Ini adalah contoh yang Anda targetkan untuk ukuran yang tepat.
7. Simpan instance IDs dalam editor teks untuk memudahkan akses nanti.

Tandai contoh untuk ukuran yang tepat

Menandai beban kerja Anda adalah alat yang ampuh untuk mengatur sumber daya Anda. AWS Tag memungkinkan Anda untuk mendapatkan visibilitas halus ke dalam biaya dan memfasilitasi tolak bayar. Untuk informasi selengkapnya tentang strategi dan metode untuk menambahkan tag ke AWS sumber daya, lihat AWS Whitepaper Best Practices for AWS Tagging Resources. Untuk contoh ini, Anda dapat menggunakan Editor AWS Tag untuk membuat penyesuaian penandaan di seluruh instance yang disediakan berlebihan yang ingin Anda targetkan untuk mengubah ukuran selama jendela pemeliharaan. Anda juga dapat menggunakan tag ini untuk melihat biaya sebelum dan sesudah perubahan.

1. Masuk ke Konsol Manajemen AWS dan buka AWS Resource Groups konsol untuk akun yang berisi instance yang ditargetkan untuk mengubah ukuran.
2. Pada bilah navigasi, di bagian Penandaan, pilih Editor Tag.
3. Untuk Wilayah, pilih Wilayah target Anda.
4. Untuk jenis Sumber Daya, pilih AWS::EC2::Instance.
5. Pilih Cari sumber daya.

6. Pada halaman hasil pencarian sumber daya, pilih semua instance yang ingin Anda perbaiki ukurannya, lalu pilih Kelola tag sumber daya yang dipilih.
7. Pilih Tambahkan tanda.
8. Untuk tombol Tag, masukkan Rightsizing. Untuk nilai Tag, masukkan diaktifkan. Kemudian, pilih Tinjau dan terapkan perubahan tag.

> **Note**
>
> Anda dapat menyertakan metadata tambahan seperti Tim atau Unit Bisnis untuk membantu memfilter nanti di Cost Explorer.

Setelah membuat dan menerapkan tag yang ditentukan pengguna ke sumber daya Anda, mungkin diperlukan waktu hingga 24 jam agar tag muncul di halaman tag alokasi biaya Anda untuk aktivasi. Setelah Anda memilih tag Anda untuk aktivasi, itu bisa memakan waktu 24 jam lagi untuk tag menjadi aktif.

Untuk pengguna tingkat lanjut, Anda dapat menggunakan AWS CloudShelldalam akun target dan Wilayah untuk menandai beberapa instance. Contoh:

```
bash
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="type-m5"
# Get a list of instance IDs
INSTANCE_IDS=$(aws ec2 describe-instances –query
 "Reservations[].Instances[].InstanceId" –output text)
# Loop through each instance ID and add the tag
for INSTANCE_ID in $INSTANCE_IDS; do
 aws ec2 create-tags –resources $INSTANCE_ID –tags Key=$TAG_KEY,Value=$TAG_VALUE
done
```

Aktifkan tag alokasi biaya untuk bekerja dengan alat AWS penagihan

Sebaiknya aktifkan tag alokasi biaya yang ditentukan pengguna. Ini memungkinkan tag Rightsizing dikenali dan difilter di alat AWS penagihan (misalnya, Cost Explorer dan). AWS Cost and Usage Report Jika Anda tidak mengaktifkan ini, opsi pemfilteran tag dan data tidak akan tersedia. Untuk

informasi tentang penggunaan tag alokasi biaya, lihat Mengaktifkan tag alokasi biaya yang ditentukan pengguna dalam dokumentasi. AWS Manajemen Penagihan dan Biaya

1. Masuk ke Konsol Manajemen AWS dan buka AWS Billing konsol.
2. Pada panel navigasi, di bagian Penagihan, pilih Tag alokasi biaya.
3. Pada tab Tag alokasi biaya yang ditentukan pengguna, masukkan Rightsizing.
4. Pilih tombol tag Rightsizing, lalu pilih Activate.

Setelah 24 jam, tag akan muncul di Cost Explorer.

Menerapkan rekomendasi ukuran yang tepat dengan Systems Manager Automation

Mengubah ukuran adalah skenario yang membutuhkan instance untuk dihentikan dan dimulai. Dalam skenario ini, Anda mungkin perlu menangani gangguan ini di jendela pemeliharaan dan membutuhkan tim yang berbeda untuk menangani pengubahan ukuran mereka sendiri. Sebelum mengubah jenis instans, tinjau Pertimbangan untuk jenis instans yang kompatibel di dokumentasi Amazon EC2.

Contoh langkah di bagian ini menerapkan rekomendasi ukuran yang tepat per akun dan Wilayah dengan menggunakan dokumen Otomasi Systems Manager yang disebut AWS- ResizeInstance. Pendekatan ini khas untuk sebagian besar organisasi karena sebagian besar organisasi memerlukan jenis instance yang berbeda untuk tujuan yang berbeda. Anda juga dapat menggunakan dokumen `AWS-ResizeInstance` otomatisasi yang sama untuk menargetkan penerapan tunggal dan multi-akun.

1. Masuk ke Konsol Manajemen AWS dan buka konsol Systems Manager.
2. Pada panel navigasi, di bagian Sumber Daya Bersama, pilih Dokumen.
3. Di bilah pencarian, masukkan AWS- ResizeInstance, lalu pilih AWS- ResizeInstance dari hasil pencarian.
4. Pilih Eksekusi otomatisasi.
5. Pada halaman runbook Execute Automation, pilih Eksekusi sederhana.
6. Di bagian Parameter input, masukkan InstanceIddan InstanceType. Simpan sisa nilai default.
7. Pilih Execute, dan kemudian tunggu otomatisasi untuk melalui langkah-langkah untuk mengubah jenis instance.

Pertimbangkan metode pengubahan ukuran alternatif

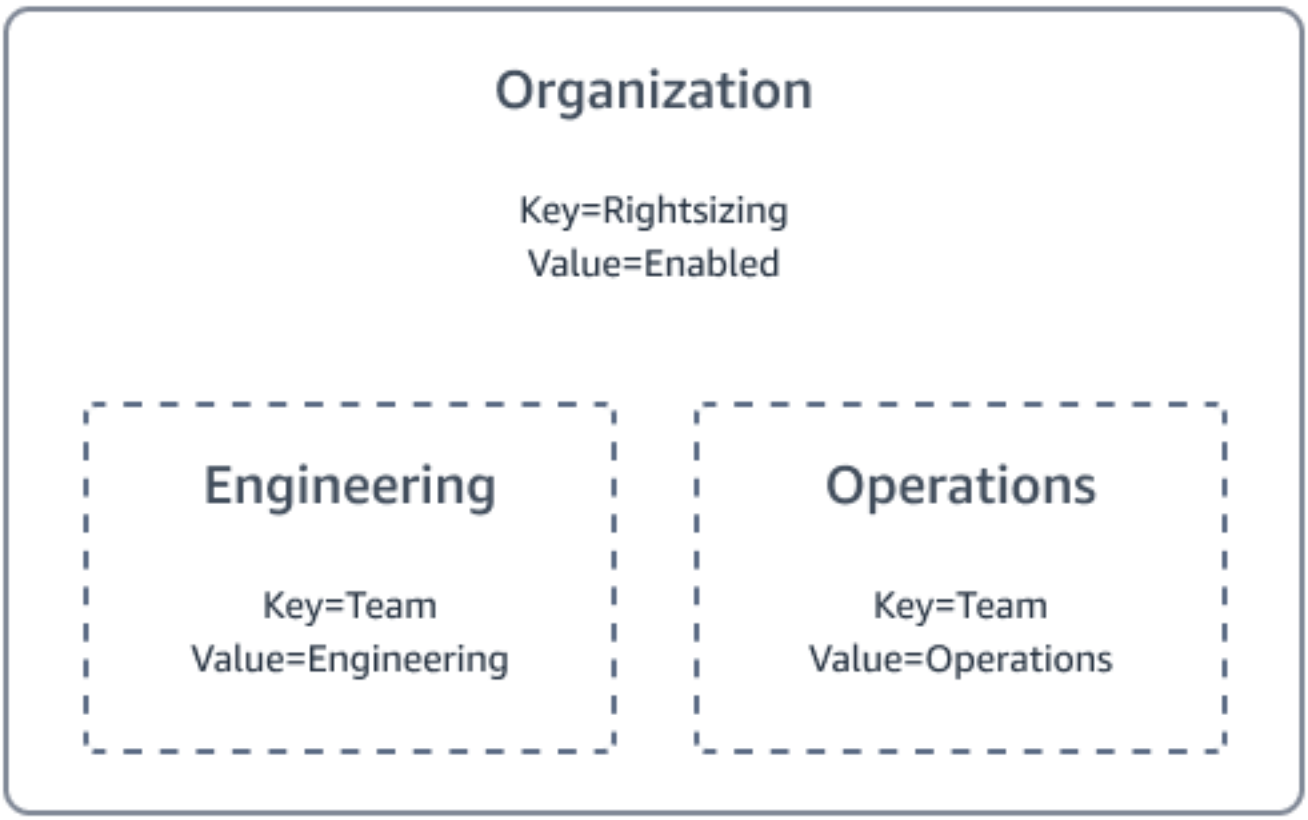
Jika Anda menggunakan template peluncuran untuk menyebarkan instance, Anda dapat memperbarui template peluncuran dengan jenis instance berukuran tepat, lalu melakukan penyegaran instance untuk mengganti instance dengan versi berukuran tepat.

Jika Anda berencana untuk menggunakan proses ukuran yang tepat di beberapa akun dan Wilayah, Anda harus membuat dokumen Otomasi Systems Manager kustom. Dokumen ini memungkinkan Anda untuk memasukkan beberapa instance sebagai parameter dan instance target yang pindah ke jenis instance tujuan yang sama (misalnya, semua instance beralih ke t3a.medium, terlepas dari jenis instance sumber).

Tinjau biaya sebelum dan sesudah di Cost Explorer

Setelah Anda mengukur sumber daya dengan benar, Anda dapat menggunakan Cost Explorer untuk menampilkan sebelum dan sesudah biaya dengan menggunakan tag Rightsizing. Ingatlah bahwa Anda dapat menggunakan tag sumber daya untuk melacak biaya. Dengan menggunakan beberapa lapisan tag, Anda dapat mencapai visibilitas granular ke dalam biaya Anda. Dalam contoh yang dibahas dalam panduan ini, tag Rightsizing digunakan untuk menerapkan tag generik ke semua instance yang ditargetkan. Kemudian, tag tim digunakan untuk mengatur sumber daya lebih lanjut. Langkah selanjutnya adalah memperkenalkan tag aplikasi untuk lebih menunjukkan dampak biaya untuk mengoperasikan aplikasi tertentu.

Diagram berikut menunjukkan struktur tag untuk sebuah organisasi.



Pertimbangkan contoh bisnis yang tepat ukuran server web produksi yang dimiliki oleh tim Operasi. Di Cost Explorer, tag Rightsizing disetel ke diaktifkan, dan tag Team diatur ke operasi. Dalam contoh ini, upaya ukuran yang tepat mengurangi biaya operasi dari 0,89 sen menjadi 0,28 sen per jam. Dengan asumsi 744 jam per bulan, biaya tahunan sebelum ukuran yang tepat adalah $7.945.92. Setelah ukuran yang tepat, biaya tahunan turun menjadi $2.499.84. Ini berarti penurunan 68,5 persen dalam biaya beban kerja tahunan. Bayangkan dampak dari ini di seluruh organisasi besar. Perlu diingat, ini dilakukan di lingkungan sampel dan instance sebagian besar menganggur. Dalam lingkungan produksi, Anda dapat melihat penghematan antara 10—35 persen.

Sekarang, pertimbangkan dampak ukuran yang tepat dari host benteng produksi yang dimiliki oleh tim Teknik. Di Cost Explorer, tag Rightsizing disetel ke diaktifkan, dan tag Team diatur ke engineering. Dalam contoh ini, upaya ukuran yang tepat mengurangi biaya dari 0,75 sen menjadi 0,44 sen per jam. Dengan asumsi 744 jam per bulan, biaya tahunan sebelum ukuran yang tepat adalah $6,696,00. Setelah ukuran yang tepat, biaya tahunan turun menjadi $3,928,32.

Jika Anda menggunakan beberapa tag, Anda dapat memfilter data ke detail biaya granular. Dalam contoh ini, tag Tim mengurangi kebisingan sehingga Anda dapat melihat dampak di tingkat tim. Karena tag Rightsizing diaktifkan, Anda juga dapat memfilter untuk setiap instance yang memiliki tag tersebut dengan nilai diaktifkan atau tidak ada nilai yang ada. Ini dapat memberikan pandangan

global tentang upaya ukuran yang tepat Anda, terutama ketika dilihat di akun manajemen (pembayar) di tingkat Cost Explorer. Tampilan ini memungkinkan Anda untuk melihat semua akun dan instance.

Pertimbangkan contoh di tingkat akun tunggal di mana tag Rightsizing disetel ke diaktifkan. Biaya operasi turun dari $1,64 per jam menjadi $0,72 sen per jam. Dengan asumsi 744 jam per bulan, biaya tahunan sebelum ukuran yang tepat adalah $14.641.92. Setelah ukuran yang tepat, biaya tahunan turun menjadi $6.428,16. Ini berarti penurunan 56 persen dalam biaya komputasi untuk akun ini.

Sebelum memulai perjalanan ukuran yang tepat, pertimbangkan hal berikut:

- AWS menawarkan banyak pilihan untuk pengurangan biaya. Ini termasuk AWS OLA, tempat AWS meninjau instans lokal Anda sebelum pindah ke. AWS AWS OLA juga memberi Anda rekomendasi ukuran yang tepat dan panduan lisensi.
- Selesaikan semua ukuran yang tepat sebelum membeli Savings Plans. Ini dapat membantu Anda menghindari pembelian berlebihan pada komitmen Savings Plans Anda.

Rekomendasi

Kami merekomendasikan langkah-langkah berikut:

1. Tinjau lanskap Anda yang ada dan pertimbangkan untuk mengonversi volume Amazon EBS gp2 menjadi volume gp3.
2. Tinjau Savings Plans.

Sumber daya tambahan

- AWS Compute Optimizer(AWS dokumentasi)
- Praktik Terbaik untuk Menandai AWS Sumber Daya (AWS Whitepaper)
- Cara mengumpulkan data dari AWS Compute Optimizer dan AWS Trusted Advisor di seluruh AWS Organizations (YouTube)
- Mengoptimalkan kinerja dan mengurangi biaya lisensi: Memanfaatkan instans SQL Server Amazon AWS Compute Optimizer EC2 (Microsoft Workloads di blog) AWS

Pilih jenis instans yang tepat untuk beban kerja Windows

Ikhtisar

Perbedaan signifikan antara beban kerja yang beroperasi di cloud dibandingkan dengan lingkungan lokal adalah praktik penyediaan berlebih. Saat membeli perangkat keras fisik untuk penggunaan lokal, Anda membuat pengeluaran modal yang diharapkan berlangsung selama durasi yang telah ditentukan, biasanya 3-5 tahun. Untuk mengakomodasi pertumbuhan yang diantisipasi selama masa pakai perangkat keras, perangkat keras diperoleh dengan lebih banyak sumber daya daripada yang dibutuhkan beban kerja Anda saat ini. Akibatnya, perangkat keras fisik sering disediakan secara berlebihan jauh melampaui kebutuhan beban kerja Anda yang sebenarnya.

Teknologi mesin virtual (VM) muncul sebagai sarana yang efektif untuk memanfaatkan sumber daya perangkat keras surplus. Administrator menyediakan secara berlebihan VMs dengan v CPUs dan RAM, memungkinkan hypervisor untuk mengelola penggunaan sumber daya fisik antara server sibuk dan idle dengan mengalokasikan sumber daya yang tidak digunakan untuk setiap VM. Saat mengelola VMs, sumber daya vCPU dan RAM yang dialokasikan untuk setiap VM berfungsi lebih sebagai gubernur sumber daya daripada indikator penggunaan aktual. Alokasi berlebihan sumber daya VM dapat dengan mudah melebihi tiga kali sumber daya komputasi yang tersedia.

Amazon Elastic Compute Cloud (Amazon EC2) menahan diri dari VMs penyediaan berlebihan pada perangkat keras yang mendasarinya, karena tidak perlu. Cloud computing adalah biaya operasional, bukan biaya modal, dan Anda hanya membayar untuk apa yang Anda gunakan. Jika beban kerja Anda membutuhkan lebih banyak sumber daya di masa depan, sediakan saat Anda benar-benar membutuhkannya, daripada melakukannya terlebih dahulu.

Ada ratusan opsi untuk memilih jenis instans Amazon EC2 yang tepat. Jika Anda berencana untuk memigrasikan beban kerja Windows ke cloud, AWS tawarkan AWS OLA untuk membantu Anda lebih memahami beban kerja Anda saat ini dan memberikan contoh kinerjanya. AWS Analisis AWS OLA bertujuan untuk mencocokkan jenis dan ukuran instans EC2 yang sesuai dengan penggunaan lokal Anda yang sebenarnya.

Jika Anda sudah memiliki beban kerja yang berjalan di Amazon EC2 dan mencari strategi pengoptimalan biaya, bagian panduan ini membantu Anda mengidentifikasi perbedaan antara instans Amazon EC2 dan penerapannya pada beban kerja Windows biasa.

Rekomendasi optimisasi biaya

Untuk mengoptimalkan biaya jenis instans EC2 Anda, kami sarankan Anda melakukan hal berikut:

- Pilih keluarga instans yang tepat untuk beban kerja Anda
- Memahami variasi harga antara arsitektur prosesor
- Memahami perbedaan harga terhadap kinerja di seluruh generasi EC2
- Migrasi ke instance yang lebih baru
- Gunakan instance burstable

Pilih keluarga instans yang tepat untuk beban kerja Anda

Penting untuk memilih keluarga contoh yang tepat untuk beban kerja Anda.

Instans Amazon EC2 dibagi menjadi beberapa kelompok berikut:

- Tujuan umum
- Komputasi yang dioptimalkan
- Memori yang dioptimalkan
- Komputasi yang dipercepat
- Penyimpanan yang dioptimalkan
- HPC dioptimalkan

Sebagian besar beban kerja Windows masuk ke dalam kategori berikut:

- Tujuan umum
- Komputasi yang dioptimalkan
- Memori yang dioptimalkan

Untuk menyederhanakan ini lebih jauh, pertimbangkan instance EC2 dasar di setiap kategori:

- Komputasi dioptimalkan - C6i
- Tujuan umum - M6i
- Memori dioptimalkan - R6i

Instans EC2 generasi sebelumnya menunjukkan perbedaan kecil dalam jenis prosesor. Misalnya, instans yang dioptimalkan komputasi C5 memiliki prosesor yang lebih cepat daripada instans tujuan umum M5 atau instans yang dioptimalkan untuk memori R5. Instans EC2 generasi terbaru (C6i, M6i,

R6i, C6a, M6a, dan R6a) semuanya menggunakan prosesor yang sama di seluruh keluarga instans. Karena prosesor konsisten di antara contoh generasi terbaru, perbedaan harga antara keluarga instans sekarang lebih bergantung pada jumlah RAM. Semakin banyak RAM yang dimiliki sebuah instance, semakin mahal harganya.

Contoh berikut menggambarkan harga per jam untuk instans 4 vCPU berbasis Intel yang berjalan di Wilayah. `us-east-1`

Instans	v CPUs	RAM	Harga per jam
c6i.xlarge	4	8	$0,17
m6i.xlarge	4	16	$0,19
r6i.xlarge	4	32	$0,25

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Contoh yang bisa meledak

Meskipun merupakan praktik terbaik dalam komputasi awan untuk mematikan sumber daya komputasi yang tidak digunakan untuk menghindari biaya, tidak semua beban kerja dapat dimatikan dan dinyalakan setiap kali dibutuhkan. Beberapa beban kerja tetap menganggur untuk waktu yang lama tetapi harus dapat diakses 24 jam sehari.

Burstable instance (T3) menawarkan cara untuk mempertahankan beban kerja runcing atau pemanfaatan rendah secara online sepanjang hari sambil tetap menjaga biaya komputasi tetap rendah. Instans EC2 burstable memiliki jumlah maksimum sumber daya vCPU yang dapat digunakan instans untuk periode singkat. Contoh ini menggunakan sistem berdasarkan kredit CPU burstable. Kredit ini diakumulasikan selama periode idle sepanjang hari. Instans burstable menawarkan vCPU-to-RAM rasio yang bervariasi, menjadikannya alternatif untuk menghitung instance yang dioptimalkan dalam beberapa kasus dan untuk contoh tujuan umum lainnya di kasus lain.

Contoh berikut menggambarkan harga per jam untuk instans T3 (yaitu, instans burstable) yang berjalan di Wilayah. `us-east-1`

Instans	v CPUs	RAM (GB)	Harga per jam
t3.nano	2	0,5	$0.0052
t3.mikro	2	1	$0,0104
t3.small	2	2	$0,0208
t3.medium	2	4	$0,0416
t3.large	2	8	$0,0832
t3.xlarge	4	16	$0,1664
t3.2xlarge	8	32	$0,3328

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Memahami variasi harga antara arsitektur prosesor

Prosesor Intel telah menjadi standar untuk instans EC2 sejak awal. Generasi awal instans EC2, seperti C5, M5, dan R5, tidak menunjukkan Intel sebagai arsitektur prosesor (karena itu adalah default). Generasi baru dari instans EC2, seperti C6i, M6i, dan R6i, menyertakan “i” untuk menunjukkan penggunaan prosesor Intel.

Perubahan anotasi arsitektur prosesor disebabkan oleh pengenalan opsi prosesor tambahan. Prosesor yang paling sebanding dengan Intel adalah AMD (dilambangkan dengan “a”). Prosesor AMD EPYC menggunakan arsitektur x86 yang sama dan menawarkan kinerja yang mirip dengan prosesor Intel tetapi dengan harga lebih rendah. Seperti yang ditunjukkan dalam contoh harga berikut, instans AMD EC2 memberikan diskon sekitar 10 persen untuk biaya komputasi dibandingkan dengan rekan-rekan Intel mereka.

Contoh Intel	Harga per jam	contoh AMD	Harga	% perbedaan
c6i.xlarge	$0,17	c6a.xlarge	$0,153	10%

Contoh Intel	Harga per jam	contoh AMD	Harga	% perbedaan
m6i.xlarge	$0,192	m6a.xlarge	$0,1728	10%
r6i.xlarge	$0,252	r6a.xlarge	$0,2268	10%

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Opsi arsitektur prosesor utama ketiga adalah prosesor AWS Graviton (dilambangkan dengan “g”) pada instans EC2. Dirancang oleh AWS, prosesor Graviton menawarkan kinerja harga terbaik di Amazon EC2. Tidak hanya prosesor Graviton saat ini 20 persen lebih murah daripada rekan-rekan Intel mereka, tetapi mereka juga memberikan peningkatan kinerja 20 persen atau lebih besar. Prosesor Graviton generasi berikutnya diharapkan dapat memperluas perbedaan kinerja ini, dengan pengujian menunjukkan peningkatan kinerja tambahan 25 persen.

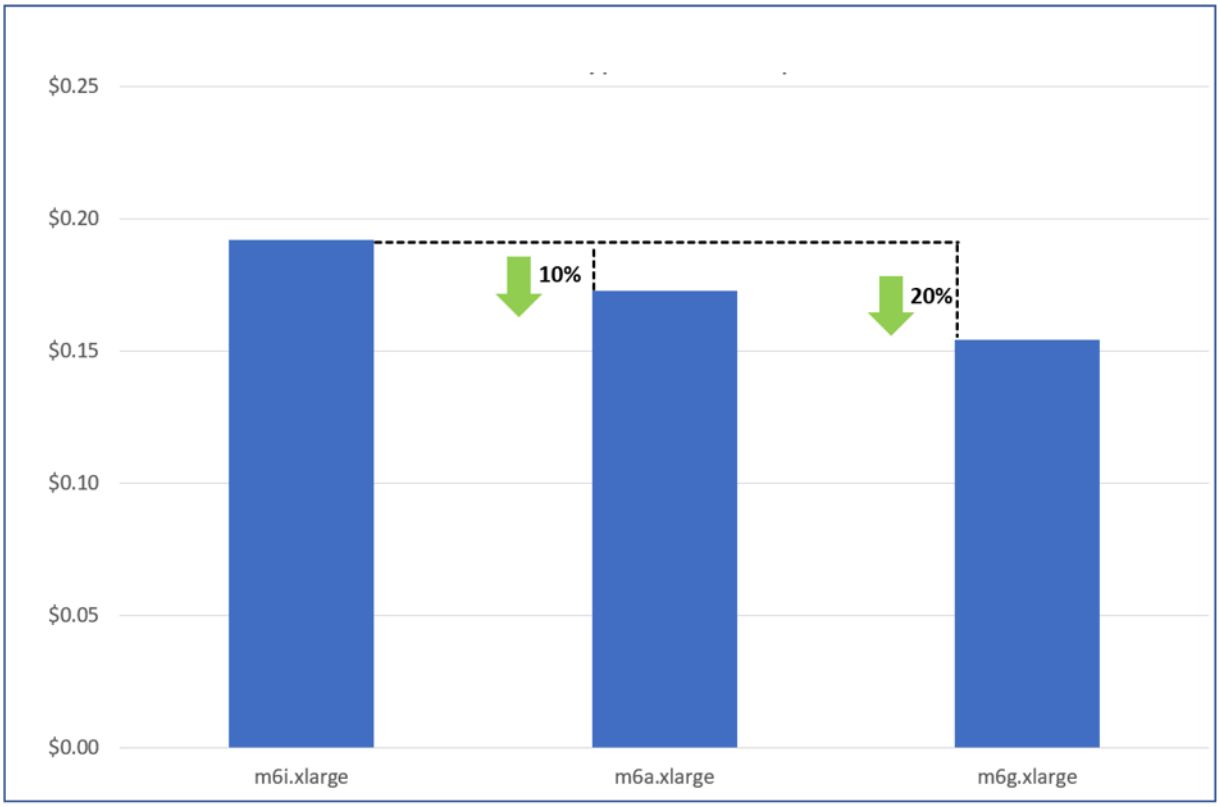
Windows Server tidak dapat berjalan pada prosesor Graviton, yang didasarkan pada arsitektur ARM. Bahkan, Windows Server hanya beroperasi pada prosesor x86. Meskipun Anda tidak dapat mencapai peningkatan kinerja harga 40 persen dengan menggunakan instance berbasis Graviton untuk Windows Server, Anda masih dapat menggunakan prosesor Graviton dengan beban kerja Microsoft tertentu. Misalnya, versi yang lebih baru dari .NET dapat berjalan di Linux. Itu berarti beban kerja ini dapat menggunakan prosesor ARM dan mendapatkan manfaat dari instans Graviton EC2 yang lebih cepat dan lebih terjangkau.

Contoh berikut menggambarkan harga per jam untuk instance Graviton yang berjalan di Wilayah. `us-east-1`

Contoh Intel	Harga per jam	Contoh Graviton	Harga per jam	% perbedaan
c6i.xlarge	$0,17	c6g.xlarge	$0,136	20%
m6i.xlarge	$0,192	m6g.xlarge	$0,154	20%
r6i.xlarge	$0,252	r6g.xlarge	$0,2016	20%

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Bagan berikut membandingkan harga instans seri M.



Memahami perbedaan kinerja harga di seluruh generasi EC2

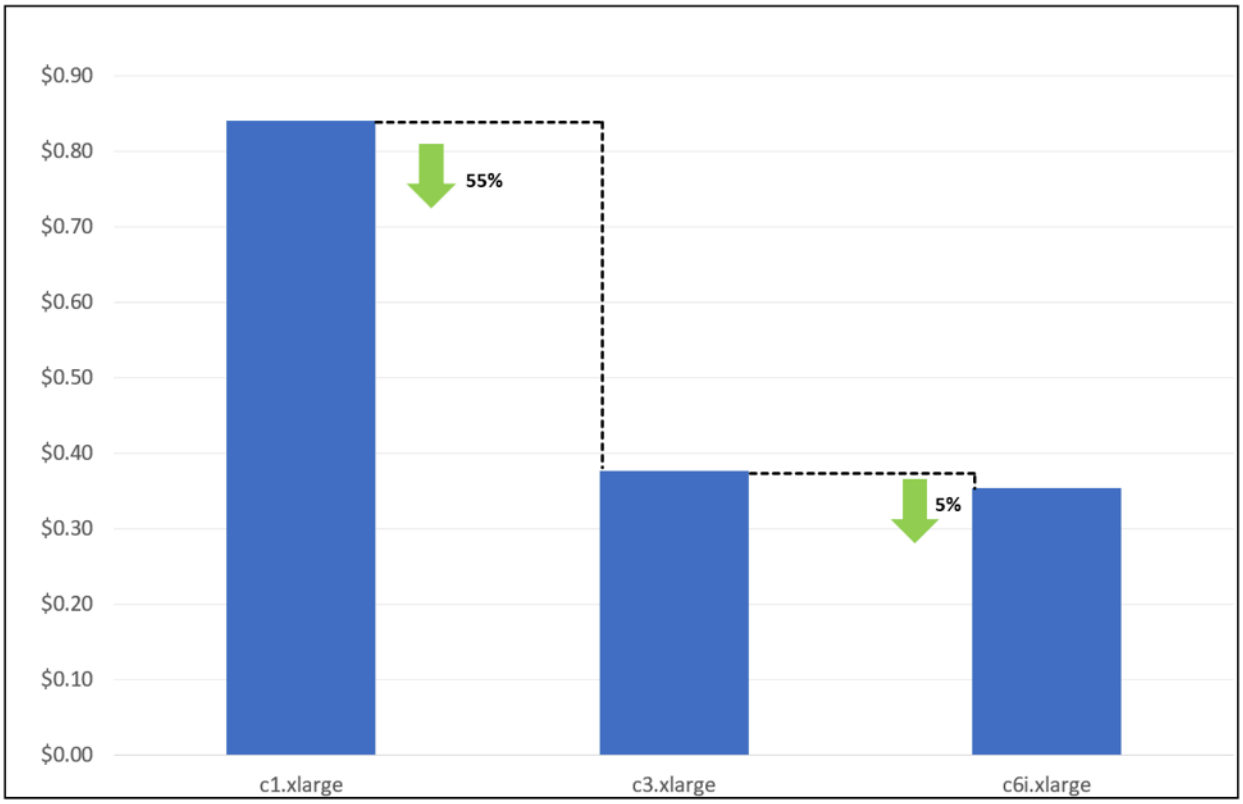
Salah satu karakteristik Amazon EC2 yang paling konsisten adalah bahwa setiap generasi baru menawarkan kinerja harga yang lebih baik daripada pendahulunya. Seperti yang ditunjukkan tabel berikut, harga instans EC2 generasi yang lebih baru menurun dengan setiap rilis berikutnya.

Hitung contoh yang dioptimalkan	Harga per jam	Contoh tujuan umum	Harga per jam	Contoh memori yang dioptimalkan	Harga per jam
C1.xLarge	$0,52	M1.xLarge	$0,35	r1.xlarge	T/A
C3.xLarge	$0,21	M3.xLarge	$0,266	r3.xlarge	$0,333

Hitung contoh yang dioptimalkan	Harga per jam	Contoh tujuan umum	Harga per jam	Contoh memori yang dioptimalkan	Harga per jam
C5.xLarge	$0,17	M5.xLarge	$0,192	r5.xlarge	$0,252

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Bagan berikut membandingkan biaya berbagai generasi instance seri C.



Namun, instance generasi ke-6 memiliki harga yang sama dengan generasi ke-5, seperti yang ditunjukkan tabel berikut.

Hitung contoh yang dioptimalkan	Harga per jam	Contoh tujuan umum	Harga per jam	Contoh memori yang dioptimalkan	Harga per jam
C5.xLarge	$0,17	M5.xLarge	$0,192	r5.xlarge	$0,252
C6i.xLarge	$0,17	M6i.xLarge	$0,192	r6i.xlarge	$0,252

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Meskipun memiliki biaya yang sama, generasi yang lebih baru memberikan kinerja harga yang unggul karena prosesor yang lebih cepat, throughput jaringan yang ditingkatkan, dan peningkatan throughput Amazon Elastic Block Store (Amazon EBS) dan IOPS.

Salah satu peningkatan kinerja harga yang paling signifikan adalah peningkatan instance X2i. Generasi instans ini menawarkan kinerja harga hingga 55 persen lebih besar dari generasi sebelumnya. Seperti yang ditunjukkan tabel berikut, x2iedn menunjukkan peningkatan dalam setiap aspek kinerja (semuanya dengan harga yang sama dengan generasi sebelumnya).

Instans	Harga per jam	v CPUs	RAM	Kecepatan prosesor	Penyimpanan instans	Jaringan	Throughput Amazon EBS	IOPS EBS
x1e.2xlarge	$1.66	8	244	2.3 GHz	237GB SSD	10 Gbps	125 MB/s	7400
x1iedn.2xlarge	$1.66	8	256	3.5 GHz	240GB SSD NVMe	25 Gbps	2500 MB/s	65000

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Contoh alur perencanaan

Pertimbangkan contoh perusahaan analitik yang melacak kendaraan pengiriman dan ingin meningkatkan kinerja SQL Server-nya. Setelah UKM MACO meninjau kemacetan kinerja perusahaan ini, perusahaan beralih dari instance x1e.2xlarge ke instance x2iedn.xlarge. Ukuran instans baru lebih kecil, tetapi penyempurnaan pada instance x2 memungkinkan peningkatan kinerja dan pengoptimalan SQL Server melalui penggunaan Buffer Pool Extensions. Hal ini memungkinkan perusahaan untuk downgrade dari edisi SQL Server Enterprise ke edisi SQL Server Standard. Ini juga memungkinkan perusahaan untuk mengurangi lisensi SQL Server dari 8 v CPUs menjadi 4 v. CPUs

Sebelum optimasi:

Server	Instans EC2	Edisi SQL Server	Biaya bulanan
Prod DB1	x1e.2xlarge	Perusahaan	$3.918,64
Prod DB2	x1e.2xlarge	Perusahaan	$3.918,64
Jumlah			$7.837,28

Setelah optimasi:

Server	Instans EC2	Edisi SQL Server	Biaya bulanan
Prod DB1	x2iedn.xlarge	Standar	$1,215.00
Prod DB2	x2iedn.xlarge	Standar	$1,215.00
Jumlah			$2,430,00

Semua digabungkan, perubahan dari instance x1e.2xlarge ke instance x2iedn.xlarge memungkinkan perusahaan dalam skenario contoh menghemat $5.407 per bulan di server basis data produksinya. Ini mengurangi total biaya beban kerja hingga 69 persen.

> **Note**
>
> Harga didasarkan pada harga per jam berdasarkan permintaan di Wilayah. `us-east-1`

Migrasi ke instance yang lebih baru

Generasi lama Amazon EC2 berjalan pada hypervisor Xen, sementara generasi baru beroperasi pada Sistem Nitro.AWS Sistem Nitro memberikan hampir semua sumber daya komputasi dan memori perangkat keras host ke instans Anda. Ini menghasilkan peningkatan kinerja secara keseluruhan. Ada pertimbangan khusus saat bermigrasi dari instance berbasis Xen ke Nitro. Misalnya, AWS Windows AMIs dikonfigurasi dengan pengaturan default dan penyesuaian yang digunakan oleh media instalasi Microsoft. Kustomisasi mencakup driver dan konfigurasi yang mendukung jenis instans generasi terbaru (instance yang dibangun di atas Sistem Nitro).

Jika Anda meluncurkan instans dari Windows khusus AMIs atau dari Windows yang AMIs disediakan oleh Amazon yang dibuat sebelum Agustus 2018, kami sarankan Anda menyelesaikan langkah-langkah dari Migrasi ke jenis instans generasi terbaru dalam dokumentasi Amazon EC2.

Gunakan instance burstable

Meskipun instans burstable adalah cara yang baik untuk menghemat biaya komputasi, kami sarankan Anda menghindarinya dalam skenario berikut:

- Spesifikasi minimum untuk Windows Server dengan Pengalaman Desktop membutuhkan 2 GB RAM. Hindari menggunakan instans t3.micro atau t3.nano dengan Windows Server karena mereka tidak memiliki jumlah minimum RAM.
- Jika beban kerja Anda runcing tetapi tidak cukup lama menganggur untuk membangun kredit burst, menggunakan instans EC2 normal lebih efisien daripada menggunakan instans burstable. Kami menyarankan untuk memantau kredit CPU Anda untuk memverifikasi ini.
- Kami menyarankan Anda menghindari penggunaan instans burstable dengan SQL Server di sebagian besar skenario. Lisensi untuk SQL Server didasarkan pada jumlah v yang CPUs ditugaskan ke sebuah instance. Jika SQL Server menganggur sebagian besar hari, Anda akan membayar untuk lisensi SQL yang tidak sepenuhnya Anda gunakan. Dalam skenario ini, kami

menyarankan Anda mengkonsolidasikan beberapa instance SQL Server ke server yang lebih besar.

Langkah selanjutnya

Kami menyarankan Anda mengambil langkah-langkah berikut berikut untuk mengoptimalkan biaya untuk instans Windows Amazon EC2:

- Gunakan instans EC2 generasi terbaru untuk kinerja harga terbaik.
- Gunakan instans EC2 dengan prosesor AMD untuk pengurangan biaya komputasi sepuluh persen.
- Maksimalkan pemanfaatan sumber daya dengan memilih jenis instans EC2 yang sesuai dengan beban kerja Anda.

Tabel berikut menunjukkan contoh titik awal khas untuk beban kerja Windows. Opsi tambahan tersedia, seperti volume penyimpanan instans untuk meningkatkan beban kerja SQL Server atau instans EC2 dengan rasio yang jauh lebih besar. vCPU-to-RAM Kami menyarankan Anda menguji beban kerja Anda secara menyeluruh dan menggunakan alat pemantauan seperti AWS Compute Optimizer untuk membantu membuat penyesuaian yang diperlukan.

Beban kerja	Khas	Opsional
Active Directory	T3, M6i	R6i
Server file	T3, M6i	C6i
Server web	T3, C6i	M6i, R6i
SQL Server	R6i	x2iedn, X2IEZN

Jika Anda harus mengubah jenis instans EC2 Anda, prosesnya biasanya hanya melibatkan reboot server sederhana. Untuk informasi selengkapnya, lihat Mengubah jenis instans dalam dokumentasi Amazon EC2.

Sebelum mengubah jenis instans, sebaiknya pertimbangkan hal berikut:

- Anda harus menghentikan instans yang didukung oleh Amazon EBS sebelum dapat mengubah jenis instansnya. Pastikan untuk merencanakan downtime saat instans Anda dihentikan.

Menghentikan instans dan mengubah tipe instansnya mungkin memerlukan waktu beberapa menit, lalu memulai ulang instans Anda mungkin memerlukan waktu yang bervariasi, tergantung skrip pemulaian aplikasi Anda. Untuk informasi selengkapnya, lihat Menghentikan dan memulai instans Anda di dokumentasi Amazon EC2.

- Ketika Anda berhenti dan memulai sebuah instance, AWS pindahkan instance ke perangkat keras baru. Jika instans Anda memiliki IPv4 alamat publik AWS , lepaskan alamat dan berikan contoh Anda IPv4 alamat publik baru. Jika Anda memerlukan IPv4 alamat publik yang tidak berubah, gunakan alamat IP Elastis.
- Anda tidak dapat mengubah jenis instance jika hibernasi diaktifkan pada instance.
- Anda tidak dapat mengubah tipe instans dari Instans Spot.
- Jika instans Anda berada dalam grup Auto Scaling, Amazon EC2 Auto Scaling menandai instans yang dihentikan sebagai tidak sehat, dan dapat menghentikannya serta meluncurkan instance pengganti. Untuk mencegahnya, Anda dapat menangguhkan proses penskalaan untuk grup saat Anda mengubah tipe instans. Untuk informasi selengkapnya, lihat Menangguhkan dan melanjutkan proses untuk grup Auto Scaling di dokumentasi Amazon EC2 Auto Scaling.
- Saat Anda mengubah jenis instance instance dengan volume penyimpanan instans, NVMe instance yang diperbarui mungkin memiliki volume penyimpanan instans tambahan, karena semua volume penyimpanan NVMe instans tersedia meskipun tidak ditentukan dalam Amazon Machine Image (AMI) atau pemetaan perangkat blok instans. Jika tidak, instans yang diperbarui memiliki jumlah volume penyimpanan instans yang sama dengan yang Anda tentukan saat meluncurkan instans asli.

Sumber daya tambahan

- Jenis instans Amazon EC2 (dokumentasi)AWS
- AWS Optimalisasi dan Penilaian Lisensi (AWS dokumentasi)

Bawa lisensi untuk beban kerja Windows dan SQL Server

Ikhtisar

Jika Anda memiliki investasi signifikan dalam beban kerja Microsoft dan perjanjian lisensi perusahaan yang ada, Anda dapat memilih dari beberapa AWS opsi untuk mendukung beban kerja ini, termasuk lisensi yang disertakan (disediakan oleh AWS) dan opsi Bring Your Own License (BYOL). Anda

dapat menggunakan Host Khusus Amazon EC2 untuk sepenuhnya memanfaatkan perjanjian lisensi Microsoft yang ada dan membawa Windows Server ke. AWS Ini dapat menghemat hingga 50 persen pada biaya instans Amazon EC2. Karena lisensi Windows menyumbang sekitar setengah dari biaya instans, membawa Windows Server ke AWS Host Khusus dapat menghasilkan penghematan biaya yang besar. Karena Windows Server tidak dapat dibawa ke penyewaan default (bersama), Host Khusus adalah pilihan ideal jika Anda ingin menggunakan lisensi yang ada untuk Windows Server aktif. AWS

Host Khusus tidak hanya untuk instans Windows Server BYOL. Mereka juga menawarkan fleksibilitas untuk mencocokkan lisensi lokal Anda untuk beban kerja SQL Server yang ada. Host Khusus mengekspos inti fisik server yang mendasarinya, dan memungkinkan Anda untuk melisensikan SQL Server pada tingkat inti fisik. Ini tidak dimungkinkan dalam penyewaan default (bersama) di mana lisensi SQL Server didasarkan pada jumlah virtual yang CPUs dialokasikan ke instance. Fitur ini memungkinkan Anda untuk melisensikan beban kerja SQL Server dengan AWS cara yang konsisten dengan strategi lisensi lokal Anda. Akibatnya, Anda dapat menghemat hingga 50 persen pada biaya lisensi SQL Server dibandingkan dengan penyewaan default (bersama), selain penghematan biaya instans, dengan menggunakan lisensi Windows yang memenuhi syarat. Untuk informasi selengkapnya tentang skenario ini, lihat bagian Memahami lisensi SQL Server dari panduan ini.

Host Khusus Amazon EC

Host Khusus Amazon EC2 pada dasarnya adalah host EC2 yang sama yang AWS digunakan untuk menjalankan penawaran komputasi EC2-nya. Perbedaannya adalah bahwa host ini sepenuhnya berdedikasi untuk satu pelanggan dan menyediakan akses eksklusif ke infrastruktur fisik yang mendasarinya. Anda dapat menggunakan Host Khusus untuk menjalankan instans Anda pada perangkat keras yang sepenuhnya didedikasikan untuk penggunaan Anda, alih-alih berbagi sumber daya dengan AWS pelanggan lain. Ini memberi Anda kontrol yang lebih besar atas sumber daya cloud dan memungkinkan Anda mengurangi biaya dengan membawa lisensi perangkat lunak Anda sendiri, seperti Windows Server dan SQL Server, ke. AWS

Ingatlah hal berikut:

- Dedicated Host adalah server fisik yang sepenuhnya didedikasikan untuk satu pelanggan. Anda mendapatkan visibilitas ke soket dan inti fisik Host Khusus sehingga Anda dapat memenuhi persyaratan kepatuhan lisensi, seperti perjanjian lisensi perangkat lunak per-soket, per-inti, atau per-VM.
- Host Khusus yang dapat mendukung beberapa ukuran instans dari keluarga instans yang sama dikenal sebagai Host Khusus yang heterogen. Keluarga contoh ini termasuk T3, A1, C5, M5, R5,

C5n, R5n, dan M5n. Sebaliknya, keluarga instans lain hanya mendukung satu ukuran instans pada Host Khusus yang sama. Ini disebut host khusus homogen.

- Host Khusus ditagih per host. Ini berarti Anda dikenakan biaya per Host Khusus terlepas dari berapa banyak instans yang berjalan di dalamnya. Harga Host Khusus bervariasi berdasarkan keluarga instans, Wilayah, dan opsi pembayaran yang dipilih. Anda dapat memilih konfigurasi optimal untuk beban kerja Anda untuk mencapai kinerja dan hasil biaya yang Anda inginkan.

Diagram ini menggambarkan perbedaan antara instance penyewaan bersama dan Host Khusus.

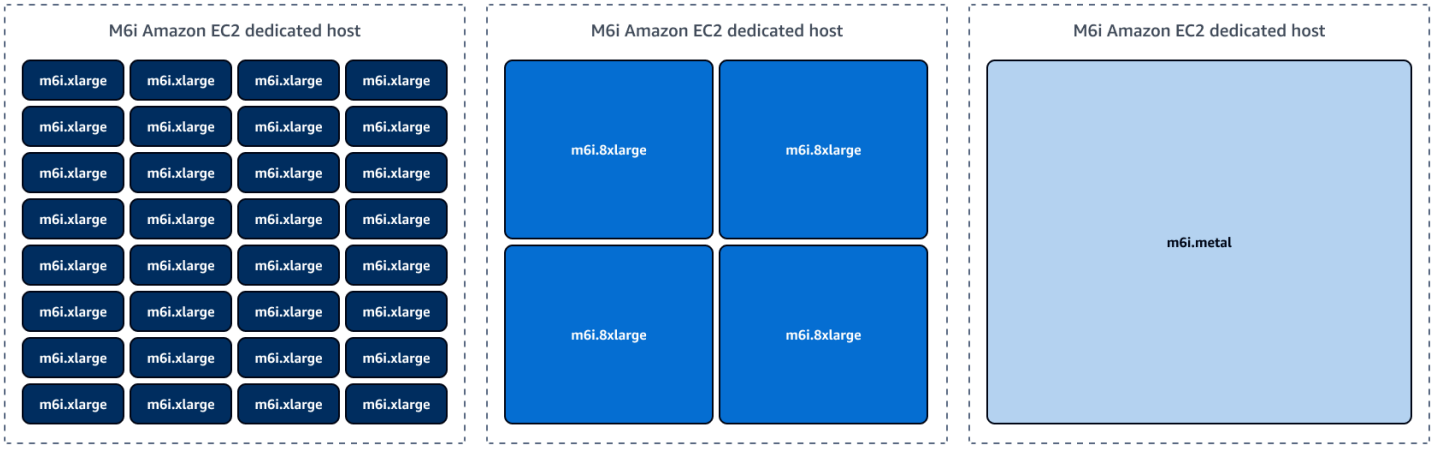
Host Berdedikasi Homogen

Pertimbangkan skenario di mana Host Khusus M6i digunakan. Host Khusus M6i dan R6i memiliki dua soket, 64 core fisik, dan tipe instans pendukung dengan ukuran yang sama. Ini disebut Host Khusus homogen. Ini berarti bahwa jumlah instans yang dapat Anda luncurkan pada satu Host Khusus M6i bergantung pada ukuran instans.

Contoh:

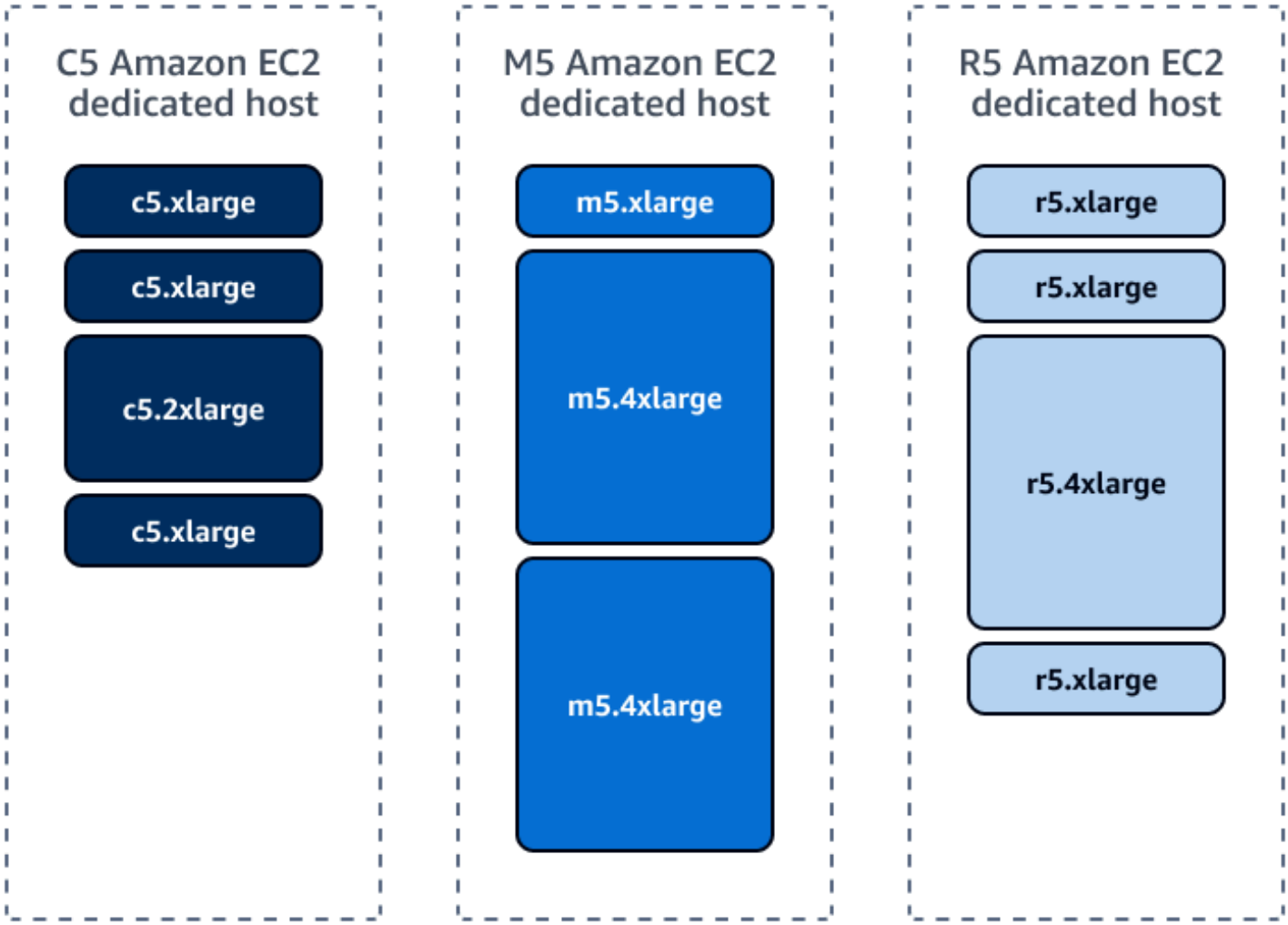
- Dalam kasus xlarge (4 vCPUs), Anda dapat meluncurkan maksimum 32 instance m6i.xlarge pada Host Khusus ini.
- Dalam kasus 8xlarge (32 vCPUs), Anda dapat meluncurkan maksimal 4 instance m6i.8xlarge pada Host Khusus ini.
- Dalam kasus metal (128 vCPUs), Anda dapat meluncurkan instans maksimum 1 m6i.metal pada Host Khusus ini.

Diagram berikut menunjukkan opsi Host Khusus untuk instans M6.



Host Berdedikasi Heterogen

Host Khusus yang mendukung beberapa ukuran instans pada host yang sama disebut sebagai Host Khusus Amazon EC2 yang heterogen. Diagram berikut menunjukkan contoh Host Khusus C5, M5, dan R5 dengan berbagai ukuran instance, seperti 2xlarge, xlarge, dan 4xlarge.



Manajemen Host Khusus

Kami menyarankan Anda mempertimbangkan hal-hal berikut sehubungan dengan pengelolaan Host Khusus Amazon EC2:

- Untuk memanfaatkan sepenuhnya Host Khusus, Anda dapat berbagi satu host di antara beberapa akun dalam organisasi Anda. Berbagi host memungkinkan pengoptimalan sumber daya dan dapat menghasilkan penghematan biaya dengan menggunakan setiap slot yang tersedia di host. Dengan berbagi Host Khusus antar unit bisnis, Anda dapat memusatkan infrastruktur TI Anda dan meningkatkan pemanfaatan sumber daya, sambil tetap mempertahankan pemisahan antar beban kerja. Jika Anda bagian dari organisasi AWS Organizations dan berbagi diaktifkan dalam organisasi Anda, maka konsumen di organisasi Anda secara otomatis diberikan akses ke Host Khusus bersama. Jika tidak, konsumen menerima undangan untuk bergabung dengan berbagi sumber daya dan diberikan akses ke Host Khusus bersama setelah menerima undangan.

- Anda dapat menjalankan Windows Server 2022 pada Host Khusus di bawah model yang disertakan lisensi, karena Windows Server 2019 adalah versi terbaru tempat Anda dapat menggunakan BYOL. Jika Anda ingin menggunakan Windows Server 2022 pada Host Khusus, Anda harus menggunakan instance yang disertakan lisensi Windows Server 2022.
- AWS License Manageradalah solusi komprehensif untuk mengelola lisensi perangkat lunak dari berbagai vendor di seluruh AWS dan lingkungan lokal. Jika Anda menggunakan License Manager, Anda bisa mendapatkan visibilitas dan kontrol yang lebih besar atas bagaimana lisensi perangkat lunak digunakan, yang mengarah pada penghematan biaya dan peningkatan kepatuhan. Anda dapat menggunakan License Manager untuk menetapkan aturan untuk meniru persyaratan lisensi unik Anda. Ini memungkinkan Anda untuk menegakkan aturan tersebut dan mencegah penyalahgunaan lisensi. Ini dapat mengurangi risiko ketidakpatuhan dan meningkatkan proses manajemen lisensi.
- Anda dapat menggunakan License Manager untuk mengotomatiskan penempatan, rilis, dan pemulihan host dengan menggunakan grup sumber daya host. Hal ini dapat meningkatkan produktivitas dan mengurangi overhead manajemen. License Manager juga menyediakan tampilan terpusat penggunaan lisensi di seluruh AWS dan lingkungan lokal berdasarkan aturan lisensi, sehingga memudahkan untuk mengelola pembelian lisensi tambahan, kepatuhan, dan audit vendor di seluruh organisasi Anda. Selanjutnya, License Manager terintegrasi dengan AWS Organizations dan AWS Resource Access Manager (AWS RAM) untuk berbagi konfigurasi lisensi di seluruh akun dan Wilayah. Ini memungkinkan Anda untuk membuat laporan untuk seluruh lingkungan Anda berdasarkan jadwal dan mengelola aturan lisensi secara terpusat dalam satu. Akun AWS Pada akhirnya, ini dapat meningkatkan tata kelola dan mengurangi kompleksitas.
- Saat merancang ketersediaan tinggi untuk Host Khusus dalam satu Wilayah, pastikan Anda telah mengalokasikan minimal dua Host Khusus di minimal dua Availability Zone untuk beban kerja kritis produksi. Untuk informasi selengkapnya, lihat Host Khusus Amazon EC2 untuk Microsoft Windows pada penerapan AWS referensi.
- Untuk setiap keluarga instans Host Khusus, ada batasan jumlah instance yang dapat Anda jalankan untuk setiap ukuran instans. Untuk informasi selengkapnya, lihat Tabel Konfigurasi Host Khusus di dokumentasi Amazon EC2.

AWS opsi lisensi

Lisensi diklasifikasikan ke dalam kategori utama berikut:

- Termasuk lisensi - Opsi lisensi ini memungkinkan Anda untuk membeli dan menggunakan lisensi sesuai permintaan, hanya membayar untuk apa yang Anda gunakan. Ini ideal untuk

kasus penggunaan di mana Anda mencari fleksibilitas dalam penggunaan lisensi Anda dan ingin menghindari biaya di muka. Anda dapat memilih dari berbagai Windows Server, SQL Server, dan produk Microsoft lainnya.

- Produk BYOL dengan Mobilitas Lisensi - Jika Anda sudah memiliki lisensi yang ada dan ingin menggunakannya di cloud, opsi lisensi ini memungkinkan Anda untuk membawa lisensi Anda sendiri ke cloud melalui program Microsoft License Mobility. Produk dengan mobilitas lisensi, seperti SQL Server dengan Jaminan Perangkat Lunak (SA), dapat dibawa ke penyewaan bersama atau khusus. Ini mengurangi biaya AWS instance.
- Produk BYOL tanpa Mobilitas Lisensi — Untuk produk Microsoft seperti Windows Server yang tidak memiliki Mobilitas Lisensi, AWS menyediakan opsi khusus untuk menggunakan produk ini di cloud. Selain itu, Dedicated Host memungkinkan lisensi pada tingkat inti fisik, sehingga memungkinkan untuk menghemat 50 persen atau lebih pada lisensi yang diperlukan untuk menjalankan beban kerja Anda. Host Khusus adalah pilihan yang sangat baik untuk beban kerja yang stabil dan dapat diprediksi yang berjalan sebagian besar waktu.

Membawa lisensi Windows Server

Membawa lisensi Windows Anda sendiri adalah salah satu strategi paling efektif untuk optimasi lisensi karena memungkinkan Anda untuk mengambil keuntungan dari investasi yang ada dan mengurangi AWS pengeluaran Anda. Skenario BYOL tertentu tidak memerlukan manfaat SA atau Mobilitas Lisensi, tetapi infrastruktur khusus Amazon EC2 selalu diperlukan. Agar memenuhi syarat, Anda harus telah membeli lisensi abadi sebelum 1 Oktober 2019 atau menambahkannya sebagai true-up berdasarkan Pendaftaran Perusahaan aktif yang berlaku sebelum 1 Oktober 2019. Dalam skenario BYOL khusus ini, Anda hanya dapat memutakhirkan lisensi ke versi yang tersedia sebelum 1 Oktober 2019. Misalnya, jika Anda menjatuhkan SA pada tahun 2017, Anda memiliki hak untuk menerapkan hanya hingga Windows Server 2016, bukan 2019. Namun, 2019 adalah versi terakhir yang memenuhi syarat untuk BYOL. AWS Untuk informasi selengkapnya, lihat Lisensi — Windows Server dalam AWS dokumentasi.

Membawa lisensi dapat secara signifikan berdampak pada biaya menjalankan beban kerja Microsoft. AWS Ketika Anda membawa lisensi Anda sendiri, Anda tidak diharuskan membayar biaya lisensi tambahan untuk instans yang berjalan di cloud, yang dapat menyebabkan penghematan biaya yang cukup besar.

Tabel berikut menunjukkan biaya bulanan sesuai permintaan untuk menjalankan instance c5.xlarge tunggal 24/7 pada berbagai konfigurasi.

Konfigurasi	Biaya bulanan (USD)
Windows Server+SQL Server edisi Perusahaan	$1,353.00 (LI)
Windows Server+SQL Server edisi Standar	$609.00 (LI)
Hanya Windows Server	$259.00 (LI)
Komputasi saja (Linux)	$127.00

Anda dapat menggunakan lisensi yang ada untuk mengurangi biaya lisensi dan menghemat uang untuk keseluruhan AWS tagihan Anda.

Agar memenuhi syarat untuk BYOL di Host Khusus Amazon EC2, Anda harus membawa lisensi perangkat lunak Anda sendiri, seperti untuk Windows Server dan SQL Server. BYOL memungkinkan Anda untuk menggunakan lisensi yang ada AWS dan dapat menghasilkan penghematan biaya. Untuk membawa lisensi Anda sendiri, Anda harus memiliki hak lisensi dari vendor perangkat lunak dan juga harus menyediakan media instalasi atau gambar untuk perangkat lunak. Media instalasi atau gambar dapat digunakan untuk meluncurkan instance pada Host Khusus. Untuk mempelajari selengkapnya tentang membuat BYOL AMI, lihat Cara membuat Windows Server Bring-Your-Own-License AMIs dari lokal dengan VM Impor/Ekspor di Microsoft Workloads di blog. AWS

> **Note**
>
> Jenis lisensi yang disetel ke Auto setara dengan opsi yang AWS disertakan lisensi. Opsi ini dapat menghasilkan pengeluaran sesuai permintaan yang tidak diinginkan. Anda perlu mengganti jenis lisensi.

Skenario pengoptimalan biaya

Ukuran yang tepat dan mengoptimalkan lisensi adalah komponen kunci dari pengoptimalan biaya. AWS Jika Anda menerapkan strategi yang tepat, Anda dapat mengurangi biaya lisensi, mempertahankan kepatuhan, dan mencapai nilai terbaik dari investasi lisensi Anda dengan menggunakan Host Khusus Amazon EC2 dan opsi BYOL.

Bagian ini mencakup contoh skenario berikut:

- Penghematan biaya dengan Host Khusus T3
- Membandingkan sewa bersama dengan Host Khusus dengan SQL Server BYOL
- Penyebaran SQL Server yang sangat tersedia

Penghematan biaya dengan Host Khusus T3

Host Khusus T3 berbeda dari Host Khusus Amazon EC2 lainnya yang secara tradisional menyediakan sumber daya CPU tetap. Host Khusus T3, sebaliknya, mendukung instans burstable yang mampu berbagi sumber daya CPU, memberikan kinerja CPU dasar, dan meledak saat diperlukan. Berbagi sumber daya CPU, juga dikenal sebagai oversubscription, adalah hal yang memungkinkan satu Host Khusus T3 untuk mendukung instans hingga empat kali lebih banyak daripada Host Khusus tujuan umum yang sebanding.

Host Khusus T3 mendorong TCO yang lebih rendah dengan menghadirkan kepadatan instans yang lebih tinggi daripada Host Khusus Amazon EC2 lainnya. Instans T3 Burstable memungkinkan Anda untuk mengkonsolidasikan jumlah instance yang lebih tinggi dengan penggunaan CPU low-to-moderate rata-rata pada host yang lebih sedikit daripada sebelumnya. Host Khusus T3 juga menawarkan ukuran instans yang lebih kecil dalam jumlah vCPU dan kombinasi memori yang lebih besar daripada Host Khusus Amazon EC2 lainnya. Ukuran instans yang lebih kecil dapat berkontribusi pada TCO yang lebih rendah dan membantu memberikan rasio konsolidasi yang setara dengan atau lebih besar dari host lokal.

Host Khusus T3 paling cocok untuk menjalankan perangkat lunak BYOL dengan pemanfaatan low-to-moderate CPU dan lisensi perangkat lunak per-soket, per-inti, atau per-VM yang memenuhi syarat, termasuk desktop Microsoft Windows, Windows Server, SQL Server, dan Oracle Database.

Gunakan Host Khusus T3 untuk mengurangi lisensi Pusat Data Windows Server (per inti)

Di lingkungan lokal, Anda memanfaatkan fakta bahwa Anda dapat dengan mudah membatalkan langganan fisik Anda CPUs di VMware host dan mencapai tingkat konsolidasi yang tinggi.

Pertimbangkan contoh berikut. Saat ini Anda menggunakan VMware host 10x36 core, 384 GB RAM di lingkungan lokal. Selain itu, setiap host menjalankan 96x2 vCPU, 4 GB RAM Windows Server mesin virtual dengan pemanfaatan CPU rata-rata rendah.

Anda sekarang dapat mencapai tingkat konsolidasi yang jauh lebih tinggi dengan memindahkan mesin virtual Anda ke Host Khusus T3, yang memiliki jumlah RAM dua kali lipat dibandingkan dengan host lokal VMware Anda saat ini. Anda dapat menjalankan jumlah server yang sama di Host Khusus

T3 dengan biaya host 50 persen lebih sedikit. Ini dapat membantu Anda mengurangi biaya lisensi Windows Server hingga 33 persen. Tabel berikut menyoroti penghematan penggunaan Host Khusus T3.

	Host lokal VMware	Tuan Rumah Khusus T3	Tabungan
Server fisik	10	5	
Inti fisik per host	36	48	
RAM per host (GB)	384	768	
2 vCPU, 4 GB RAM per host VMs	96	192	
Jumlah total VMs	960	960	
Total lisensi Pusat Data Windows Server (per inti) = (Jumlah server * Jumlah inti fisik)	10 * 36 = 360	5 * 48 = 240	33%

Membandingkan sewa bersama dengan Host Khusus dengan SQL Server BYOL

Pertimbangkan contoh praktis untuk menunjukkan nilai Host Khusus Amazon EC2. Dalam skenario ini, organisasi menjalankan beban kerja SQL Server di lingkungan lokal dengan 240 core dan ingin menerapkan beban kerja yang sama dengan biaya efektif. AWS Jika organisasi ini membawa lisensi mereka sendiri (BYOL), mereka terus membayar SA dan mengurangi jumlah core secara langsung mempengaruhi biaya mereka.

Diagram berikut membandingkan AWS penghematan antara hak Microsoft dan SQL Server.

Microsoft entitlements (Enterprise Agreements)	
	Number of cores
SQL Server Enterprise edition	208
SQL Server Standard edition	32
Total SA cost	**$341,000**

SQL Server savings with AWS	
AWS shared vCPUs	AWS BYOL/Dedicated Hosts cores
120	96
20	-
$197,418	**$151,355**

Dengan mengukur instans yang tepat pada penyewaan AWS bersama, Anda dapat mengurangi lisensi SQL Server menjadi 140 core. Ini menghasilkan biaya SA $197.000.

Host Khusus Amazon EC2 memungkinkan Anda melisensikan SQL Server pada tingkat inti fisik. Ini tidak mungkin dalam penyewaan bersama di mana lisensi SQL Server didasarkan pada jumlah v yang CPUs dialokasikan ke instance. Akibatnya, dengan menggunakan dua Host Khusus R5 dengan masing-masing 48 core, Anda hanya perlu menutupi 96 core, bukan 140 v yang CPUs diperlukan pada penyewaan bersama. Dengan menerapkan Host Khusus R5 dan melisensikan beban kerja pada tingkat fisik, Anda dapat menurunkan jumlah lisensi edisi SQL Server Enterprise yang diperlukan menjadi 96 core. Ini berarti Anda dapat menyebarkan sebanyak 192 core (akuntansi untuk hyper-threading) beban kerja SQL Server, sambil tetap memenuhi persyaratan lisensi dan mencapai penghematan biaya yang signifikan.

Dalam hal ini, organisasi membayar sekitar $341.000 per tahun dalam biaya SA. Setelah menentukan ukuran yang tepat pada sewa bersama, mereka mengurangi biaya menjadi $197.000 dengan 140 vCPU. Host Khusus Amazon EC2 selanjutnya mengurangi biaya menjadi $151.000 (penurunan sekitar 56 persen).

Penyebaran SQL Server yang sangat tersedia

Contoh ini menganalisis bagaimana biaya dapat mempengaruhi penyebaran SQL Server AWS dengan berbagai pertimbangan lisensi. Misalkan sebuah organisasi perlu menyebarkan enam server SQL Server Enterprise AWS untuk mendukung tiga aplikasi. Server ini membutuhkan ketersediaan tinggi dan masing-masing memiliki 16 v CPUs dan 256 GB RAM. Lihat detail skenario berikut:

- Server - SQL Server
- Edisi sistem operasi - Pusat Data Windows Server 2019
- Edisi SQL Server - SQL Server Enterprise 2019
- vCPU — 16
- Memori (GB) - 256
- Kuantitas - 6

Untuk mengoptimalkan biaya AWS tanpa mengorbankan kinerja, kami menyarankan Anda untuk mengukur instans yang tepat berdasarkan pemanfaatan CPU, memori, jaringan, dan disk (IOPS/BW). Setelah mengukur beban kerja dengan benar, letakkan pada tipe instance x2iedn.4xlarge, yang menawarkan 16 v. CPUs Namun, jenis instance ini juga menyertakan dua kali memori yang diperlukan untuk beban kerja. Optimalisasi lebih lanjut masih dimungkinkan.

Skenario 1

Sebuah organisasi menyebarkan enam server SQL Server Enterprise pada penyewaan AWS bersama dengan menggunakan opsi yang disertakan lisensi untuk Windows dan SQL Server. Dengan opsi ini, biaya lisensi Windows dan SQL Server dimasukkan ke dalam harga instans. Lihat detail skenario berikut:

- Sewa bersama (contoh) - x2iedn.4xlarge
- Biaya per jam (USD) - $10.0705
- Biaya bulanan per unit (USD) - $7,351.47
- Jumlah server — 6
- CPU — 16
- Memori — 512
- Biaya bulanan untuk 6 server - $44,108

Skenario 2

Sebuah organisasi memiliki SA dan BYOL untuk SQL Server pada penyewaan bersama. Ini berarti bahwa organisasi menggunakan opsi yang disertakan lisensi untuk Windows, tetapi menyediakan lisensi SQL Server mereka sendiri berdasarkan jumlah v yang CPUs dialokasikan ke instance. Karena organisasi memiliki enam server SQL Server Enterprise dengan CPUs masing-masing 16 v, total 96 v CPUs diperlukan. Lihat detail skenario berikut:

- Sewa bersama (contoh) - x2iedn.4xlarge
- Biaya per jam (USD) — $4.0705
- Biaya bulanan per unit (USD) - $2971.47
- Jumlah server — 6
- CPU — 16
- Memori — 512
- Inti BYOL - 96
- Biaya bulanan untuk 6 server - $17,828

Dengan membawa lisensi SQL Server mereka sendiri dengan SA, organisasi dalam skenario ini dapat mencapai penghematan biaya dibandingkan dengan menggunakan opsi yang disertakan

lisensi untuk SQL Server. Penghematan biaya yang tepat tergantung pada harga dan ketentuan perjanjian lisensi tertentu. Dalam skenario ini, AWS biaya berkurang sebesar $26.280 per bulan saat membawa lisensi SQL Server Enterprise. AWS

Skenario 3

Sebuah organisasi memiliki BYOL untuk Windows dan SQL Server di Host Khusus Amazon EC2. Ini berarti bahwa organisasi akan menetapkan lisensi pada tingkat inti fisik, memungkinkan mereka untuk melisensikan hanya inti fisik host. Lisensi pada tingkat inti fisik memungkinkan Anda untuk menyebarkan jumlah maksimum instance tanpa mempengaruhi lisensi yang diperlukan. Model lisensi ini biasanya digunakan dengan Windows Server Datacenter dan edisi SQL Server Enterprise.

Skenario ini menggunakan dua Host Khusus Amazon EC2 X2IEZN. Setiap host memiliki 24 inti fisik dan 48 vCPUs. Ini memberikan kapasitas yang memadai untuk enam server SQL Server Enterprise dengan 16 v CPUs dan 256 GB RAM masing-masing. Lihat detail skenario berikut:

- Jumlah host khusus — 2
- Keluarga contoh — x2iezn
- Biaya per jam (USD) - $11.009
- Biaya bulanan per unit (USD) - $8,036
- Inti fisik — 48
- Tersedia vCPU - 96
- Lisensi inti Windows Server diperlukan - 24
- Lisensi yang diperlukan untuk inti SQL Server Enterprise - 24
- Biaya bulanan - 16,073

Total biaya untuk dua Host Khusus Amazon EC2 keluarga X2IEZN adalah $16.073 per bulan. Untuk informasi selengkapnya tentang harga, lihat AWS Kalkulator Harga perkiraan untuk skenario ini. Organisasi dalam skenario ini dapat menghemat $1.755.65 per bulan dengan membawa lisensi Windows mereka. Jika mereka menggunakan Host Khusus Amazon EC2, mereka juga dapat mengurangi jumlah lisensi SQL Server yang diperlukan. Dalam penyewaan bersama, mereka akan membutuhkan 96 lisensi SQL Server Enterprise untuk mencakup enam server SQL Server Enterprise dengan masing-masing 16 vCPU. Namun, dengan menggunakan Host Khusus Amazon EC2 dan lisensi pada tingkat inti fisik, mereka dapat mengurangi jumlah lisensi yang diperlukan menjadi 48 core.

Detail berikut membandingkan biaya dari contoh 3 dan menunjukkan berapa banyak yang dapat Anda hemat dengan menerapkan beban kerja di Host Khusus Amazon EC2 dengan opsi BYOL dibandingkan dengan skenario lainnya.

- Server lokal - SQL Server
- vCPU — 16
- Memori - 256
- Jumlah server — 6
- Biaya bulanan untuk skenario 1: Windows (LI) +SQL Server Enterprise (LI) - $44,108
- Biaya bulanan untuk skenario 2: Windows (LI) +SQL Server Enterprise (BYOL) - $17,828
- Biaya bulanan untuk skenario 3: Windows (LI) +SQL Server Enterprise (BYOL) di Host Khusus Amazon EC2 - $16.073

> **Note**
>
> Biaya didasarkan pada harga berdasarkan permintaan. Anda dapat mengurangi biaya lebih lanjut dengan menggunakan Savings Plans atau Dedicated Reserved Instances. Opsi ini menawarkan model penetapan harga yang fleksibel dengan penghematan biaya yang signifikan dibandingkan dengan harga sesuai permintaan. Dengan rencana ini, Anda dapat berkomitmen untuk jangka waktu satu atau tiga tahun. Untuk informasi selengkapnya, lihat bagian Optimalkan pengeluaran untuk Windows di Amazon EC2 dari panduan ini.

Pertimbangkan opsi pembayaran berikut untuk Host Khusus Amazon EC2:

- Host Khusus (dokumentasi Amazon EC2)
- Reservasi Tuan Rumah Khusus (dokumentasi Amazon EC2)
- Savings Plans (dokumentasi Amazon EC2)

AWS Kalkulator HargaSekarang mendukung harga Dedicated Host. Ini dapat membantu Anda memilih Host Khusus yang mendasarinya.

Rekomendasi optimisasi biaya

Kami menyarankan Anda mengambil langkah-langkah berikut untuk mengoptimalkan biaya Anda dengan menggunakan AWS Cost Explorer:

1. Aktifkan Cost Explorer.
2. Gunakan Cost Explorer untuk melihat dan menganalisis biaya dan penggunaan penerapan Host Khusus Amazon EC2 Anda.
3. Validasi bahwa Anda menjalankan BYOL. Anda dapat menampilkan detail platform berikut dan nilai operasi penggunaan pada instans atau halaman AMI di konsol Amazon EC2, atau dalam respons yang dikembalikan oleh `describe-images` perintah atau. `describe-instances`
   - Detail platform: Windows, Operasi penggunaan:: 0002 RunInstances (Termasuk lisensi)
   - Detail platform: Windows BYOL, Operasi penggunaan: :0800 RunInstances

Sumber daya tambahan

- Jenis lisensi yang memenuhi syarat untuk konversi jenis lisensi (AWS License Manager dokumentasi)
- AWS License Manager Lokakarya & Tuan Rumah Khusus (AWS License Manager Workshop)
- Host Khusus Amazon EC2 (dokumentasi FAQs AWS)
- Cara membuat Windows Server Bring-Your-Own-License AMIs dari lokal dengan VM Impor/Ekspor (Microsoft Workloads di blog) AWS
- VM Impor/Ekspor (dokumentasi)AWS
- Amazon Web Services dan Microsoft: Pertanyaan yang Sering Diajukan (AWS dokumentasi)
- Konversi jenis lisensi di License Manager (AWS License Manager dokumentasi)
- Menyebarkan SQL Server yang sangat tersedia di Host Khusus Amazon EC2 (Blog Operasi & Migrasi Cloud)AWS

Optimalkan pengeluaran untuk Windows di Amazon EC2

Ikhtisar

Salah satu kekhawatiran utama tentang migrasi server ke AWS adalah biaya infrastruktur. Memang benar bahwa salah satu manfaat cloud adalah membayar sumber daya sesuai permintaan, tetapi ada beban kerja produksi yang perlu tersedia 24/7/365. Savings Plans dirancang untuk menghemat uang pada AWS penggunaan kondisi tunak Anda di seluruh instans EC2,, dan. AWS Lambda AWS Fargate

Savings Plans menawarkan model harga yang fleksibel dan dapat membantu Anda mengurangi harga di Amazon EC2, Fargate, Lambda, dan SageMaker Amazon AI dengan imbalan komitmen terhadap jumlah penggunaan yang konsisten (misalnya, $10/jam). Anda berkomitmen untuk jumlah yang konsisten dari pengeluaran komputasi per jam selama satu atau tiga tahun dan, sebagai gantinya, Anda menerima diskon untuk penggunaan itu.

Anda dapat memilih dari tiga opsi pembayaran yang berbeda dengan Savings Plans:

- Opsi No Upfront tidak memerlukan pembayaran di muka, dan komitmen Anda dibebankan murni setiap bulan.
- Opsi Partial Upfront menawarkan harga yang lebih rendah untuk Savings Plans. Anda dikenakan biaya setidaknya setengah dari komitmen Anda di muka dan sisanya dibebankan setiap bulan.
- Opsi All Upfront menawarkan harga terendah dan seluruh komitmen Anda dibebankan dalam satu pembayaran.

Anda dapat melacak masa kedaluwarsa Savings Plans dan Savings Plans antrian yang akan datang. AWS Cost Explorer Anda dapat menggunakan peringatan Savings Plans untuk menerima peringatan email awal 1, 7, 30, atau 60 hari sebelum tanggal kedaluwarsa paket Anda, atau ketika komitmen antri untuk pembelian. Pemberitahuan ini juga mengingatkan Anda pada tanggal kedaluwarsa. Anda dapat mengirim notifikasi hingga 10 penerima email.

Memahami Savings Plans

Setiap jenis penggunaan komputasi memiliki tingkat permintaan dan tarif Savings Plans. Jika Anda berkomitmen untuk $10/jam penggunaan komputasi, maka Anda mendapatkan harga Savings Plans untuk semua penggunaan hingga $10 dengan tarif Savings Plans. Setiap penggunaan di luar komitmen pengeluaran komputasi dibebankan pada tingkat permintaan reguler. Anda dapat memulai dengan Savings Plans dengan menggunakan Cost Explorer di file Konsol Manajemen AWS.

Anda dapat dengan mudah membuat komitmen terhadap Savings Plans dengan menggunakan rekomendasi yang disediakan di Cost Explorer untuk mewujudkan penghematan terbesar. Komitmen per jam yang direkomendasikan didasarkan pada penggunaan berdasarkan permintaan historis Anda dan pilihan jenis paket, jangka waktu, dan opsi pembayaran Anda. Savings Plans pertama kali diterapkan ke akun yang membeli paket, dan kemudian dibagikan ke akun lain dalam keluarga penagihan konsolidasi.

> **Note**
>
> Opsi berbagi Savings Plans di AWS Organizations diaktifkan secara default. Anda dapat menolak opsi ini di AWS Billing konsol akun pembayar. Anda dapat mengunjungi halaman Rekomendasi untuk melihat Savings Plans yang AWS direkomendasikan untuk membantu Anda menghemat penggunaan yang memenuhi syarat. Rekomendasi ini dapat disegarkan kapan saja untuk memudahkan Anda membeli Savings Plans yang optimal.

Compute Savings Plans

Compute Savings Plans memberikan fleksibilitas paling besar dan membantu mengurangi biaya Anda. Paket ini secara otomatis berlaku untuk penggunaan instans EC2 terlepas dari keluarga instans, ukuran, Availability Zone, Region, sistem operasi, atau tenancy. Mereka juga berlaku untuk penggunaan Fargate atau Lambda. Misalnya, dengan Compute Savings Plans, Anda dapat mengubah instans C4 ke M5, mengalihkan beban kerja dari UE (Irlandia) ke UE (London), atau memindahkan beban kerja dari EC2 ke Fargate atau Lambda kapan saja. Anda secara otomatis terus membayar harga Savings Plans.

EC2 Instance Savings Plans

EC2 Instance Savings Plans memberikan diskon terdalam sebagai imbalan atas komitmen untuk penggunaan keluarga instans individu di suatu Wilayah (misalnya, berkomitmen pada tingkat penggunaan M5 yang konsisten di Virginia N.). Ini secara otomatis memberi Anda diskon pada harga sesuai permintaan dari keluarga instans yang dipilih di Wilayah tersebut terlepas dari Availability Zone, ukuran, sistem operasi, atau penyewaan. EC2 Instance Savings Plans memungkinkan Anda mengubah penggunaan antar instans dalam keluarga di Wilayah tersebut. Misalnya, Anda dapat berpindah dari c5.xlarge yang menjalankan Windows ke c5.2xlarge yang menjalankan Linux, dan secara otomatis mendapat manfaat dari harga Savings Plans.

Baik Compute dan Instance Savings Plans EC2 berlaku untuk instans EC2 yang merupakan bagian dari Amazon EMR, Amazon Elastic Kubernetes Service (Amazon EKS), dan Amazon Elastic Container Service (Amazon ECS) cluster. Biaya Amazon EMR, Amazon EKS, dan Amazon ECS tidak ditanggung oleh Savings Plans, tetapi instans EC2 yang mendasarinya adalah. EC2 Instance Savings Plans diterapkan sebelum Compute Savings Plans karena Compute Savings Plans memiliki penerapan yang lebih luas.

> **Note**
>
> Anda tidak dapat mengubah Savings Plans dengan mudah setelah Anda membuat komitmen. Kami menyarankan Anda merencanakan dengan cermat sebelum membuat komitmen pada salah satu opsi Savings Plans. Savings Plans menawarkan harga yang lebih rendah dibandingkan dengan harga sesuai permintaan dengan imbalan komitmen, dan tidak dapat dibatalkan selama jangka waktu tersebut.

Contoh komitmen per jam

Jika Anda membeli Savings Plans, Anda membuat komitmen moneter per jam untuk jangka waktu rencana. Jika Anda berkomitmen untuk $10/jam penggunaan komputasi, harga Savings Plans secara otomatis diterapkan pada semua penggunaan hingga $10 dolar setiap jam. Setiap penggunaan di luar komitmen dibebankan pada tingkat permintaan reguler. Anda dapat menggunakan alat rekomendasi pembelian Savings Plans di Cost Explorer untuk mendapatkan komitmen yang direkomendasikan yang dapat memaksimalkan penghematan Anda. Komitmen keuangan per jam untuk rencana tertentu tidak dapat dimodifikasi untuk jangka waktu rencana. Jika Anda menginginkan peningkatan komitmen setelah menganalisis penggunaan, maka Anda dapat membeli Savings Plans tambahan untuk menutupi penggunaan berlebih.

Manfaat dari Savings Plans

Dibandingkan dengan Instans Cadangan, Savings Plans menawarkan model harga yang lebih fleksibel yang dapat menghemat uang Anda sambil memanfaatkan pilihan opsi komputasi yang lebih luas yang ditawarkan oleh Savings Plans. Savings Plans menawarkan diskon, bahkan saat kebutuhan komputasi Anda berubah. Ini dapat membantu Anda mengikuti lingkungan dinamis Anda yang terus berubah tanpa menimbulkan overhead manajemen tambahan. Berikut adalah beberapa manfaat lain menggunakan Savings Plans:

- Mudah digunakan — Dapatkan diskon otomatis dengan imbalan komitmen moneter.
- Fleksibilitas — Komitmen tunggal yang berlaku di berbagai jenis penggunaan.
- Potensi tabungan — Ada berbagai cara untuk menabung. Pertimbangkan contoh berikut:
  - Penghematan 60 persen pada beban kerja Windows Server menggunakan Compute Savings Plans (d2.8xlarge, 3 tahun, all upfront, windows, shared tenancy, us-east-2)
  - Penghematan 73 persen pada beban kerja Windows Server menggunakan EC2 Instance Savings Plans (d2.8xlarge, 3 tahun, all upfront, windows, shared tenancy, us-east-2)

- Penghematan 28—41 persen untuk jenis instans non-eksotis (keluarga t3, 3 tahun, semua di muka, jendela, sewa bersama, us-east-2)
- Penghematan rata-rata 25-40 persen untuk Windows Server

> **Note**
>
> EC2 Instance Savings Plans menawarkan diskon yang lebih besar daripada Compute Savings Plans karena fleksibilitas yang berkurang. Anda berkomitmen untuk menggunakan dengan harga diskon.

Setiap jenis penggunaan komputasi memiliki tingkat Savings Plans dan on-demand rate. Tabel berikut menunjukkan Savings Plans dan tarif sesuai permintaan untuk setiap jenis sistem operasi. Anda dikenakan tarif Savings Plans pada penggunaan yang dilakukan dan penggunaan apa pun di luar komitmen dibebankan dengan tarif sesuai permintaan reguler.

Nama instans	Tarif Savings Plans	Penghematan sesuai permintaan	Tingkat permintaan	Sistem operasi	Region	Opsi pembayaran	Panjang jangka waktu
x2iedn.xlarge	$0,32	61%	$0,83	Linux	AS Timur (Virginia Utara)	Tidak ada di muka	3
x2iedn.xlarge	$2,01	50%	$1,02	Windows	AS Timur (Virginia Utara)	Tidak ada di muka	3
x2iedn.xlarge	$1,02	20%	$2,52	Lisensi Windows disertakan+edisi SQL Server	AS Timur (Virginia Utara)	Tidak ada di muka	3

Nama instans	Tarif Savings Plans	Penghemat an sesuai permintaa n	Tingkat permintaa n	Sistem operasi	Region	Opsi pembayara n	Panjang jangka waktu
				Enterpris e			
x2iedn.xl arge	$0,32	61%	$0,83	BYOL	AS Timur (Virginia Utara)	Tidak ada di muka	3

Savings Plans termasuk sistem operasi, dan mereka memiliki diskon terpisah untuk BYOL. Semuanya dipecah dalam kalkulator Compute Savings Plans.

Model harga Instans Cadangan

AWS memiliki model harga lain berdasarkan komitmen yang dikenal sebagai Instans Cadangan. Model ini dapat menjadi masalah jika komputasi Anda berubah setelah Anda membuat komitmen, menyebabkan Instans Cadangan tidak digunakan. Savings Plans dirancang untuk menawarkan pengurangan biaya yang serupa dengan Instans Cadangan Standar dan Konvertibel, tetapi dengan fleksibilitas yang jauh lebih besar. Compute Savings Plans memberikan harga yang lebih rendah untuk penggunaan instans EC2 terlepas dari keluarga instans, ukuran, sistem operasi, sewa, atau Wilayah. Mereka juga memungkinkan fleksibilitas maksimum.

Tabel berikut dapat membantu Anda memilih antara Savings Plans atau Instans Cadangan.

	Instans Terpesan	EC2 Instance Savings Plans	Compute Savings Plans
Rata-rata diskon 1 tahun	Hingga 38%	Hingga 29%	Hingga 29%
Rata-rata diskon 3 tahun	Hingga 58%	Hingga 73%	Hingga 60%
Keluarga instans	Tetap	Tetap	Fleksibel

	Instans Terpesan	EC2 Instance Savings Plans	Compute Savings Plans
Ukuran instans	Tetap (bukan Linux)	Fleksibel	Fleksibel
Geografi	1 Wilayah	1 Wilayah	Fleksibel
Sistem operasi	Tetap	Fleksibel	Fleksibel
Layanan	Amazon EC2 atau Amazon RDS	Amazon EC2	Amazon EC2, Fargate, Lambda
Opsi pembayaran	Semua, sebagian, tidak ada di muka	Semua, sebagian, tidak ada di muka	Semua, sebagian, tidak ada di muka
Batas instans	20 per Zona Ketersediaan	Tidak ada batas	Tidak ada batas

> **Note**
>
> Savings Plans bekerja dengan memberi Anda diskon berdasarkan komitmen moneter per jam. Komitmen keuangan per jam tidak dapat dibatalkan atau diubah selama jangka waktu paket Anda, tetapi Anda dapat membeli Savings Plans tambahan untuk menutupi penggunaan tambahan. Ini memungkinkan Anda untuk menjaga komitmen per jam yang konsisten seiring pertumbuhan armada Anda.

Anda dapat menggunakan alat seperti AWS Cost Exploreratau Dasbor AWS Cloud Intelijen untuk melacak komitmen Anda. Cost Explorer menyediakan garis target cakupan yang dapat membantu organisasi Anda merencanakan strategi cakupan Savings Plans. Jika 75 persen dari beban kerja Anda adalah kondisi mapan, maka 75 persen adalah target yang baik. Ini menyisakan 25 persen pengeluaran sesuai permintaan/variabel berdasarkan beban kerja yang dinamis. Jika Anda perlu meningkatkannya menjadi 85 persen, Anda dapat membeli komitmen Savings Plans lain untuk meningkatkan komitmen moneter per jam.

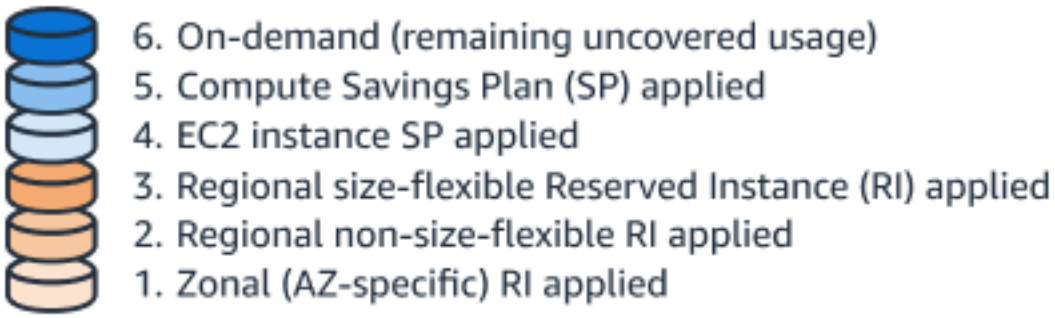
> **Note**
>
> Kami menyarankan Anda membeli Savings Plans alih-alih Instans Cadangan, tetapi kedua model komitmen dapat bekerja sama jika Anda sudah membeli Instans Cadangan.

Pertimbangkan contoh di mana Anda membeli Instans Cadangan, tetapi Anda ingin mulai mencoba opsi Savings Plans. Ada logika untuk kombinasi ini untuk diterapkan pada penagihan akhir Anda. Berikut hierarki yang dapat Anda terapkan pada: Akun AWS

1. Zonal Reserved Instance berlaku untuk akun yang memilikinya. Jika Instans Cadangan memiliki jam tersisa, itu berlaku untuk seluruh organisasi.
2. Instans Cadangan Regional yang tidak fleksibel untuk Windows berlaku untuk penggunaan yang cocok pada akun yang memilikinya. Apa pun yang tersisa diluncurkan ke seluruh organisasi.
3. Instans Cadangan Regional yang fleksibel ukuran berlaku untuk akun yang memilikinya (contoh terkecil dalam keluarga terlebih dahulu dan naik ke instance yang lebih besar), dan kemudian ke organisasi lainnya.
4. Instans Cadangan Regional berlaku untuk reservasi kapasitas sesuai permintaan yang tidak digunakan.
5. EC2 Instance Savings Plans berlaku dalam akun yang membelinya.
6. Compute Savings Plans berlaku dalam akun yang membelinya.

> **Note**
>
> Diskon dimulai dengan penggunaan yang menghasilkan diskon tertinggi dan kemudian turun ke diskon terkecil. Instans Windows secara tradisional memiliki potensi diskon yang lebih rendah daripada Linux untuk jenis instans yang paling umum (misalnya, T3, M6, dan C5). Ini berarti bahwa instance Linux menguntungkan lebih dari instance Windows dalam banyak kasus.

Grafik berikut menunjukkan harga setelah membagi Instans Cadangan dari Savings Plans. Baik Compute dan EC2 Instance Savings Plans berlaku untuk menjalankan instans terlebih dahulu dan kemudian ke Reservasi Kapasitas Sesuai Permintaan yang tidak digunakan.



Skenario pengoptimalan biaya

Bagian ini mencakup skenario pengoptimalan biaya untuk Host Khusus Amazon EC2 dan instans Amazon EC2 yang menggunakan model penagihan yang disertakan lisensi.

Host Khusus Amazon EC

Pertimbangkan skenario di mana Anda akan memigrasikan beban kerja Windows lokal ke. AWS Pusat data Anda memiliki server berikut:

- Dua server dengan 16 vCPU dan 128 GB RAM
- Dua server dengan 32 vCPU dan 164 GB RAM
- Satu server dengan 8 vCPU dan 64 GB RAM
- 16 server dengan vCPU dan 32 GB RAM

Selain itu, asumsikan bahwa Anda dapat membawa lisensi Anda sendiri AWS karena Anda memiliki cukup lisensi untuk dibawa. Tabel berikut menunjukkan instance server yang dapat Anda gunakan. AWS

Tipe instans	CPU	RAM	Jumlah
r5.4xlarge	16	128	2
r5.8xlarge	32	256	2
r5.2xlarge	8	64	1
r5.xlarge	4	32	16

Tipe instans	CPU	RAM	Jumlah
			21

Analisis menunjukkan bahwa 21 mesin virtual ini dapat didistribusikan di dua Host Khusus dengan host keluarga instans R5. Tabel berikut menunjukkan biaya dari dua Host Khusus ini.

Skenario sesuai permintaan Host Khusus	Pembayaran di muka	1 bulan	1 tahun	3 tahun	AWS Kalkulator Harga
Sesuai permintaan	Tidak ada	$10.123	$121.475	$364.392	AWS Kalkulator Harga estimasi
Savings Plans 1 tahun	Tidak ada	$7.447	$89.362	–	AWS Kalkulator Harga estimasi
Savings Plans 3 tahun	Tidak ada	$5.476	$65.712	$197,128	AWS Kalkulator Harga estimasi
Savings Plans 3 tahun dengan pembayaran dimuka	$84.438	$2.755	$117.499	$183.618	AWS Kalkulator Harga estimasi

Jika Anda memiliki server yang ingin Anda migrasi AWS, harga akhir untuk Savings Plans 1 tahun adalah $89.362, bukan $121.475 untuk harga sesuai permintaan. Ini merupakan diskon 26,5 persen setelah satu tahun. Jika Anda mempertimbangkan untuk tinggal AWS untuk jangka waktu yang lebih

lama, maka Anda dapat memilih Savings Plans 3 tahun untuk penghematan biaya yang lebih dalam. Pada akhir tiga tahun, Anda membayar $197,128, bukan $364.392. Ini menghasilkan penghematan 46 persen dari jumlah total setelah tiga tahun.

Instans Amazon EC2 dengan lisensi disertakan

Pertimbangkan skenario di mana Anda akan memigrasikan satu aplikasi tiga tingkat ke AWS, dan Anda ingin menggunakan lisensi yang disediakan oleh. AWS Selanjutnya, asumsikan bahwa aplikasi Anda bekerja dengan server berikut:

- Dua server web dengan dua v CPUs dan 4 GB RAM
- Dua server aplikasi dengan delapan v CPUs dan 16 GB RAM
- Dua server database dengan 16 v CPUs dan 64 GB RAM (menggunakan SQL Server Standard edition)

Tabel berikut menunjukkan instance server yang dapat Anda gunakan. AWS

Tipe instans	CPU	RAM	Jumlah
c5.large	2	4	2
c5.2xlarge	8	16	2
r5.2xlarge	8	64	2
			6 server

Tabel berikut menunjukkan biaya server ini di AWS.

Lisensi disertakan oleh AWS	Pembayaran di muka	1 bulan	1 tahun	3 tahun	AWS Kalkulator Harga
Sesuai permintaan	Tidak ada	$3.912	$46.950	$140,849	AWS Kalkulator Harga estimasi

Lisensi disertakan oleh AWS	Pembayaran di muka	1 bulan	1 tahun	3 tahun	AWS Kalkulator Harga
Savings Plans 1 tahun	Tidak ada	$3.466	$41.952		AWS Kalkulator Harga estimasi
Savings Plans 3 tahun tanpa pembayaran di muka	Tidak ada	$3,189	$38.264	$114.804	AWS Kalkulator Harga estimasi
Savings Plans 3 tahun dengan pembayaran dimuka	$112.110	Tidak ada	Tidak ada	Tidak ada	AWS Kalkulator Harga estimasi

Jika Anda ingin menjalankan server ini untuk lingkungan produksi (24/7) dengan harga sesuai permintaan, Anda membayar biaya bulanan sebesar $3.912. Membayar biaya bulanan ini setara dengan $46.950 setelah satu tahun dan total $140.849 setelah tiga tahun.

Jika Anda memilih Savings Plans 1 tahun tanpa pembayaran di muka, biaya bulanan berkurang menjadi $3,466. Pada akhir tahun pertama, Anda membayar $41.952. Ini adalah total discount sebesar 11 persen. Jika Anda memilih Savings Plans 3 tahun tanpa pembayaran di muka, biaya bulanan berkurang menjadi $3,189. Pada akhir tiga tahun, Anda membayar $114.804. Itu memberi Anda penghematan 18,5 persen.

Rekomendasi optimisasi biaya

Kedua skenario membantu Anda menghemat uang saat merencanakan dan memperkirakan beban kerja Anda. AWS Penting untuk diketahui bahwa diskon dalam skenario kedua kurang dibandingkan dengan skenario pertama. Dalam skenario kedua, harga lisensi sudah termasuk pada harga server

cloud. AWS tidak menawarkan diskon pada harga lisensi, tetapi Anda selalu dapat membawa lisensi Anda (dalam skenario tertentu) dan selalu AWS dapat menjamin compute/instance harga terbaik.

Kami menyarankan Anda melakukan hal berikut untuk mengontrol AWS pengeluaran Anda pada sumber daya komputasi dan instans:

- Rekomendasi akses
- Sesuaikan rekomendasi sesuai dengan kebutuhan Anda
- Tinjau komitmen per jam

Rekomendasi akses

Anda dapat menggunakan konsol Amazon EC2 untuk mengakses rekomendasi untuk Savings Plans Anda. Anda bahkan dapat mengunduh rekomendasi Anda untuk ditinjau nanti dalam format CSV. Untuk informasi selengkapnya, lihat Memantau Savings Plans Anda di dokumentasi Savings Plans.

Sesuaikan rekomendasi sesuai kebutuhan Anda

Buka konsol Amazon EC2, perluas bagian Instans, lalu pilih Savings Plans. Halaman ini menunjukkan contoh dan menghitung harga sebelum dan sesudah membuat rekomendasi. Anda juga dapat menyesuaikan faktor-faktor berikut untuk rekomendasi Anda:

- Jangka waktu — Misalnya, 1-3 tahun
- Opsi pembayaran — Misalnya, di muka, sebagian di muka, atau tidak ada di muka
- Sejarah — Misalnya, 7, 30 atau 60 hari terakhir

Tinjau komitmen per jam

Dengan menggunakan contoh yang sama, asumsikan Anda memiliki instance yang berjalan 24/7. Rekomendasi adalah menggunakan Savings Plans. Menurut ukurannya, Anda memiliki harga sesuai permintaan $120/jam. Anda memiliki opsi untuk melakukan $90/jam, tetapi ini dapat bervariasi tergantung pada opsi Wilayah, instance, dan pembelian. Dalam contoh ini, Anda dapat menghemat 25 persen dibandingkan dengan biaya sesuai permintaan. Anda juga dapat melacak pemanfaatan dan cakupan Anda, jika mereka berada di bawah ambang batas yang Anda tentukan, dan mengonfigurasi peringatan ketika anggaran akan berakhir.

Rekomendasi ulasan

Kami menyarankan Anda meninjau rekomendasi Savings Plans dengan cermat. AWS tidak akan mengubah apa pun tanpa izin Anda. Ini hanya rekomendasi dan terserah Anda untuk menerapkannya atau tidak.

Beli paket

Buka konsol Amazon EC2, perluas bagian Instans, lalu pilih Savings Plans. Kemudian, pilih Purchase Savings Plans. Berdasarkan kebutuhan Anda, Anda dapat memilih opsi berikut: istilah, Wilayah, keluarga contoh, komitmen per jam, opsi pembayaran, dan bahkan tanggal mulai. Anda dapat memilih dari Compute Savings Plans, EC2 Instance Savings Plans SageMaker , dan AI Savings Plans. Untuk informasi selengkapnya, lihat Purchasing Savings Plans di dokumentasi Savings Plans.

Dapatkan laporan pemanfaatan

Setelah Anda membeli Savings Plans, Anda bisa mendapatkan laporan pemanfaatan. Laporan ini membantu Anda memeriksa pemanfaatan Anda, melihat apakah paket yang dibeli cukup untuk menutupi dan memaksimalkan diskon, dan membatalkan atau menambahkan diskon baru. Laporan ini dapat diekspor ke format lain seperti CSV. Untuk informasi selengkapnya, lihat Menggunakan laporan pemanfaatan dalam dokumentasi Savings Plans.

Ikuti praktik terbaik pembelian

Kami menyarankan Anda mengikuti praktik terbaik ini sebelum membeli Savings Plans:

- Gunakan AWS Trusted Advisoruntuk menghapus sumber daya EC2 yang menganggur.
- Lakukan ukuran yang tepat sebelum pembelian Savings Plans.
- Tetapkan tarif per jam yang Anda pertahankan secara konsisten selama 30-60 hari.
- Beli komitmen untuk menutupi tarif per jam yang konsisten sesuai keinginan organisasi Anda. Pertimbangkan fluktuasi permintaan atau musim.
- Pilih anggaran Savings Plans tinjauan triwulanan untuk mempertahankan tingkat yang konsisten (misalnya, target cakupan 70 persen untuk cakupan Savings Plans). Jika tarif turun di bawah cakupan yang diinginkan, belilah Savings Plans tambahan sebagai true-up untuk memenuhi tujuan pertanggungan Anda.

Sumber daya tambahan

- Savings Plans untuk Instans Cadangan Amazon EC2 (Whitepaper)AWS
- Memahami bagaimana Savings Plans berlaku untuk AWS penggunaan Anda (dokumentasi Savings Plans)
- Mengumumkan penagihan per detik untuk Instans EC2 Windows Server dan SQL Server (dokumentasi)AWS
- AWS Seri Optimasi Biaya: Video Savings Plans | Amazon Web Services () YouTube

Pantau biaya menggunakan AWS alat

Ikhtisar

Visibilitas biaya adalah faktor kunci dalam mengoptimalkan biaya. AWS AWS memiliki sejumlah alat yang dapat Anda gunakan untuk memvisualisasikan biaya dan membuat peringatan sebagai reaksi terhadap biaya tersebut. Ini termasuk alat, seperti AWS Budgets, yang membantu Anda melacak dan melaporkan pengeluaran Anda. Bagian ini mencakup cara-cara khusus untuk memantau Windows Anda pada AWS pengeluaran, sehingga Anda dapat melacak dan bereaksi sesuai dengan kebutuhan anggaran Anda. Ini termasuk menambahkan tag yang diperlukan ke sumber daya Windows EC2 Anda. Tag ini memungkinkan Anda untuk memonitor Windows EC2 dan layanan Microsoft lainnya dengan benar dengan menggunakan AWS Budgets.

Dengan memantau pengeluaran dan membuat peringatan dengan AWS alat, Anda dapat lebih mengetahui tentang pengeluaran saat ini, pengeluaran yang diproyeksikan, dan anomali pengeluaran. Jika Anda menggunakan Savings Plans untuk membantu mengurangi harga instans EC2 per jam, sebaiknya Anda melihat keseluruhan pemanfaatan dan cakupan Savings Plans. Ini dapat membantu Anda memastikan bahwa Anda terus menyadari tabungan. Anda dapat menggunakan AWS Cost Explorer untuk melihat inventaris Savings Plans dan mendapatkan rekomendasi untuk Savings Plans tambahan berdasarkan penggunaan sebelumnya. Anda juga dapat melacak pengeluaran tertentu dengan menggunakan AWS Budgetsdan mengatur AWS Cost Anomaly Detection.

Rekomendasi optimisasi biaya

Kami menyarankan Anda mengambil langkah-langkah berikut untuk mengoptimalkan biaya Anda dengan menggunakan AWS Budgets, Cost Explorer, dan deteksi anomali:

- Menandai sumber daya Windows EC2
- Mengatur peringatan dengan menggunakan AWS Budgets
- Aktifkan Deteksi Anomali Biaya
- Dapatkan analisis pengeluaran waktu nyata
- Lihat pengeluaran yang disertakan lisensi untuk Windows dengan menggunakan Cost Explorer

Menandai sumber daya Windows EC2

Untuk memantau AWS pengeluaran Anda secara efektif, Anda harus menetapkan strategi penandaan untuk beban kerja yang ingin Anda pantau. Ini penting agar Anda dapat mengelompokkan sumber daya secara kategoris dan mendapatkan pemberitahuan tentang pengeluaran tertentu, sebagai lawan dari pengeluaran penggunaan umum. Anda dapat menggunakan sumber daya penandaan yang tidak hanya membantu biaya tetapi juga dapat digunakan untuk tujuan lain seperti AWS Systems Manager otomatisasi. Selain itu, kami menyarankan Anda menerapkan beberapa manajemen untuk tag yang diperlukan.

Untuk melacak pengeluaran Anda di AWS Budgets, Cost Explorer, dan Deteksi Anomali Biaya, Anda harus memastikan bahwa tag yang tepat tersedia. Anda dapat menggunakan tag untuk menyiapkan anggaran khusus untuk item yang cocok dengan tag tersebut sehingga Anda diberi tahu saat pengeluaran meningkat.

Misalnya, Anda dapat menggunakan tag sederhana seperti key=OS Value=Windows. Ini menempatkan semua instance Windows Anda bersama-sama ke dalam satu grup yang dapat Anda lacak pengeluarannya. Anda juga dapat menggunakan tag untuk item lain, seperti Systems Manager. Setelah Anda membuat tag, Anda harus mengaktifkan tag untuk pelacakan biaya. Pertimbangkan untuk menambahkan AWS Config aturan yang memantau tag yang dilampirkan ke sumber daya tertentu. AWS Config dapat memberi tahu Anda jika ada sumber daya yang berjalan yang tidak berisi tag yang sesuai, yang memberi Anda representasi akurat dari pengeluaran Windows EC2 Anda.

Setelah tag Anda di tempat, Anda dapat membuat anggaran khusus di AWS Billing. Ini memberikan visibilitas ke dalam pengeluaran Windows EC2 Anda. Anda dapat menetapkan anggaran harian atau anggaran bulanan.

Siapkan lansiran menggunakan AWS Budgets

Dalam skenario contoh ini, Anda membuat anggaran harian untuk Windows EC2. Ini adalah anggaran berulang yang menggunakan opsi penyesuaian otomatis untuk melacak pengeluaran

Anda dan menyesuaikan anggaran yang sesuai. Jika Anda memiliki lingkungan statis, Anda dapat menggunakan anggaran tetap sebagai gantinya. Pastikan untuk memilih rentang waktu dasar (misalnya, 30 hari).

1. Masuk ke Konsol Manajemen AWS dan buka AWS Cost Management konsol.
2. Di panel navigasi, pilih Budgets.
3. Di bagian atas halaman, pilih Buat anggaran.
4. Di bawah Pengaturan anggaran, pilih Sesuaikan (lanjutan).
5. Di bawah Jenis anggaran, pilih Anggaran biaya. Lalu, pilih Selanjutnya.
6. Di bawah Detail, untuk nama Anggaran, masukkan nama anggaran Anda. Misalnya, pengeluaran Windows EC2.
7. Di bawah Tetapkan jumlah anggaran, untuk Periode, pilih Harian.
8. Untuk jenis perpanjangan Anggaran, pilih Anggaran berulang untuk anggaran yang disetel ulang setelah periode anggaran.
9. Untuk tanggal Mulai, pilih tanggal atau periode mulai untuk mulai melacak jumlah anggaran Anda.
10. Untuk metode Budgeting, pilih Auto-adjustment (New).
11. Untuk rentang waktu Baseline, pilih Rentang kustom, lalu masukkan 30 hari.
12. Pilih Berikutnya.
13. Di bagian Lingkup anggaran, pilih Filter dimensi AWS biaya tertentu. Di sinilah tag digunakan untuk membuat dimensi yang tepat. AWS Budgets tidak mendukung Jenis Platform sebagai opsi di filternya. Untuk alasan ini, Anda harus menerapkan tag OS.
14. Pilih Tambahkan filter, lalu pilih opsi Tag dari Dimensi.
15. Pilih tag OS, lalu pilih nilai Windows untuk ini untuk membuat anggaran untuk tag.
16. Pilih Berikutnya.
17. Pada halaman Konfigurasi peringatan, pilih Tambahkan ambang peringatan. Di sini Anda mengatur dua peringatan: satu untuk ambang batas 50 persen dan satu untuk ambang batas 100 persen. Jika peringatan ambang batas 50 persen dilanggar sebelum titik tengah dalam sebulan, itu akan memberikan peringatan. Dengan begitu, Anda dapat memeriksa apakah pengeluaran Anda lebih dari yang diharapkan dan bereaksi sebelum mencapai akhir bulan.
18. Untuk Threshold, masukkan 50 dan pilih% dari jumlah yang dianggarkan.
19. Untuk Trigger, pilih Actual.
20. Untuk penerima email, masukkan alamat email. Tambahkan peringatan lain untuk ambang batas 100.

> **Note**
>
> Contoh ini menggunakan pemberitahuan email untuk peringatan, tetapi Anda juga dapat menggunakan pendekatan lain, seperti Slack.

Aktifkan Deteksi Anomali Biaya

Anda dapat menggunakan tag biaya Anda untuk mengatur peringatan pengeluaran yang merupakan anomali. Misalnya, Anda dapat menggunakan AWS Cost Anomaly Detectionuntuk membuat monitor untuk pengeluaran Anda dan mendapatkan peringatan ketika sistem mendeteksi pengeluaran abnormal di akun Anda.

Untuk menyiapkan monitor dan peringatan untuk tag key=OS dan Value=Windows yang Anda buat sebelumnya, lakukan hal berikut:

1. Masuk ke Konsol Manajemen AWS dan buka AWS Cost Management konsol.
2. Di panel navigasi, memilih Deteksi Anomali Biaya.
3. Pilih tab Monitor biaya, lalu pilih Buat monitor.
4. Pada Langkah 1, pilih Tag Alokasi Biaya sebagai jenis monitor Anda.
5. Untuk kunci Tag Alokasi Biaya, pilih pembelanjaan Windows EC2.
6. Untuk nilai Tag Alokasi Biaya, pilih Windows.
7. Untuk Nama monitor Anda, masukkan pengeluaran Windows EC2.
8. Pilih Berikutnya.
9. Untuk membuat langganan peringatan, pilih Buat langganan baru. Jika Anda memiliki langganan yang ada, pilih Memilih langganan yang ada.
10. Untuk nama Langganan, masukkan anomali pengeluaran Windows EC2.
11. Untuk Frekuensi peringatan, pilih Ringkasan harian.
12. Untuk penerima Pemberitahuan, masukkan alamat email Anda.
13. Pilih Tambahkan ambang batas. Untuk Threshold, masukkan 10 dan kemudian pilih persen di atas kecepatan yang diharapkan.
14. Pilih Buat monitor.

Dapatkan tampilan real-time tentang pengeluaran

Peringatan adalah alat yang berguna untuk memantau pengeluaran Windows EC2 Anda, tetapi Anda harus menggunakan Cost Explorer jika Anda ingin tampilan real-time ke dalam pengeluaran. Tonton video ini untuk mempelajari bagaimana Cost Explorer memungkinkan Anda menganalisis dan mengurangi biaya EC2 Anda. Untuk informasi lebih lanjut, tonton video AWS Mendukung Anda | Memahami dan Mengurangi Biaya EC2 Anda di YouTube.

Lihat pengeluaran yang termasuk lisensi untuk Windows

Anda dapat melihat pengeluaran EC2 Windows di akun Anda dengan menggunakan Cost Explorer. Untuk melihat pengeluaran yang disertakan lisensi untuk Windows, Anda harus mengatur filter yang benar berikut di Cost Explorer:

- Untuk Platform, pilih Windows (Amazon VPC). Untuk operasi API, pilih: 0002. RunInstance Ini adalah AWS Billing kode untuk instance Windows EC2 yang disertakan lisensi.
- Jika Anda ingin melihat pengeluaran instans BYOL Anda, ubahRunInstance: 0002 menjadi: 0800. RunInstance Ini adalah kode penagihan untuk Windows EC2 BYOL.

Dengan visibilitas ini di Cost Explorer, Anda dapat dengan cepat memfilter biaya Anda ke persis apa yang Anda belanjakan untuk Windows EC2. Jika Anda ingin menyelam lebih dalam ke AWS pengeluaran Anda, Anda dapat menggunakan AWS Cost and Usage Report untuk memfilter ke pengeluaran di tingkat instans individu. Anda juga dapat membuat laporan yang dapat divisualisasikan di Amazon Quick dan membuat dasbor yang disesuaikan.

Untuk informasi lebih lanjut, tonton video AWS Mendukung Anda - Memvisualisasikan Laporan Biaya dan Penggunaan Anda di YouTube.

Sumber daya tambahan

- Menyiapkan tag yang diperlukan dengan AWS Config (AWS Config dokumentasi)
- AWS Budgets Tutorial - Setup Alerts untuk AWS Billing | Amazon Web Services () YouTube
- AWS Cost and Usage Report Query Library (AWS Well-Architected Labs)

SQL Server

Pelanggan telah menjalankan beban kerja Microsoft AWS selama lebih dari 15 tahun, lebih lama dari penyedia cloud lainnya. Ini sebagian besar karena AWS memiliki pengalaman paling banyak dengan aplikasi Microsoft di cloud dan menawarkan platform terbaik untuk Windows Server dan Microsoft SQL Server di bidang berikut:

- Kinerja dan keandalan yang lebih tinggi
- Layanan keamanan dan identitas yang lebih besar
- Lebih banyak dukungan migrasi
- Kemampuan terluas dan terdalam
- Total biaya kepemilikan (TCO) yang lebih rendah
- Opsi lisensi yang fleksibel

AWS mendukung semua yang diperlukan untuk membangun dan menjalankan aplikasi Windows yang mengandalkan SQL Server, termasuk Active Directory, .NET, SQL Server, Windows desktop sebagai layanan, dan semua versi Windows Server yang didukung. Dengan keahlian yang telah terbukti, AWS dapat membantu Anda dengan mudah mengangkat dan menggeser, memfaktorkan ulang, atau bahkan memodernisasi beban kerja Windows Anda.

Bagian panduan ini mencakup topik-topik berikut:

- Pilih ketersediaan tinggi dan solusi pemulihan bencana
- Memahami lisensi SQL Server
- Pilih instans EC2 yang tepat untuk beban kerja SQL Server
- Mengkonsolidasikan contoh
- Bandingkan edisi SQL Server
- Evaluasi edisi Pengembang SQL Server
- Mengevaluasi SQL Server di Linux
- Optimalkan strategi pencadangan SQL Server
- Memodernisasi database SQL Server
- Optimalkan penyimpanan untuk SQL Server
- Optimalkan lisensi SQL Server dengan menggunakan Compute Optimizer
- Optimalkan ukuran SQL Server dengan menggunakan Compute Optimizer

- Meninjau Trusted Advisor rekomendasi untuk beban kerja SQL Server

Pilih ketersediaan tinggi dan solusi pemulihan bencana

Ikhtisar

Kami menyarankan Anda merancang arsitektur untuk penerapan SQL Server Anda AWS yang memenuhi kebutuhan bisnis Anda sambil juga memenuhi tujuan pemulihan bencana (DR) Anda, termasuk tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO) Anda. Solusi berikut dapat membantu Anda merancang arsitektur yang tepat untuk SQL Server di Amazon Elastic Compute Cloud (Amazon EC2) sekaligus mengoptimalkan biaya untuk beban kerja SQL Server Anda.

- Grup ketersediaan SQL Server Always On — Grup ketersediaan SQL Server Always On menyediakan ketersediaan tinggi dan pemulihan bencana (HA/DR) solutions for SQL Server databases. An availability group consists of a set of user databases that fail over together. Always On availability groups also provide redundancy at the database level, but don't require shared storage—each replica has its own local storage. You can deploy this feature as an HA/DRsolusi. Untuk informasi selengkapnya, lihat Apa itu grup ketersediaan Selalu Aktif? dalam dokumentasi Microsoft.
- SQL Server Always On failover cluster instance (FCI) - SQL Server Selalu Aktif FCIs menggunakan Windows Server Failover Clustering (WSFC) untuk menyediakan HA pada tingkat instans SQL Server. FCIs membutuhkan penyimpanan bersama untuk meng-host database. Anda dapat menggunakan penyimpanan blok bersama atau penyimpanan file bersama. Misalnya, Anda dapat menggunakan Amazon FSx untuk Windows File Server atau Amazon FSx untuk NetApp ONTAP sebagai solusi penyimpanan bersama dengan beberapa Availability Zone. Untuk informasi selengkapnya, lihat Selalu Aktif Instans Kluster Failover (SQL Server) di dokumentasi Microsoft.
- SIOS DataKeeper - SIOS DataKeeper dapat membantu Anda memenuhi persyaratan HA dan DR dengan mengaktifkan SQL Server FCI yang mencakup Availability Zone dan. Wilayah AWS SIOS DataKeeper membuat SAN virtual berkerumun dengan menggunakan volume Amazon Elastic Block Store (Amazon EBS) lokal dan menggunakan replikasi sinkron antara Availability Zones untuk HA, saat menggunakan replikasi asinkron antar Wilayah dan untuk pemulihan bencana. Untuk informasi selengkapnya, lihat Perlindungan Ketersediaan Tinggi untuk Aplikasi Windows dalam dokumentasi SIOS.
- Grup ketersediaan terdistribusi — Grup ketersediaan terdistribusi adalah jenis grup ketersediaan khusus yang mencakup dua grup ketersediaan Selalu Aktif yang terpisah. Grup ketersediaan dapat berada di dua Wilayah terpisah (misalnya, `us-east-1` dan`us-west-1`). Anda dapat

menganggap grup ketersediaan terdistribusi sebagai grup ketersediaan grup ketersediaan karena grup ketersediaan Selalu Aktif yang mendasari dikonfigurasi pada dua kluster WSFC yang berbeda. Edisi SQL Server Enterprise diperlukan untuk menyebarkan grup ketersediaan terdistribusi. Untuk informasi selengkapnya, lihat Grup ketersediaan terdistribusi di dokumentasi Microsoft.

- Pengiriman log - Anda dapat menerapkan pengiriman log untuk melindungi basis data Anda di beberapa Wilayah, jika suatu Wilayah terkena dampak dan menjadi tidak tersedia. Bergantung pada transaksi dan frekuensi pengiriman log, Anda dapat mencapai RPO dan RTO dalam beberapa menit. Untuk informasi selengkapnya, lihat Tentang Pengiriman Log (SQL Server) di dokumentasi Microsoft.
- AWS Elastic Disaster Recovery— Elastic Disaster Recovery adalah aplikasi perangkat lunak sebagai layanan (SaaS) yang mengelola replikasi server dari infrastruktur apa pun AWS untuk tujuan DR. Anda juga dapat menggunakan Elastic Disaster Recovery untuk mereplikasi SQL Server di seluruh Wilayah. Elastic Disaster Recovery adalah solusi berbasis agen yang mereplikasi seluruh mesin virtual, termasuk sistem operasi, semua aplikasi yang diinstal, dan semua database ke area pementasan. Untuk informasi lebih lanjut, lihat Apa itu Pemulihan Bencana Elastis? dalam dokumentasi Pemulihan Bencana Elastis.
- AWS Database Migration Service (AWS DMS) — AWS DMS mendukung migrasi langsung data ke dan dari AWS, termasuk Wilayah yang berbeda. Anda dapat menggunakan fitur ini untuk menyiapkan instance SQL Server terpisah di Wilayah yang berbeda untuk berfungsi sebagai database pemulihan bencana. Untuk informasi lebih lanjut, lihat Apa itu AWS Database Migration Service? dalam AWS DMS dokumentasi.

SQL Server Selalu Aktif pada grup ketersediaan

Jika Anda menggunakan edisi SQL Server Enterprise hanya untuk grup ketersediaan Always On ketersediaan tinggi, maka Anda dapat menurunkan versi ke edisi Standar SQL Server dengan memanfaatkan grup ketersediaan dasar. Anda dapat mengurangi biaya dari 65-75 persen dengan menggunakan grup ketersediaan dasar, bukan grup ketersediaan Selalu Aktif.

> **Note**
>
> Untuk informasi tambahan tentang perbedaan biaya antara edisi SQL Server yang berbeda, lihat bagian Bandingkan edisi SQL Server dari panduan ini.

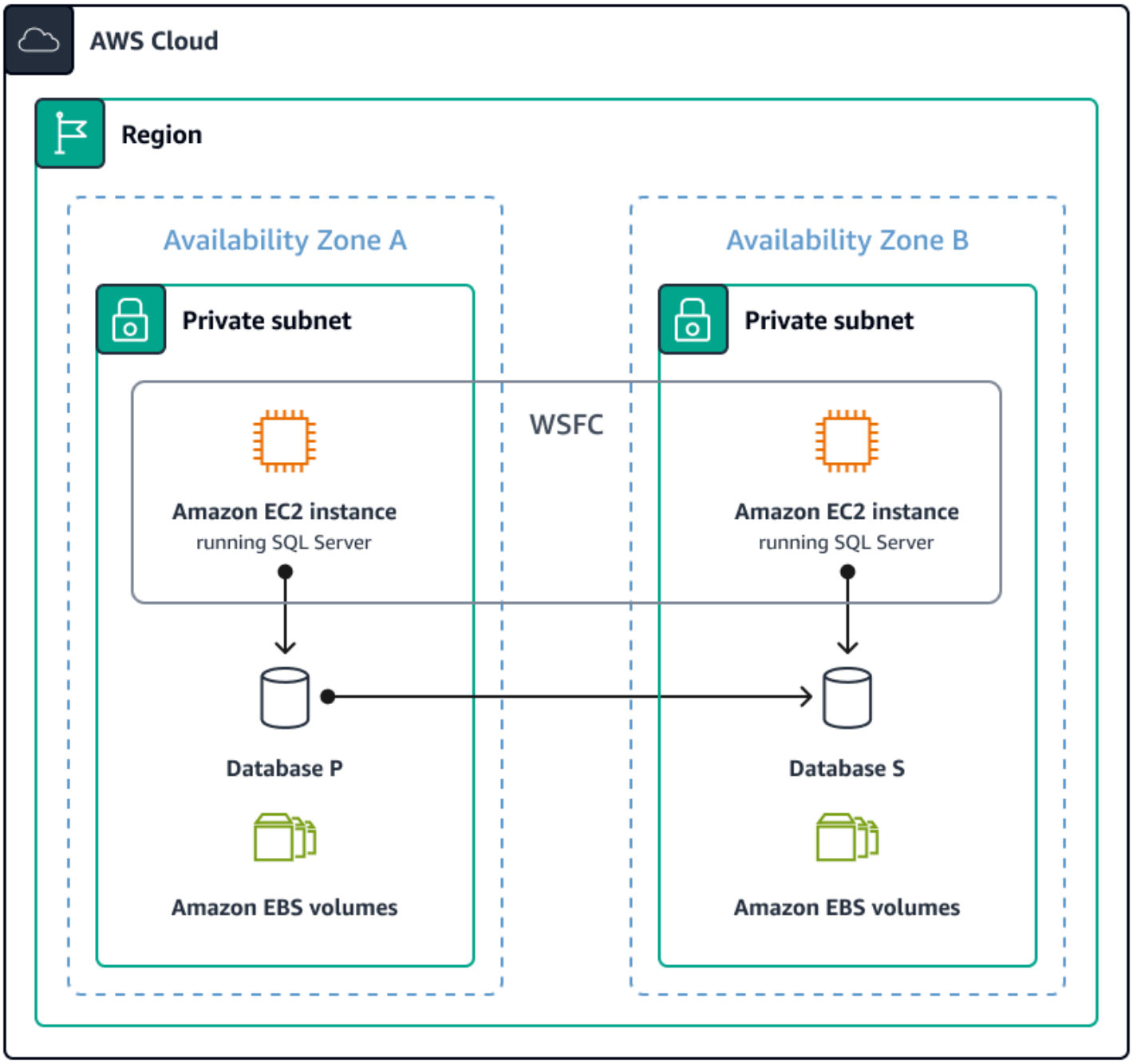
Fitur

- Tersedia dalam edisi Standar SQL Server
- Batas dua replika (primer dan sekunder)
- Tidak ada akses baca pada replika sekunder
- Tidak ada pemeriksaan integritas pada replika sekunder

Batasan

- Support untuk hanya satu database ketersediaan per grup ketersediaan
- Grup ketersediaan dasar tidak dapat menjadi bagian dari grup ketersediaan terdistribusi

Diagram berikut menunjukkan contoh arsitektur untuk solusi Windows Server Failover Cluster.



SQL Server Selalu Pada instance cluster failover

Anda dapat menggunakan instance failover cluster (FCIs) untuk memastikan operasi database berkelanjutan sambil meminimalkan waktu henti dan mengurangi risiko kehilangan data. FCIs menawarkan solusi yang andal jika Anda mencari ketersediaan tinggi untuk database SQL Server Anda tanpa konfigurasi replika baca.

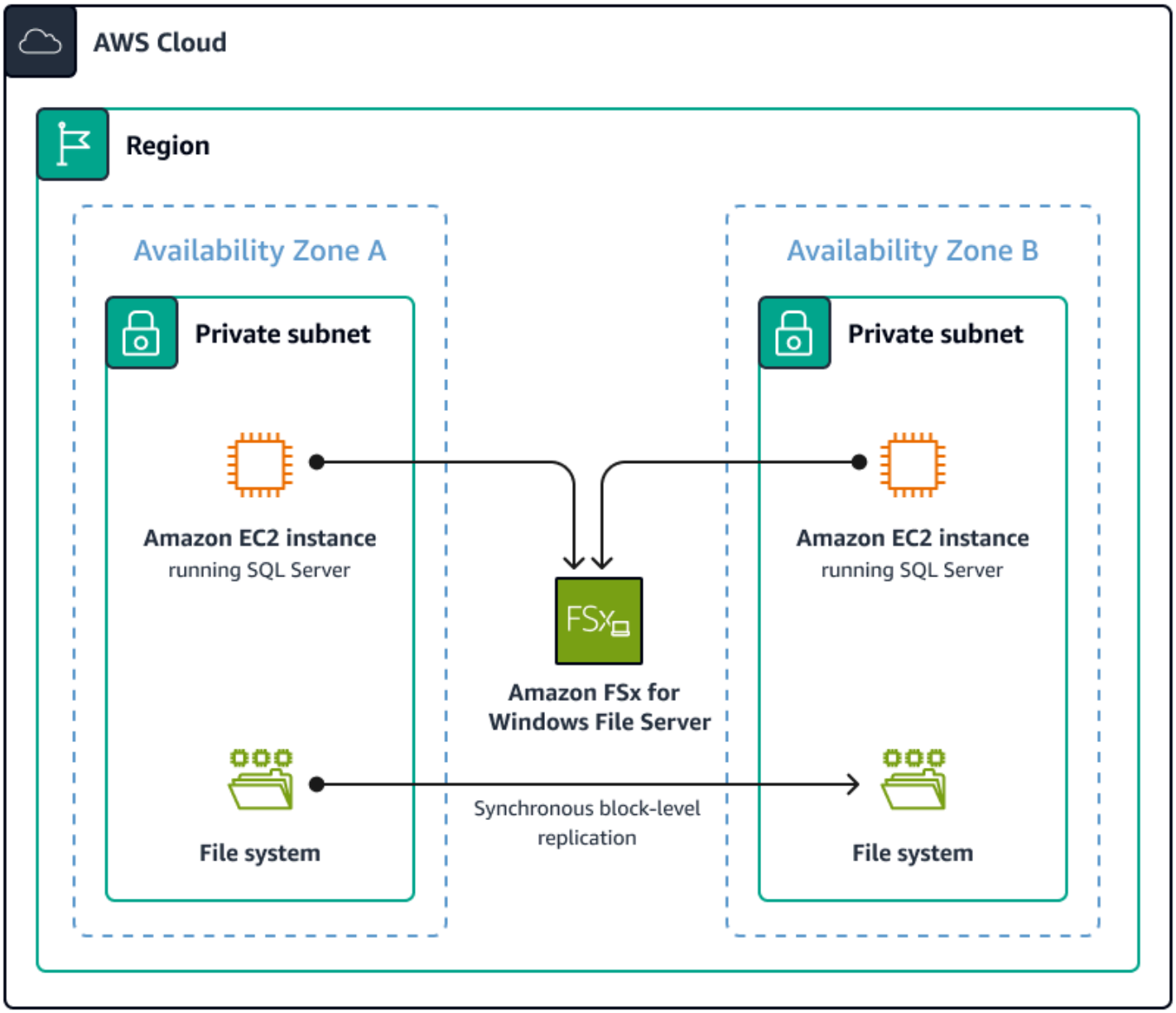
Tidak seperti grup ketersediaan, FCIs dapat memberikan solusi failover yang dapat diandalkan tanpa memerlukan edisi SQL Server Enterprise. Sebaliknya, hanya FCIs memerlukan lisensi edisi Standar SQL Server. Anda dapat menggunakan FCIs untuk mengurangi biaya lisensi SQL Server sebesar 65-75 persen.

> **Note**
>
> Untuk informasi tambahan tentang perbedaan biaya antara edisi SQL Server, lihat bagian Bandingkan edisi SQL Server dari panduan ini.

Pertimbangkan hal berikut:

- Amazon FSx untuk Windows File Server menawarkan solusi canggih untuk memenuhi persyaratan penyimpanan bersama SQL Server FCI Anda. Anda dapat menggunakan FSx Windows File Server untuk menghindari kebutuhan untuk membeli lisensi untuk solusi replikasi penyimpanan dan mengelola penyimpanan bersama sendiri. Hal ini dapat menghasilkan penghematan biaya yang signifikan sebesar 30-40 persen. Untuk informasi selengkapnya, lihat posting Menyederhanakan penerapan ketersediaan tinggi Microsoft SQL Server menggunakan FSx Amazon untuk Windows File Server di AWS Blog Penyimpanan.
- Dengan ringkasan manfaat Jaminan Perangkat Lunak (PDF yang dapat diunduh) dan model Bring Your Own License (BYOL), Anda dapat memanfaatkan manfaat failover pasif, selama server sekunder pasif. Ini menghasilkan penghematan biaya untuk lisensi SQL karena Anda tidak perlu memberikan lisensi ke node pasif cluster.

Diagram berikut menunjukkan contoh arsitektur untuk SQL Server FCI dengan menggunakan FSx untuk Windows File Server.



SIOS DataKeeper

Kami menyarankan Anda mempertimbangkan persyaratan penyimpanan bersama jika Anda berencana untuk menggunakan SQL Server FCIs . AWS Instalasi lokal tradisional biasanya menggunakan jaringan area penyimpanan (SAN) untuk memenuhi persyaratan penyimpanan bersama, tetapi ini bukan opsi yang layak. AWS Amazon FSx untuk Windows File Server adalah solusi penyimpanan yang direkomendasikan untuk SQL Server FCI aktif AWS, tetapi memiliki keterbatasan yang mencegah penambahan server cluster yang berbeda. Wilayah AWS

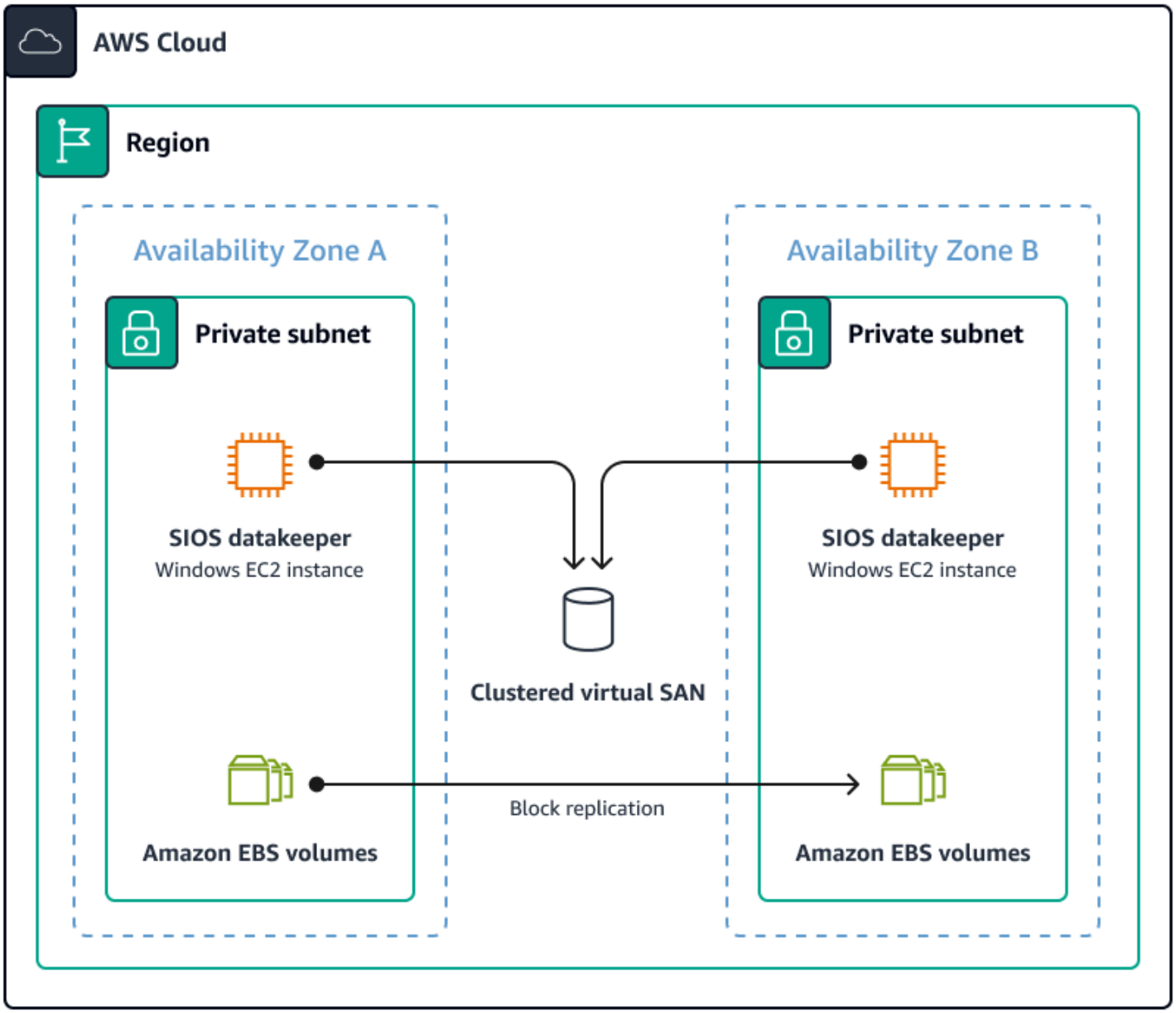
Anda dapat menggunakan SIOS DataKeeper untuk membuat SQL Server FCI yang mencakup Availability Zone dan Regions sekaligus mengurangi biaya sebesar 58—71 persen. SIOS DataKeeper dapat membantu Anda mencapai manfaat ketersediaan tinggi dari FCI. Ini menjadikan SIOS solusi DataKeeper yang hemat biaya dan dapat diandalkan untuk organisasi.

Pertimbangkan manfaat tambahan berikut menggunakan SIOS DataKeeper:

- SIOS DataKeeper membuat SAN virtual berkerumun dengan menggunakan volume EBS lokal dan menggunakan replikasi sinkron antara Availability Zones untuk ketersediaan tinggi. Untuk pemulihan bencana, SIOS DataKeeper menggunakan replikasi asinkron antar Wilayah.
- SIOS DataKeeper menyediakan fitur pengelompokan kelas perusahaan dengan menggunakan edisi Standar SQL Server. Ini mengurangi biaya lisensi SQL Server antara 65-75 persen dibandingkan dengan menerapkan ketersediaan tinggi dengan grup ketersediaan SQL Server Always On yang menggunakan edisi SQL Server Enterprise. Dengan SIOS DataKeeper, Anda dapat membuat lingkungan SQL Server yang sangat tersedia, fleksibel, dan hemat biaya yang memenuhi kebutuhan organisasi Anda.

> **Note**
>
> Untuk informasi tambahan tentang perbedaan biaya antara edisi SQL Server, lihat bagian Bandingkan edisi SQL Server dari panduan ini.

Diagram berikut menunjukkan contoh arsitektur untuk SQL Server FCI menggunakan solusi SAN virtual berkerumun.

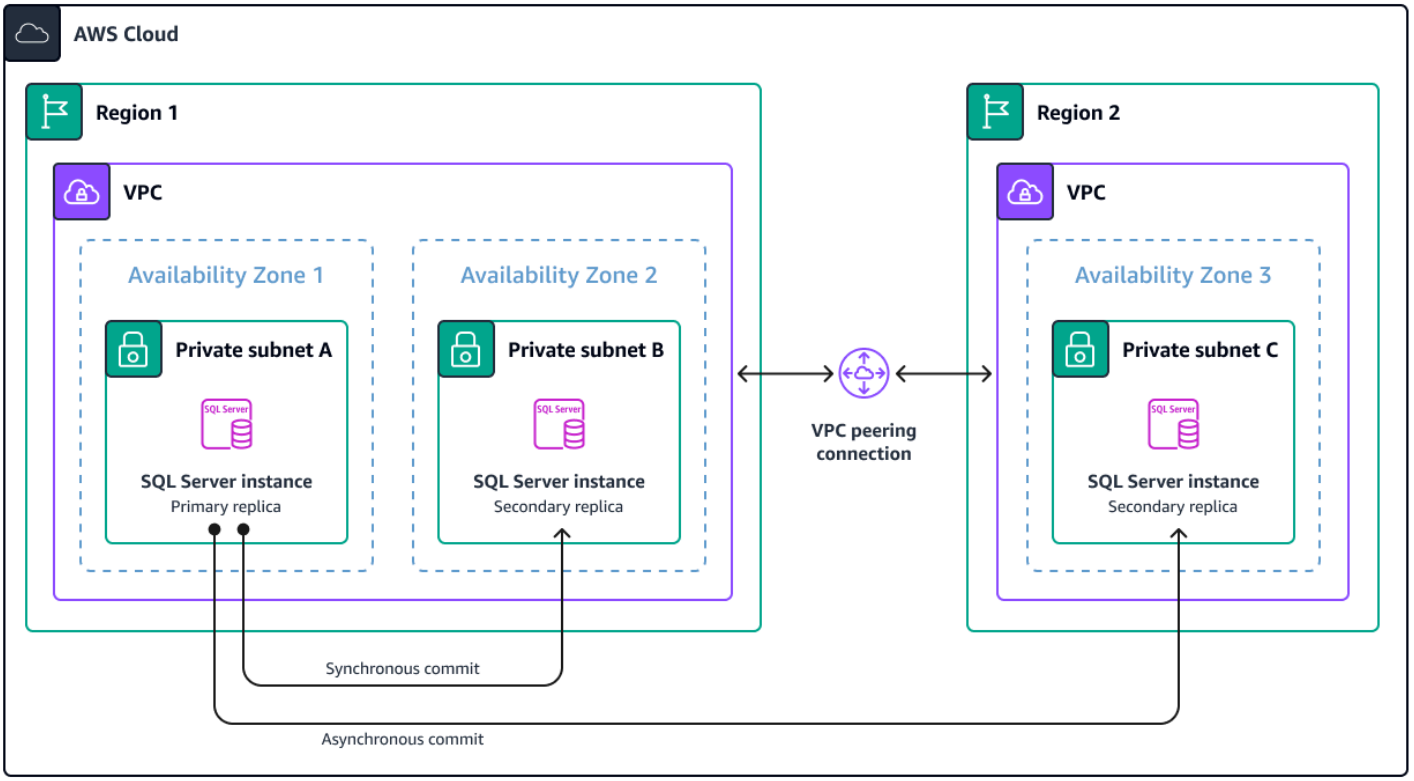


Selalu Aktif pada grup ketersediaan

Anda dapat menggunakan grup ketersediaan Selalu Aktif untuk tujuan ketersediaan tinggi dan pemulihan bencana. Anda dapat mencapai ketersediaan tinggi dengan menerapkan SQL Server di dua Availability Zone dalam satu Region. Anda dapat mencapai pemulihan bencana dengan memperluas grup ketersediaan di seluruh Wilayah.

Diagram berikut menunjukkan contoh arsitektur untuk solusi berdasarkan grup ketersediaan Always On. Replika di Region 1 diagram menggunakan Synchronous Commit, yang menyediakan failover

otomatis dari grup ketersediaan. Replika di Wilayah 2 menggunakan Komit Asinkron, yang akan memerlukan failover manual dari grup ketersediaan.



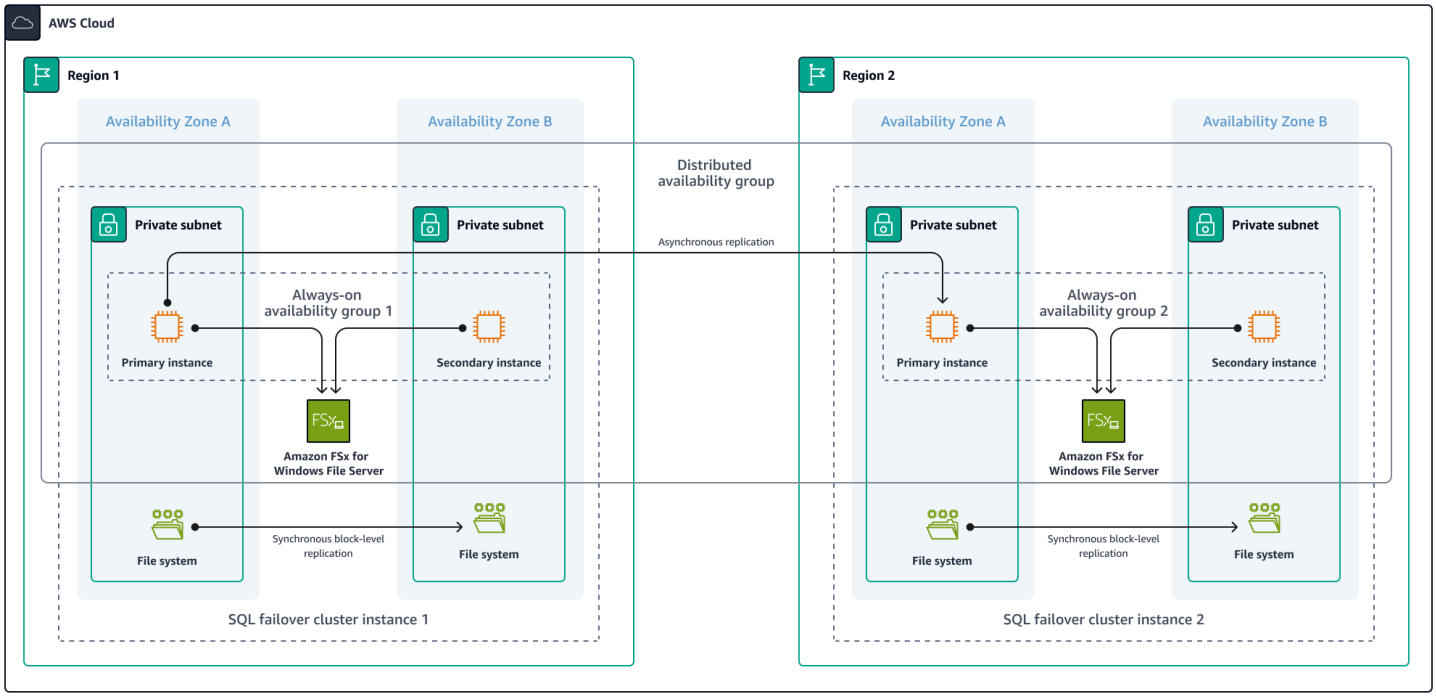
Grup ketersediaan terdistribusi

Untuk penerapan SQL Server yang sangat penting di mana Anda tidak dapat berkompromi pada keandalan atau pemulihan bencana, kami merekomendasikan pendekatan Multi-wilayah. Mendistribusikan grup ketersediaan Anda di beberapa Wilayah adalah solusi paling tangguh untuk menjaga kelangsungan bisnis dan meminimalkan waktu henti.

Arsitektur ini memanfaatkan sepenuhnya kemampuan Amazon FSx untuk Windows File Server, termasuk penyimpanan bersama, replikasi tingkat blok sinkron, dan SQL Server. FCIs Kemampuan ini memungkinkan Anda untuk membuat lingkungan SQL Server yang sangat tersedia yang mencakup beberapa Availability Zone. Dengan mereplikasi pengaturan ini di Wilayah lain, Anda mendapatkan sistem yang sepenuhnya berlebihan yang dapat menangani gangguan yang paling parah sekalipun. Apa yang membedakan solusi ini adalah tingkat fleksibilitas dan keamanan yang diberikannya. Arsitektur domain-independen dari grup ketersediaan terdistribusi memungkinkan server cluster Windows yang mendasari untuk bergabung dengan domain Active Directory yang berbeda, sementara otentikasi berbasis sertifikat memastikan perlindungan maksimum untuk lingkungan SQL Server Anda dan menyediakan persyaratan RTO dan RPO yang tinggi untuk

strategi DR Multi-wilayah. Untuk informasi tentang membangun arsitektur Multi-wilayah, lihat Catatan Lapangan: Membangun Arsitektur Multi-Wilayah untuk SQL Server menggunakan FCI dan Grup Ketersediaan Terdistribusi di Blog Arsitektur. AWS

Diagram berikut menunjukkan contoh arsitektur untuk solusi Multi-region menggunakan grup ketersediaan terdistribusi.



Pengiriman log

Pengiriman log adalah metode yang terbukti, andal, dan hemat biaya untuk melindungi basis data Anda di seluruh Wilayah jika terjadi pemadaman yang tidak terduga. Organizations telah menggunakan pengiriman log untuk melindungi data mereka selama beberapa dekade.

Jika Anda menerapkan pengiriman log AWS, Anda dapat mencapai RPO dan RTO dalam hitungan menit, tergantung pada frekuensi transaksi dan pekerjaan pengiriman log. Jika suatu Wilayah tidak dapat diakses, pengiriman log membuat data Anda aman dan dapat dipulihkan.

Pertimbangkan manfaat tambahan berikut menggunakan pengiriman log:

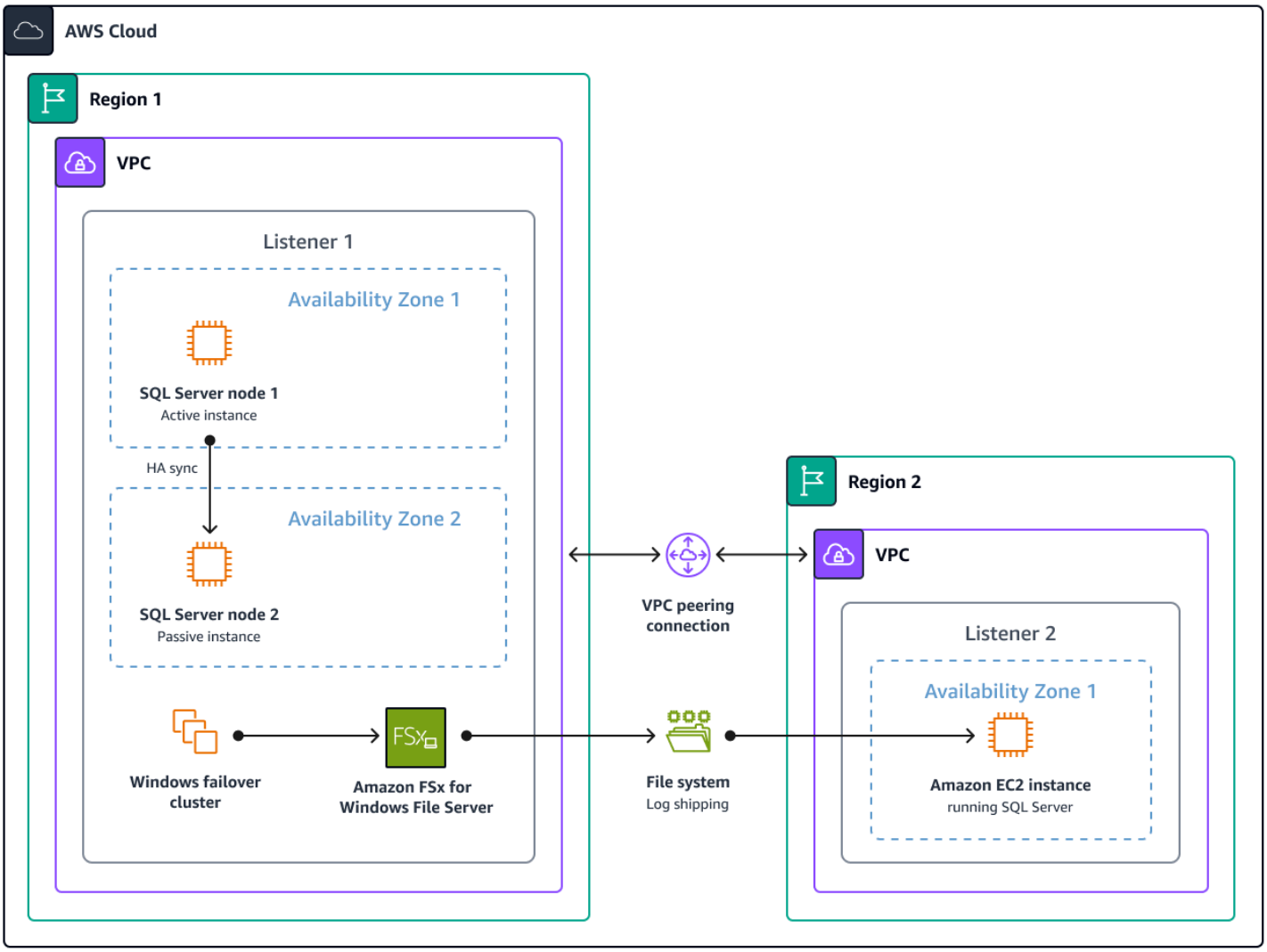
- Kurangi biaya dan penuhi kebutuhan bisnis Anda dengan menggunakan pengiriman log untuk ketahanan pemulihan bencana di seluruh Wilayah. Pengiriman log mengurangi TCO Anda karena Anda hanya memerlukan lisensi edisi SQL Server Standard atau SQL Server Web edition.

- Hapus biaya lisensi dari recovery/passive server bencana dengan menggunakan pengiriman log dengan Jaminan Perangkat Lunak aktif. Hanya primary/active SQL Server yang perlu dilisensikan saat Anda menggunakan pengiriman log dengan Jaminan Perangkat Lunak.
- Kurangi biaya lisensi SQL Server sebesar 65-75 persen dengan menghapus kebutuhan edisi SQL Server Enterprise untuk menyiapkan grup ketersediaan terdistribusi antar Wilayah. Anda dapat melakukan ini dengan menggunakan edisi Standar SQL Server dan SQL Server FCIs dikombinasikan dengan pengiriman log untuk memenuhi persyaratan pemulihan bencana Anda.

> **Note**
>
> Untuk informasi tambahan tentang perbedaan biaya antara edisi SQL Server, lihat bagian Bandingkan edisi SQL Server dari panduan ini.

Untuk informasi selengkapnya, lihat Memperluas SQL Server DR menggunakan pengiriman log untuk SQL Server FCI dengan konfigurasi Amazon FSx untuk Windows di Blog Arsitektur. AWS

Diagram berikut menunjukkan contoh arsitektur untuk solusi pengiriman log.

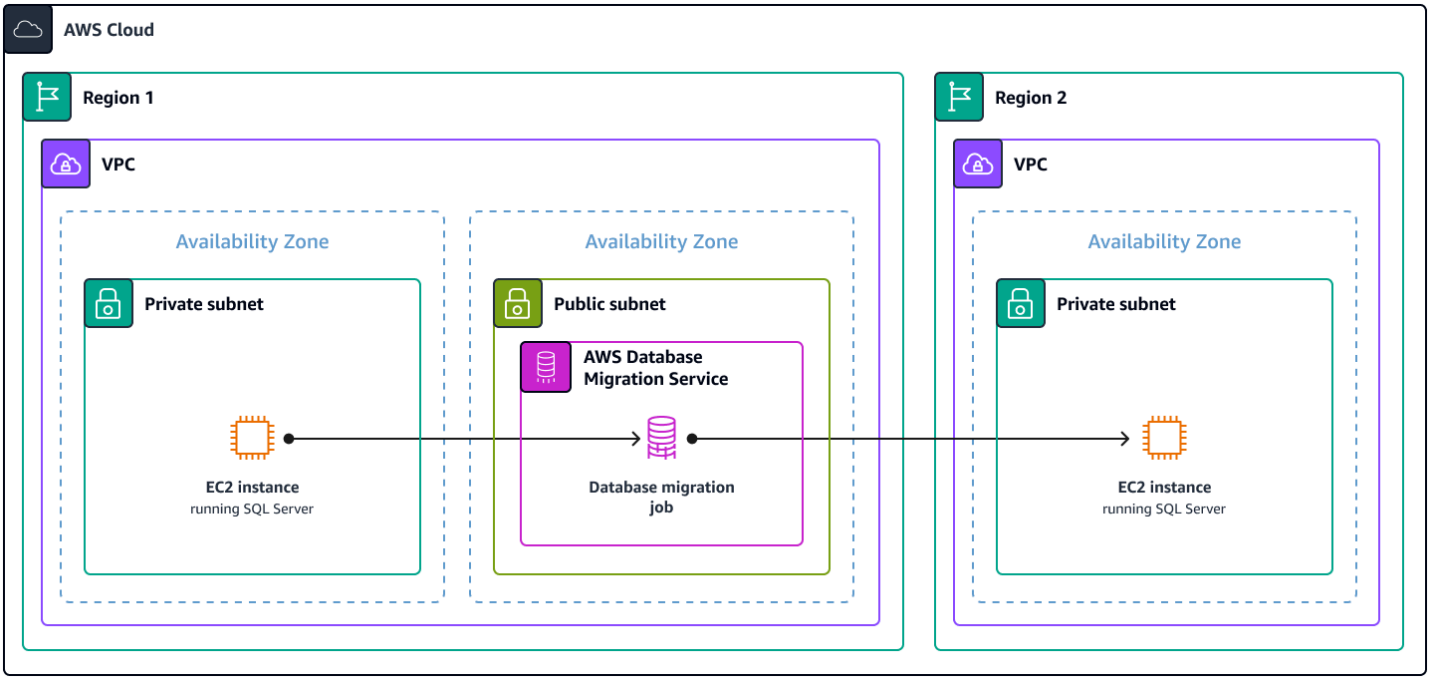


AWS Database Migration Service

Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk merancang HA/DR solusi berdasarkan kebutuhan aplikasi Anda. AWS DMS memungkinkan Anda untuk dengan mudah menyalin data ke database SQL Server sekunder di Wilayah yang sama (HA) atau lintas Wilayah (DR). Pendekatan ini secara teknis baik, dan memungkinkan Anda memaksimalkan investasi Anda dalam AWS infrastruktur sambil mengoptimalkan penggunaan sumber daya Anda.

AWS DMS adalah layanan yang hemat biaya. Anda hanya dikenakan biaya untuk sumber daya CPU yang digunakan selama proses transfer dan penyimpanan log tambahan apa pun. Ini berarti Anda dapat memperoleh manfaat dari solusi ini tanpa menimbulkan biaya tambahan yang signifikan. Anda dapat menggunakannya AWS DMS untuk memastikan data Anda tersedia dan dapat diakses, sambil meminimalkan biaya yang terkait dengan lisensi dan penggunaan sumber daya.

Diagram berikut menunjukkan contoh arsitektur untuk solusi berdasarkan AWS DMS.



AWS Elastic Disaster Recovery

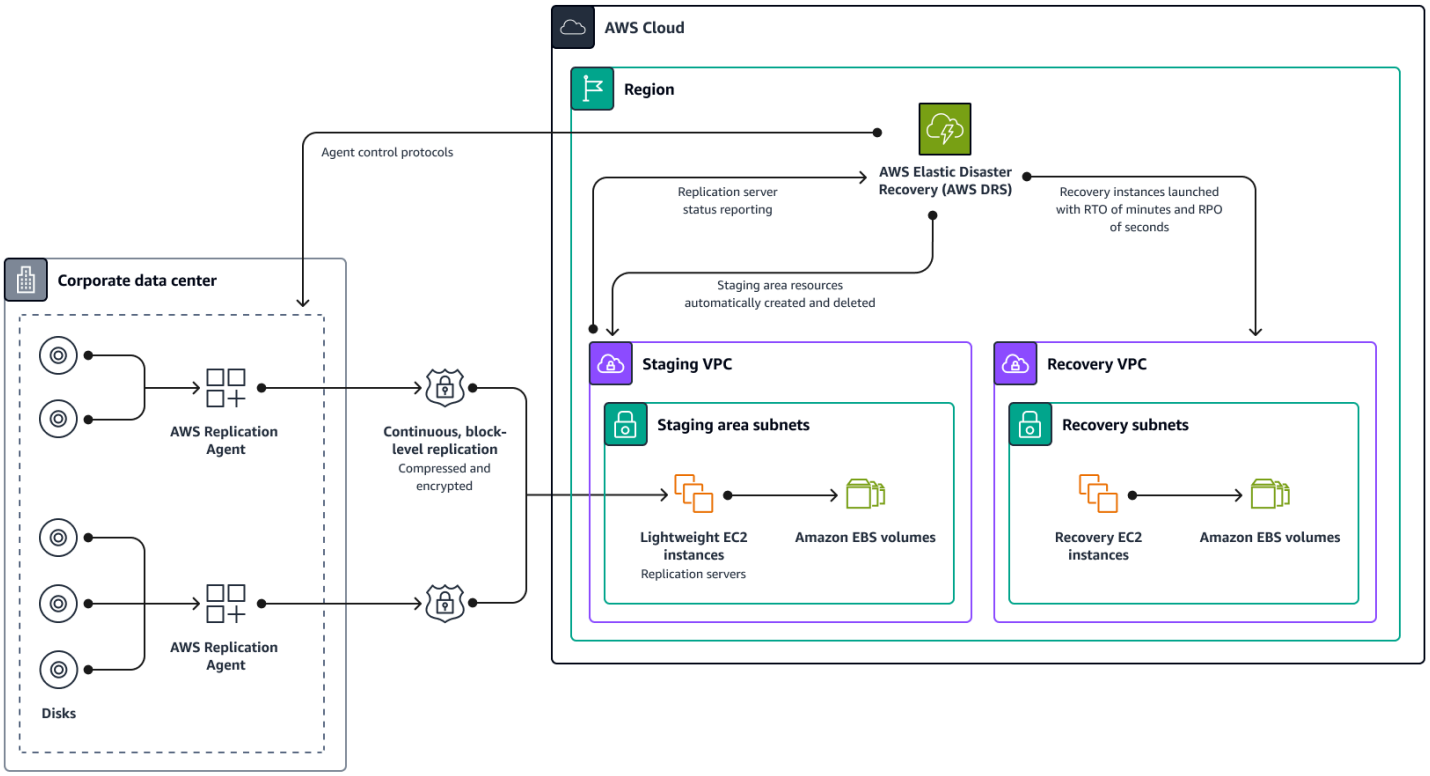
Beberapa organisasi harus memastikan bahwa semua aplikasi bisnis penting memiliki rencana pemulihan bencana. Di masa lalu, banyak dari organisasi ini berinvestasi besar-besaran dalam solusi pemulihan bencana tradisional, yang mengharuskan Anda untuk membangun dan memelihara seluruh infrastruktur duplikat. Pendekatan ini mahal, memakan waktu, dan sulit untuk skala.

Sekarang, Anda dapat menggunakan AWS Elastic Disaster Recovery untuk menghilangkan kebutuhan untuk pra-membangun infrastruktur pemulihan bencana. Mesin pemulihan bencana tidak dimulai di Elastic Disaster Recovery sampai diperlukan, jadi Anda hanya membayar untuk apa yang Anda gunakan saat Anda membutuhkannya. Ini berarti Anda dapat secara signifikan mengurangi lisensi perangkat lunak dan biaya komputasi berkinerja tinggi.

Selain itu, area pementasan untuk solusi pemulihan bencana berisi volume Amazon Elastic Block Store (Amazon EBS) berbiaya rendah. Volume EBS semakin mengurangi biaya penyediaan sumber daya duplikat. Ini memungkinkan Anda untuk mengurangi biaya pemulihan bencana secara keseluruhan sambil tetap mempertahankan solusi pemulihan bencana yang kuat dan andal yang memenuhi persyaratan bisnis Anda. Anda dapat menggunakan Elastic Disaster Recovery untuk fokus pada aktivitas bisnis inti Anda, sambil AWS mengurus infrastruktur yang mendasarinya untuk solusi pemulihan bencana Anda.

Untuk SQL Server, Anda dapat menggunakan Elastic Disaster Recovery sebagai opsi pemulihan bencana yang hemat biaya. Lisensi untuk node pasif dalam arsitektur SQL Server yang toleran kesalahan dan sangat tersedia tercakup jika Anda menggunakan Jaminan Perangkat Lunak aktif. Namun, Anda masih membayar biaya komputasi untuk server pasif untuk online. Dengan Elastic Disaster Recovery, server utama dapat mereplikasi ke lingkungan DR tanpa perlu mempertahankan Jaminan Perangkat Lunak aktif dan tanpa harus membayar biaya komputasi pemulihan bencana. Kombinasi penghematan ini dapat mengurangi biaya pemulihan bencana SQL Server Anda hingga 50 persen atau lebih.

Diagram berikut menunjukkan contoh arsitektur untuk solusi berdasarkan Elastic Disaster Recovery.



Untuk informasi selengkapnya, lihat [Cara mengatur ketersediaan tinggi untuk SQL Server di situs DR yang dipulihkan menggunakan AWS Elastic Disaster Recovery] Beban Kerja Microsoft di AWS Blog.

Perbandingan biaya

Tabel berikut membandingkan biaya HA/DR solusi yang tercakup dalam bagian ini. Asumsi berikut dibuat untuk tujuan perbandingan ini:

- Jenis contoh - r5d.xlarge
- Jenis lisensi - Lisensi disertakan untuk Windows dan SQL Server

- Wilayah — `us-east-1`

Solusi	Ketersediaan tinggi	Pemulihan bencana	Edisi perusahaan	Edisi standar	Biaya
Pengiriman log	Tidak	Ya	Ya	Ya	Edisi SQL Server Enterprise: $32.674.8 (2 node) SQL Server edisi Standar: $14.804.4 (2 node)
Selalu Aktif pada grup ketersediaan	Ya	Ya	Ya	Ya, tetapi grup ketersediaan dasar (2 node)	Edisi SQL Server Enterprise: $32.674.8 (2 node) SQL Server edisi Standar: $14.804.4 (2 node)
Selalu Aktif FCIs	Ya	Tidak	Ya	Ya (2 node)	SQL Server edisi Standar: $14.804.4
Grup ketersediaan terdistribusi	Ya	Ya	Ya	Tidak	Edisi SQL Server Enterprise: $65.349.6 (4 node)

Solusi	Ketersediaan tinggi	Pemulihan bencana	Edisi perusahaan	Edisi standar	Biaya
Elastic Disaster Recovery	Tidak	Ya	Ya	Ya	Sekitar $107,48/bulan untuk replikasi 1 instans dan 1 TB penyimpanan Catatan: Elastic Disaster Recovery ditagih setiap jam, per server replikasi. Biayanya sama, terlepas dari jumlah disk, ukuran penyimpanan, jumlah peluncuran bor atau pemulihan, atau Wilayah yang Anda replikasi.

Solusi	Ketersediaan tinggi	Pemulihan bencana	Edisi perusahaan	Edisi standar	Biaya
Penjaga Data SIOS	Ya	Ya	Ya	Ya	Grup ketersediaan Selalu Aktif dengan Jaminan Perangkat Lunak (2 node, 24 core): $213.480 Cluster SQL Server 2-node berjalan pada edisi Standar SQL Server dengan SIOS DataKeeper dan Jaminan Perangkat Lunak: $61.530 (2 node)
AWS DMS	Tidak	Ya	Ya	Ya	$745.38/bulan untuk instans r5.xlarge dan penyimpanan 1 TB

Rekomendasi optimisasi biaya

Kami menyarankan Anda mengambil langkah-langkah berikut untuk memilih HA/DR solusi yang memenuhi persyaratan organisasi Anda:

- Tinjau bagian Pilih instans EC2 yang tepat untuk beban kerja SQL Server dari panduan ini.
- Tentukan IOPS dan persyaratan throughput beban kerja Anda dengan menjalankan penghitung kinerja selama beban kerja puncak:
  - IOPS = disk reads/sec +disk menulis/detik
  - Throughput = disk read bytes/sec +disk write bytes/detik
- Gunakan jenis volume penyimpanan berikut untuk kinerja dan penghematan biaya yang lebih baik:
  - NVMe penyimpanan instance untuk `tempdb` dan ekstensi kolam penyangga
  - volume io2 untuk file database
- Gunakan AWS Trusted Advisoruntuk rekomendasi tentang pengoptimalan biaya untuk SQL Server di Amazon EC2. Anda tidak perlu menginstal agen Trusted Advisor untuk melakukan pemeriksaan pengoptimalan SQL Server. Trusted Advisor memeriksa konfigurasi instans yang disertakan dengan lisensi Amazon EC2 SQL Server, seperti CPUs virtual (CPUsv), versi, dan edisi. Kemudian, Trusted Advisor buat rekomendasi berdasarkan praktik terbaik.
- Gunakan AWS Compute Optimizer untuk instans Amazon EC2 dan rekomendasi ukuran tepat Amazon EBS.
- Gunakan AWS Kalkulator Hargauntuk merancang HA/DR strategi Anda untuk estimasi biaya.
- Untuk menentukan apakah menurunkan versi dari edisi SQL Server Enterprise ke edisi Standar SQL Server adalah opsi yang memungkinkan, gunakan tampilan manajemen dinamis sys dm_db_persisted_sku_features untuk mengidentifikasi fitur khusus edisi yang aktif dalam database saat ini.

  > **Note**
  >
  > Side-by-side migrasi diperlukan untuk perubahan edisi SQL Server saat menggunakan instans EC2 yang disertakan lisensi.

- Lakukan latihan pemulihan bencana setengah tahunan atau tahunan untuk merancang desain yang lebih baik yang dapat memulihkan database dengan RTO dan RPO yang ditentukan. Ini juga dapat membantu Anda mengidentifikasi kelemahan arsitektur apa pun.

Sumber daya tambahan

- Sederhanakan penerapan ketersediaan tinggi Microsoft SQL Server Anda menggunakan FSx Amazon untuk Windows File Server AWS (Blog Penyimpanan)
- Catatan Lapangan: Membangun Arsitektur Multi-Wilayah untuk SQL Server menggunakan FCI dan Grup Ketersediaan Terdistribusi (AWS Blog Arsitektur)
- Arsitek pemulihan bencana untuk SQL Server pada AWS: Bagian 1 (Blog AWS Database)
- Microsoft SQL ketersediaan tinggi dengan Amazon FSx untuk Windows () YouTube
- Memaksimalkan Kinerja Microsoft SQL Server dengan Amazon EBS (Blog Penyimpanan)AWS
- Membandingkan pola penyimpanan lokal Anda dengan layanan AWS Penyimpanan (Blog AWS Penyimpanan)
- Berencana untuk mengganti pusat data NAS dengan Amazon FSx File Gateway (Blog AWS Penyimpanan)
- Mengoptimalkan biaya untuk penerapan SQL Server ketersediaan tinggi Anda di AWS (Blog Penyimpanan)AWS
- Cara mengatur pemulihan bencana untuk SQL Server Always On Availability Groups menggunakan AWS Elastic Disaster Recovery(Microsoft Workloads on) AWS
- Cara mengatur ketersediaan tinggi untuk SQL Server di situs DR yang dipulihkan menggunakan AWS Elastic Disaster Recovery (Microsoft Workloads on) AWS

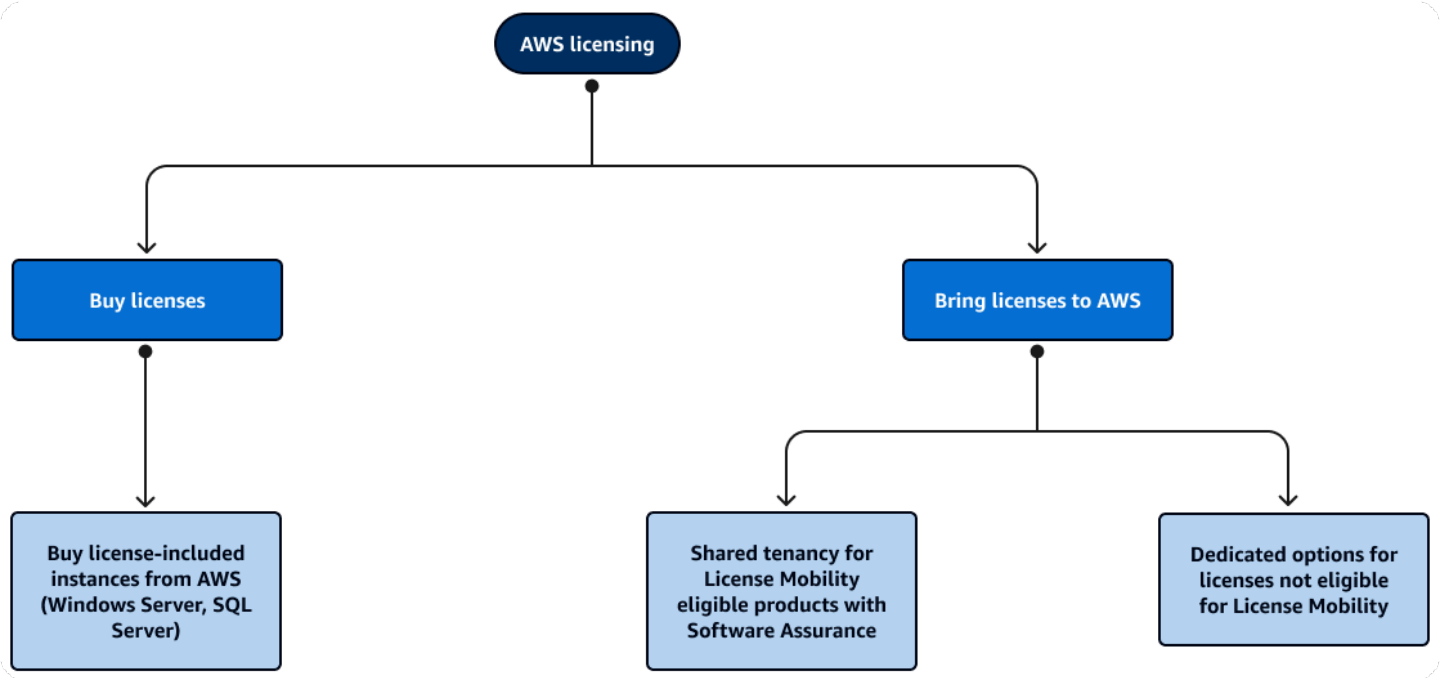
Memahami lisensi SQL Server

Ikhtisar

Karena semakin banyak bisnis memindahkan beban kerja mereka ke cloud, mengoptimalkan biaya pada platform cloud telah menjadi prioritas utama. Lisensi adalah salah satu biaya paling signifikan yang terkait dengan menjalankan beban kerja Microsoft. AWS Bagian ini menjelaskan cara mengoptimalkan biaya AWS dengan mengoptimalkan lisensi Microsoft untuk SQL Server.

AWS opsi lisensi

AWS menawarkan berbagai pilihan optimasi biaya yang fleksibel untuk lisensi. Opsi lisensi ini dirancang untuk membantu Anda mengurangi biaya, menjaga kepatuhan, dan memenuhi kebutuhan bisnis Anda.



AWS mengkategorikan lisensi menjadi tiga jenis utama:

1. Termasuk lisensi - Opsi lisensi ini memungkinkan Anda untuk membeli dan menggunakan lisensi sesuai permintaan, hanya membayar untuk apa yang Anda gunakan. Opsi yang disertakan lisensi sangat ideal untuk skenario di mana Anda memerlukan fleksibilitas dalam penggunaan lisensi Anda dan ingin menghindari biaya di muka. Anda dapat memilih dari berbagai Windows Server, SQL Server, dan produk Microsoft lainnya.
2. Bawa produk Lisensi Anda Sendiri (BYOL) dengan mobilitas lisensi - Opsi lisensi ini dirancang untuk skenario di mana Anda sudah memiliki lisensi yang ada dan ingin menggunakannya di cloud. AWS memungkinkan pelanggan untuk membawa lisensi mereka sendiri ke cloud melalui program Mobilitas Lisensi Microsoft. Anda dapat membawa produk yang memiliki Mobilitas Lisensi, seperti SQL Server dengan Jaminan Perangkat Lunak (SA), ke penyewaan bersama atau khusus untuk mengurangi biaya instans Anda AWS .
3. Produk BYOL tanpa mobilitas lisensi — Untuk produk Microsoft yang tidak memiliki Mobilitas Lisensi, seperti Windows Server, AWS menawarkan opsi khusus untuk menggunakan produk ini di cloud. Selain itu, host khusus menawarkan kesempatan untuk melisensikan di tingkat inti fisik. Ini dapat menghemat 50 persen atau lebih pada lisensi yang diperlukan untuk menjalankan beban kerja Anda. Host khusus adalah pilihan yang bagus untuk beban kerja yang stabil dan dapat diprediksi berjalan sebagian besar waktu.

Dampak biaya membawa lisensi

Membawa lisensi dapat berdampak signifikan pada biaya menjalankan beban kerja Microsoft. AWS Jika Anda membawa lisensi sendiri, Anda tidak perlu membayar biaya lisensi tambahan untuk instans yang berjalan di cloud. Hal ini dapat menyebabkan penghematan biaya yang signifikan.

Perbandingan berikut menunjukkan biaya bulanan sesuai permintaan untuk menjalankan satu instance c5.xlarge 24/7:

- Windows Server+SQL Server Enterprise edisi: $1353/bulan (Lisensi termasuk)
- Windows Server+SQL Server edisi Standar: $609/bulan (Lisensi termasuk)
- Hanya Windows Server: $259/bulan (Termasuk lisensi)
- Hanya komputasi (Linux): $127/bulan

Pada akhirnya, membawa lisensi Anda sendiri dapat berdampak signifikan pada biaya menjalankan beban kerja Microsoft. AWS Jika Anda menggunakan lisensi yang ada, Anda dapat mengurangi biaya lisensi dan menghemat uang untuk keseluruhan AWS tagihan Anda.

Optimalisasi lisensi

AWS Optimalisasi dan Penilaian Lisensi (AWS OLA) dapat membantu Anda mengoptimalkan lisensi Anda dengan mengurangi biaya komputasi dan lisensi. AWS OLA dirancang untuk mengevaluasi persyaratan lisensi Anda untuk beban kerja yang berjalan pada AWS atau untuk beban kerja yang direncanakan untuk migrasi. AWS OLA memberikan rekomendasi untuk mengoptimalkan penggunaan lisensi.

Salah satu strategi utama untuk mengoptimalkan penggunaan lisensi adalah contoh ukuran yang tepat. Ukuran yang tepat melibatkan pemilihan jenis instans yang tepat untuk beban kerja Anda berdasarkan persyaratan CPU, memori, dan penyimpanannya. Dengan memilih ukuran instans yang sesuai, Anda dapat memastikan bahwa Anda menggunakan sumber daya dengan cara yang hemat biaya. Hal ini dapat menyebabkan penghematan biaya yang signifikan.

Dengan lisensi perangkat lunak Microsoft, jumlah inti yang dijalankan perangkat lunak merupakan faktor penting dalam menentukan biaya lisensi. Misalnya, lisensi Windows Server dan SQL Server biasanya dilisensikan pada jumlah core. Dengan instans ukuran yang tepat, Anda dapat menurunkan jumlah inti yang dijalankan perangkat lunak Microsoft dan, pada gilirannya, mengurangi biaya instance dan jumlah lisensi yang diperlukan.

Rekomendasi optimisasi biaya

Mengoptimalkan lisensi adalah komponen kunci dari pengoptimalan biaya. AWS Dengan menerapkan strategi yang tepat, Anda dapat mengurangi biaya lisensi, mempertahankan kepatuhan, dan mencapai nilai terbaik dari investasi lisensi Anda. Bagian ini menguraikan beberapa strategi untuk optimasi lisensi.

Bawa lisensi Windows Server Anda yang memenuhi syarat

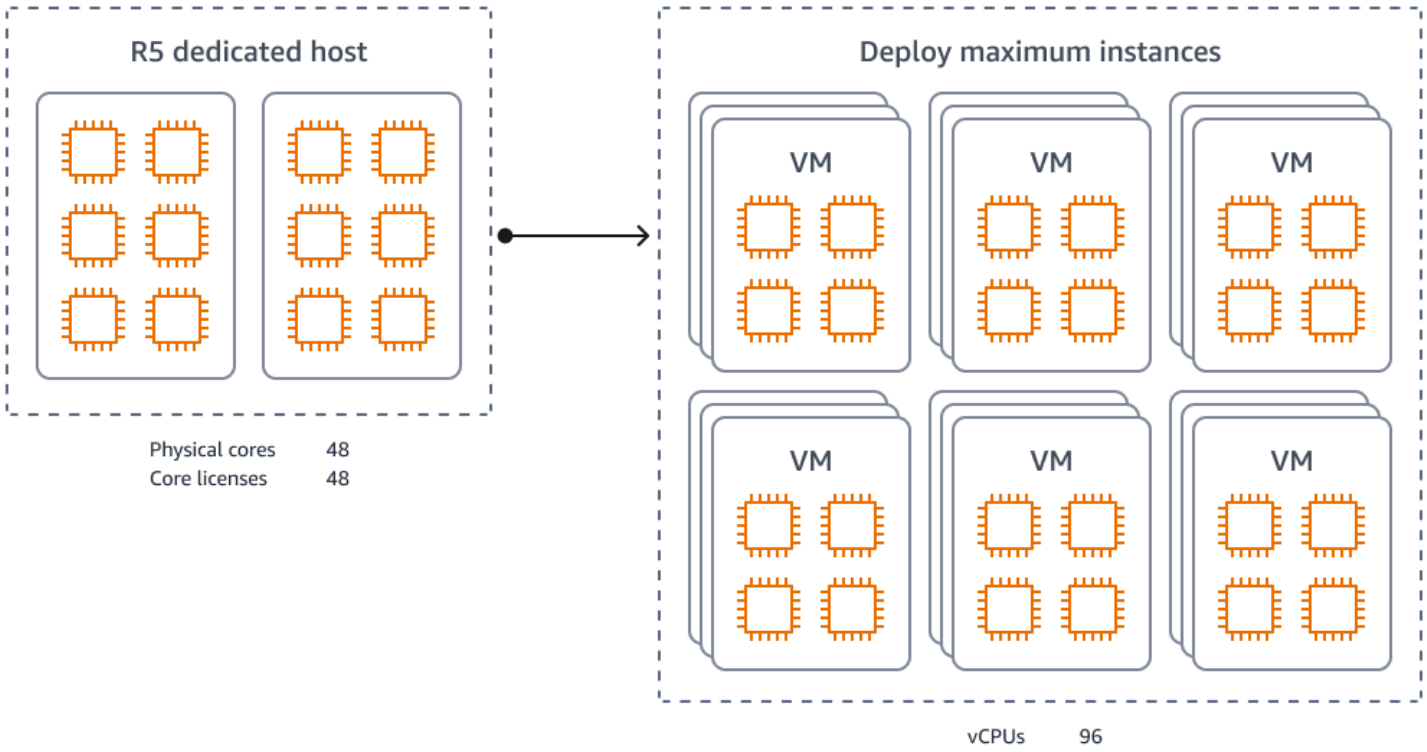
Membawa lisensi Windows Server Anda sendiri adalah salah satu strategi paling efektif untuk optimasi lisensi. Strategi ini memungkinkan Anda memanfaatkan investasi yang ada untuk mengurangi AWS pengeluaran Anda.

Misalnya, Anda dapat menerapkan Windows Server 2019 dan versi sebelumnya di Host Khusus Amazon EC2 jika Anda membeli lisensi sebelum 1/10/2019 atau membeli lisensi sebagai true-up berdasarkan Perjanjian Perusahaan aktif yang ditandatangani sebelum tanggal tersebut. Aturan ini didasarkan pada perubahan yang dibuat Microsoft pada tahun 2019 terhadap syarat dan ketentuan lisensi untuk produk tanpa Mobilitas Lisensi, seperti Windows Server, ketika digunakan di Penyedia Terdaftar (misalnya, Alibaba AWS, atau Google Cloud). Di bawah persyaratan baru, Anda tidak dapat membawa lisensi Windows Server Anda sendiri AWS tetapi harus menggunakan instance yang disertakan lisensi. Namun, jika Anda membeli lisensi abadi sebelum tanggal tersebut, maka Anda masih dapat menerapkan lisensi Windows Server tersebut di Host Khusus Amazon EC2.

Lisensi tingkat fisik

Lisensi pada tingkat inti fisik memungkinkan Anda untuk melisensikan hanya inti fisik host, sehingga Anda kemudian dapat menerapkan jumlah maksimum instance tanpa memengaruhi jumlah lisensi yang diperlukan. Ini biasanya dilakukan dengan menggunakan Windows Server Datacenter dan edisi SQL Server Enterprise.

Sebagai contoh, pertimbangkan host khusus R5 dengan 48 core, yang diterjemahkan menjadi 96 v. CPUs Jika Anda menggunakan edisi Windows Server Datacenter, Anda hanya perlu 48 lisensi. Ini memungkinkan Anda untuk menerapkan kombinasi instance hingga 96 vCPUs, seperti yang ditunjukkan diagram berikut.



Pendekatan ini bisa sangat hemat biaya jika Anda memiliki beban kerja yang cukup untuk memaksimalkan jumlah instance yang dapat Anda jalankan di host. Dengan melisensikan pada tingkat inti fisik, Anda dapat menghindari biaya lisensi tambahan untuk setiap contoh dan mencapai nilai terbaik untuk investasi lisensi Anda.

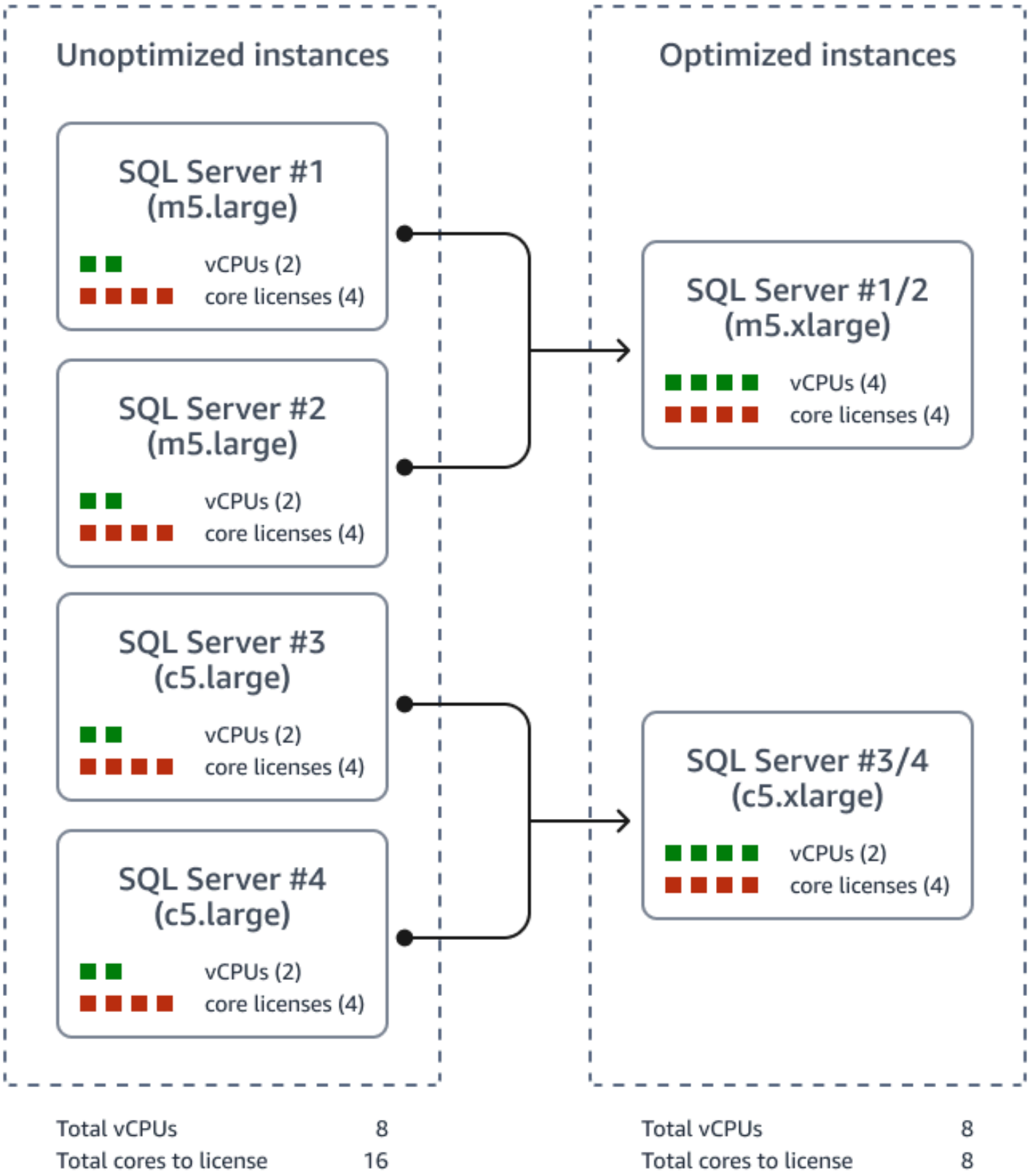
Lisensi pada tingkat inti fisik SQL Server

Dalam penyewaan bersama, lisensi SQL Server didasarkan pada jumlah v yang CPUs dialokasikan ke instance. Sebaliknya, dengan host khusus, Anda dapat melisensikan edisi SQL Server Enterprise di tingkat inti fisik atau di tingkat vCPU.

Seperti contoh sebelumnya dari host khusus R5, jika Anda melisensikan edisi SQL Server Enterprise pada tingkat inti fisik, maka Anda hanya perlu 48 lisensi edisi SQL Server Enterprise untuk melisensikan host. Sebaliknya, dalam penyewaan bersama, di mana satu-satunya pilihan adalah melisensikan oleh vCPU, Anda harus memiliki 96 lisensi edisi SQL Server Enterprise untuk beban kerja yang sama. Oleh karena itu, dedicated host dapat menghemat hingga 50 persen pada biaya lisensi SQL Server dibandingkan dengan penyewaan bersama. Ini selain menghemat biaya instans dengan membawa lisensi Windows yang memenuhi syarat.

Mengkonsolidasikan instance SQL Server

Konsolidasi SQL Server adalah proses menggabungkan beberapa instance SQL Server ke satu server. SQL Server memerlukan minimal empat lisensi inti per instance, bahkan jika instance hanya memiliki dua v. CPUs Ini berarti bahwa menjalankan SQL Server di server dengan kurang dari empat inti dapat menyebabkan Anda melisensikan instance ini secara berlebihan dan menggunakan lebih banyak lisensi daripada yang diperlukan.


Unoptimized instances
SQL Server #1 (m5.large)
vCPUs (2)
core licenses (4)
SQL Server #2 (m5.large)
vCPUs (2)
core licenses (4)
SQL Server #3 (c5.large)
vCPUs (2)
core licenses (4)
SQL Server #4 (c5.large)
vCPUs (2)
core licenses (4)
Optimized instances
SQL Server #1/2 (m5.xlarge)
vCPUs (4)
core licenses (4)
SQL Server #3/4 (c5.xlarge)
vCPUs (2)
core licenses (4)
Total vCPUs 8
Total cores to license 16
Total vCPUs 8
Total cores to license 8

Misalnya, mengkonsolidasikan dua instance dengan CPUs masing-masing dua v menjadi satu instance dengan empat v CPUs dapat mengurangi persyaratan lisensi sebesar 50 persen. Ini karena hanya empat lisensi inti yang diperlukan, bukan delapan.

Untuk informasi selengkapnya tentang konsolidasi, lihat bagian konsolidasi SQL Server dari panduan ini.

Turunkan versi SQL Server edisi

Mengubah edisi SQL Server dapat menjadi strategi utama untuk mengoptimalkan penggunaan lisensi dan mengurangi biaya. Edisi Enterprise SQL Server jauh lebih mahal daripada edisi Standar, sehingga penurunan versi dapat menghasilkan penghematan biaya yang signifikan.

Enkripsi Data Transparan (TDE) dan grup ketersediaan Selalu Aktif adalah dua fitur populer dalam edisi SQL Server Enterprise. Namun, ada alternatif hemat biaya untuk fitur-fitur ini yang dapat Anda pertimbangkan jika Anda tidak memerlukan set fitur lengkap edisi SQL Server Enterprise. Misalnya, Anda bisa mendapatkan TDE dalam edisi Standar SQL Server dimulai dengan SQL Server 2019. Sebagai pengganti grup ketersediaan Selalu Aktif, Anda dapat menggunakan pengelompokan failover dengan penyimpanan bersama FSx untuk Windows File Server untuk ketersediaan tinggi dengan edisi Standar SQL Server.

Dengan menurunkan versi dari edisi SQL Server Enterprise ke edisi Standar SQL Server, Anda dapat secara signifikan mengurangi biaya lisensi Anda. Untuk informasi selengkapnya, lihat Mengoptimalkan biaya untuk penerapan SQL Server ketersediaan tinggi Anda pada AWS posting di Blog Penyimpanan. AWS

Selain mengurangi biaya lisensi, menurunkan versi edisi SQL Server dapat membantu mengurangi pengeluaran Jaminan Perangkat Lunak Anda dan membantu Anda menghindari true-up di masa depan. Jika Anda mengembalikan lisensi yang tidak digunakan ke rak, Anda dapat menghindari biaya lisensi tambahan dan mencapai nilai terbaik dari investasi lisensi Anda.

Penting untuk mengevaluasi beban kerja SQL Server Anda dengan hati-hati dan menentukan fitur mana yang penting untuk kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat Menilai lingkungan Anda di Panduan AWS Preskriptif, dan tentukan apakah database Microsoft SQL Server Anda menggunakan fitur khusus edisi SQL Server Enterprise.

Jika Anda memilih edisi SQL Server yang tepat dan menggunakan alternatif untuk fitur edisi SQL Server Enterprise, Anda dapat mencapai penghematan biaya yang signifikan sambil mempertahankan kepatuhan dan memenuhi kebutuhan bisnis Anda. Untuk informasi selengkapnya tentang opsi penurunan versi, lihat bagian Bandingkan edisi SQL Server dari panduan ini.

Gunakan edisi SQL Server Developer di lingkungan non-produksi

Di lingkungan non-produksi, Anda dapat menerapkan edisi SQL Server yang dapat dilisensikan, seperti edisi Enterprise atau Standar, dengan menggunakan langganan MSDN di lingkungan lokal. Namun, langganan MSDN tidak memiliki Mobilitas Lisensi. Jadi, jika Anda bermigrasi ke AWS, Anda tidak dapat membawa lisensi tersebut. Anda harus menggunakan edisi SQL Server Developer sebagai gantinya.

SQL Server Developer edition adalah edisi SQL Server berfitur lengkap yang tersedia secara gratis. Edisi ini tersedia untuk SQL Server versi 2016 dan yang lebih baru. Anda dapat mengunduhnya dari situs web Microsoft. Edisi SQL Server Developer dimaksudkan untuk digunakan di semua lingkungan non-produksi, seperti pengembangan, pengujian, dan pementasan, selama tidak terhubung ke data produksi langsung.

Jika Anda menggunakan edisi SQL Server Developer di lingkungan non-produksi, Anda dapat menghindari biaya lisensi tambahan. Untuk informasi selengkapnya, lihat bagian Evaluasi SQL Server Developer edition dari panduan ini.

Optimalkan CPU untuk beban kerja SQL Server

Dalam beberapa kasus, Anda mungkin diminta untuk memilih jenis instans dengan CPUs lebih dari yang diperlukan untuk beban kerja Anda karena faktor lain seperti RAM atau batas jaringan. Namun, AWS berikan solusi untuk membantu Anda mengoptimalkan biaya lisensi Anda dalam situasi ini.

Anda dapat, seperti kebanyakan pelanggan yang membawa lisensi inti SQL Server, menonaktifkan hyperthreading atau mematikan CPU pada instans EC2 untuk membatasi jumlah yang tersedia untuk host. CPUs Opsi ini memungkinkan Anda untuk memanfaatkan kemampuan instans lainnya, seperti RAM, sambil tetap menghemat biaya pembelian lisensi tambahan.

Misalnya, jika Anda menerapkan instance r5.4xlarge karena beban kerja Anda membutuhkan memori 128 GB tetapi Anda hanya membutuhkan delapan inti SQL Server, maka Anda dapat menonaktifkan hyperthreading instance hanya dengan delapan aktif. CPUs Dengan melakukan ini, Anda dapat menghemat 50 persen pada lisensi SQL Server yang diperlukan, karena Anda hanya perlu melisensikan delapan core yang sedang digunakan secara aktif.

Tipe instans	Jumlah v CPUs	VCPU aktif dengan fitur Optimalkan CPUs	Penghematan lisensi SQL Server
r5.4xlarge	16	8	50%

Tipe instans	Jumlah v CPUs	VCPU aktif dengan fitur Optimalkan CPUs	Penghematan lisensi SQL Server
r5.12xlarge	48	8	83%

Fitur Optimalkan CPU dapat dikonfigurasi selama konfigurasi peluncuran Amazon EC2 atau dengan memodifikasi instans yang ada. Ini juga dapat diterapkan pada instans BYOL dan Amazon EC2 yang disertakan lisensi. Fleksibilitas ini membantu Anda mengukur CPU sesuai kebutuhan beban kerja Anda, sekaligus mengurangi Windows Server dan melisensikan. SQL Server Untuk instans Amazon EC2 yang disertakan dengan lisensi, CPUs pengurangan memberikan penghematan instan pada biaya lisensi.

Jika Anda mengukur instans dengan benar, Anda dapat memastikan bahwa Anda menggunakan jenis instans yang paling hemat biaya untuk beban kerja Anda. Saat AWS memperkenalkan jenis instans baru, penting untuk mengevaluasi apakah instans baru ini dapat memenuhi persyaratan beban kerja dengan inti yang lebih sedikit.

Sumber daya tambahan

- Amazon Web Services dan Microsoft: Pertanyaan yang Sering Diajukan (AWS dokumentasi)

Pilih instans EC2 yang tepat untuk beban kerja SQL Server

> **Important**
>
> Sebelum Anda membaca bagian ini, kami sarankan Anda membaca terlebih dahulu Memahami lisensi SQL Server dan Pilih jenis instans yang tepat untuk beban kerja Windows bagian panduan ini.

Ikhtisar

Microsoft SQL Server telah berjalan di instans Amazon Elastic Compute Cloud (Amazon EC2) selama lebih dari 15 tahun. AWS telah mengambil pengalaman itu dan menggunakannya untuk membantu mengembangkan instans Amazon EC2 agar sesuai dengan beban kerja SQL Server yang berjalan dari spesifikasi minimal hingga cluster Multi-wilayah berkinerja tinggi.

Memilih instans EC2 yang benar untuk SQL Server sebagian besar tergantung pada beban kerja Anda. Memahami bagaimana SQL Server dilisensikan, cara menggunakan memori, dan bagaimana fitur SQL Server selaras dengan penawaran Amazon EC2 dapat membantu memandu Anda ke instans EC2 terbaik untuk aplikasi Anda.

Bagian ini membahas berbagai beban kerja SQL Server dan bagaimana mereka dapat dipasangkan dengan instans EC2 tertentu untuk menjaga lisensi dan biaya komputasi Anda seminimal mungkin.

Perbandingan biaya

Amazon EC2 memungkinkan Anda untuk Membawa Lisensi Anda Sendiri (BYOL) atau membayar saat Anda pergi dengan lisensi Windows Server dan SQL Server. Untuk pay-as-you-go lisensi, biaya lisensi untuk lisensi Windows Server dan SQL Server dimasukkan ke dalam biaya per jam dari instans EC2. Misalnya, Anda dapat memiliki yang berbeda AMIs dengan harga yang berbeda. Harga AMI bergantung pada edisi SQL Server yang dijalankan AMI.

Harga Windows Server dan SQL Server tidak diperinci. Anda tidak akan menemukan harga terperinci pada alat seperti. AWS Kalkulator Harga Jika Anda memilih kombinasi yang berbeda dari penawaran yang termasuk lisensi, biaya lisensi dapat disimpulkan, seperti yang ditunjukkan tabel berikut.

Instans EC2	AMI	Harga komputasi	Harga lisensi Windows	Harga lisensi SQL	Harga total
r5.xlarge	Linux (harga komputasi)	$183,96	-	-	$183,96
r5.xlarge	Linux + Pengembang SQL	$183,96	$0	$0	$183,96
r5.xlarge	Server Windows (LI)	$183,96	$134,32	-	$318.28
r5.xlarge	Pengembang Windows+SQL	$183,96	$134,32	$0	$318.28
r5.xlarge	Windows+SQL Web (LI)	$183,96	$134,32	$49,64	$367,92

Instans EC2	AMI	Harga komputasi	Harga lisensi Windows	Harga lisensi SQL	Harga total
r5.xlarge	Windows+SQL Standar (LI)	$183,96	$134,32	$350.4	$668,68
r5.xlarge	Windows +SQL Perusahaan (LI)	$183,96	$134,32	$1095	$1413,28

> **Note**
>
> Harga di tabel sebelumnya didasarkan pada harga sesuai permintaan di Wilayah. `us-east-1`

Metode yang paling hemat biaya untuk menjalankan SQL Server adalah tetap pada edisi tingkat yang lebih rendah sampai Anda memerlukan fitur dari edisi tingkat yang lebih tinggi. Untuk informasi selengkapnya, lihat bagian Bandingkan edisi SQL Server dari panduan ini. Upgrade dari SQL Server Web edition ke SQL Server Standard edition lebih dari tujuh kali biaya lisensi SQL Server dan lebih dari tiga kali biaya pindah dari edisi Standar ke edisi Enterprise. Perbedaan dalam biaya perizinan merupakan faktor utama yang perlu dipertimbangkan dan dieksplorasi di bagian ini.

Skenario pengoptimalan biaya

Pertimbangkan contoh skenario di mana perusahaan analitik yang melacak kendaraan pengiriman berusaha meningkatkan kinerja SQL Server-nya. Setelah pakar MACO meninjau kemacetan kinerja perusahaan, perusahaan beralih dari instance x1e.2xlarge ke instance x2iedn.xlarge. Meskipun ukuran instans lebih kecil, penyempurnaan pada instans x2 meningkatkan kinerja dan pengoptimalan SQL Server dengan menggunakan ekstensi kumpulan buffer. Ini memungkinkan perusahaan untuk menurunkan versi dari edisi SQL Server Enterprise ke edisi Standar SQL Server dan mengurangi lisensi SQL Server dari 8 v menjadi 4 v. CPUs CPUs

Sebelum optimasi:

Server	Instans EC2	Edisi SQL Server	Biaya bulanan
Prod DB1	x1e.2xlarge	Perusahaan	$3.918,64
Prod DB2	x1e.2xlarge	Perusahaan	$3.918,64
Jumlah			$7.837,28

Setelah optimasi:

Server	Instans EC2	Edisi SQL Server	Biaya bulanan
Prod DB1	x2iedn.xlarge	Standar	$1,215.00
Prod DB2	x2iedn.xlarge	Standar	$1,215.00
Jumlah			$2,430,00

Perubahan gabungan dari instans x1e.2xlarge ke instance x2iedn.xlarge memungkinkan pelanggan contoh menghemat $5.407 per bulan di server basis data produksi mereka. Ini mengurangi total biaya beban kerja sebesar 69 persen.

> **Note**
>
> Harga di tabel sebelumnya didasarkan pada harga sesuai permintaan di Wilayah. `us-east-1`

Rekomendasi optimisasi biaya

Instans memori yang dioptimalkan

Salah satu aspek terpenting dari SQL Server adalah memahami ketergantungannya pada memori. SQL Server mencoba menggunakan semua RAM yang tersedia yang tidak digunakan oleh sistem operasi (hingga 2 TB untuk instalasi default). Ini dilakukan karena alasan kinerja. Bekerja dengan data dalam memori jauh lebih berkinerja daripada harus terus-menerus menarik data dari disk,

membuat perubahan, dan kemudian menulisnya kembali ke disk. Sebaliknya, SQL Server mencoba memuat sebanyak mungkin data dari database terlampir dan menyimpan data tersebut dalam RAM. Perubahan yang dilakukan pada data terjadi di memori dan dikeraskan ke disk di lain waktu.

> **Note**
>
> Untuk penjelasan rinci tentang bagaimana SQL Server menulis perubahan, lihat Menulis Halaman dalam dokumentasi Microsoft.

Karena SQL Server berkinerja lebih baik dengan jumlah RAM yang lebih besar, kami biasanya menyarankan untuk memulai dengan jenis instans yang dioptimalkan untuk memori Amazon EC2. Instans yang dioptimalkan memori serbaguna dan menawarkan berbagai opsi berbeda. Keluarga R memiliki vCPU-to-RAM rasio 1-ke-8 dan memiliki opsi untuk prosesor Intel, prosesor AMD, jaringan yang disempurnakan, kinerja EBS yang ditingkatkan, penyimpanan instans, dan kecepatan prosesor yang ditingkatkan. Untuk beban kerja yang berat memori, ada juga keluarga X yang menggabungkan banyak opsi yang sama dan memperluas rasio menjadi 1 banding 32. vCPU-to-RAM Karena keserbagunaan instance yang dioptimalkan memori, Anda dapat menerapkannya ke beban kerja SQL Server dari semua bentuk dan ukuran.

Beban kerja di bawah sumber daya minimal (kurang dari 4 vCPUs)

Meskipun beberapa kasus penggunaan bekerja dengan baik dengan instans burstable (T3), sebaiknya Anda menghindari penggunaan instans burstable untuk beban kerja SQL Server. Lisensi untuk SQL Server didasarkan pada jumlah v yang CPUs ditugaskan ke sebuah instance. Jika SQL Server menganggur hampir sepanjang hari dan memperoleh kredit burst, Anda membayar lisensi SQL yang tidak sepenuhnya Anda gunakan. Selain itu, SQL Server memiliki persyaratan lisensi minimum 4 core per server. Ini berarti jika Anda memiliki beban kerja SQL Server yang tidak memerlukan daya komputasi CPUs senilai 4 v, Anda membayar lisensi SQL Server yang tidak Anda gunakan. Dalam skenario ini, akan lebih baik untuk mengkonsolidasikan beberapa instance SQL Server ke server yang lebih besar.

Beban kerja menggunakan sumber daya minimal (kurang dari 64 GB RAM)

Banyak beban kerja SQL Server di bawah 64 GB RAM tidak memprioritaskan kinerja tinggi atau ketersediaan tinggi. Untuk jenis beban kerja ini, edisi Web SQL Server mungkin cocok jika aplikasi tercakup dalam pembatasan lisensi Microsoft.

> **Important**
>
> SQL Server Web edition memiliki kasus penggunaan terbatas berdasarkan persyaratan lisensi Microsoft. SQL Server Web edition hanya dapat digunakan untuk mendukung halaman web publik dan internet yang dapat diakses, situs web, aplikasi web, dan layanan web. Ini mungkin tidak digunakan untuk mendukung line-of-business aplikasi (misalnya, manajemen hubungan pelanggan, manajemen sumber daya perusahaan, dan aplikasi serupa lainnya).

SQL Server Web edition skala hingga 32 v CPUs dan 64 GB RAM dan 86 persen lebih murah daripada SQL Server Standard edition. Untuk beban kerja sumber daya yang rendah, menggunakan instance yang dioptimalkan memori AMD seperti r6a, yang memiliki harga komputasi 10 persen lebih murah daripada rekan Intel, juga merupakan cara yang baik untuk menjaga biaya lisensi komputasi dan SQL seminimal mungkin.

Beban kerja dengan sumber daya rata-rata (kurang dari 128 GB RAM)

SQL Server Standard edition digunakan pada sebagian besar beban kerja SQL Server hingga 128 GB RAM. SQL Server Standard edition adalah 65-75 persen lebih murah daripada edisi SQL Server Enterprise dan dapat meningkatkan skala hingga 48 v CPUs dan 128 GB RAM. Karena batasan RAM 128 GB biasanya tercapai sebelum batasan 48 vCPU, ini adalah fokus sebagian besar pelanggan yang ingin menghindari peningkatan ke edisi SQL Server Enterprise.

SQL Server memiliki fitur yang disebut ekstensi buffer pool. Fitur ini memungkinkan SQL Server untuk menggunakan sebagian disk untuk bertindak sebagai perpanjangan RAM. Ekstensi buffer pool berfungsi dengan baik bila dikombinasikan dengan penyimpanan ultra-cepat, seperti yang NVMe SSDs digunakan dalam penyimpanan instans Amazon EC2. Instans Amazon EC2 yang berisi penyimpanan instans dilambangkan dengan “d” dalam nama instance (misalnya, r5d, r6id, dan x2iedn).

Ekstensi buffer pool bukan pengganti RAM normal. Namun, jika Anda memerlukan lebih dari 128 GB RAM, Anda dapat menggunakan ekstensi kumpulan buffer dengan instans EC2 seperti r6id.4xlarge dan x2iedn.xlarge untuk menunda peningkatan ke lisensi edisi Enterprise.

Beban kerja kinerja tinggi (RAM lebih dari 128 GB)

Beban kerja SQL Server yang membutuhkan kinerja tinggi menantang untuk pengoptimalan biaya karena ketergantungan mereka pada banyak sumber daya. Namun, memahami perbedaan dalam contoh EC2 dapat mencegah Anda membuat pilihan yang salah.

Tabel berikut menunjukkan berbagai instans EC2 yang dioptimalkan memori dan batas kinerjanya.

	r5b	r6idn	r7iz	x2iedn	x2iezn
Prosesor	3.1 GHz Prosesor Intel Xeon Generasi ke-2	3.5 GHz Prosesor Intel Xeon Generasi ke-3	3.9 GHz Prosesor Intel Xeon Scalable Generasi ke-4	3.5 GHz Prosesor Intel Xeon Generasi ke-3	4.5 GHz Prosesor Intel Xeon Generasi ke-2
Rasio CPU: RAM	1:8	1:8	1:8	1:32	1:32
Maks vCPU	96	128	128	128	48
RAM Maks	768 GB	1.024 GB	1.024 GB	4.096 GB	1.536 GB
Penyimpanan instans	–	NVMe SSD (4x 1900 GB)	–	NVMe SSD (2x 1900 GB)	–
io2 Blok Ekspres	Didukung	Didukung	Didukung	Didukung	–
Max EBS IOPS	260.000	350.000	160.000	260.000	80.000
Throughput EBS maks	60 Gbps	80 Gbps	40 Gbps	80 Gbps	19 Gbps
Bandwidth jaringan maks	25 Gbps	200 Gbps	50 Gbps	100 Gbps	100 Gbps

Setiap contoh digunakan untuk tujuan yang berbeda. Memahami beban kerja SQL Server dapat membantu Anda memilih jenis instans yang terbaik untuk Anda.

Detail tentang atribut:

- r5b — Atribut “b” di r5b berarti tipe instance ini difokuskan pada kinerja EBS yang tinggi. Pada generasi kelima dari instance memori yang dioptimalkan, r5b adalah pilihan yang lebih disukai. Ini adalah tipe instance pertama yang memanfaatkan volume io2 Block Express dan mencapai IOPS penyimpanan maksimum 260.000. Jenis instans r5b masih merupakan alternatif hemat biaya untuk kebutuhan kinerja EBS yang tinggi.
- r6idn — Instans yang dioptimalkan memori generasi keenam menawarkan peningkatan yang cukup besar dibandingkan generasi sebelumnya. Peningkatan kinerja EBS dari r5b diambil selangkah lebih maju dengan r6idn, meningkatkan IOPS maksimum menjadi 350.000. R6idn juga memiliki volume penyimpanan instance untuk ekstensi tempdb dan buffer pool untuk lebih meningkatkan kinerja SQL Server.
- x2iedn — X2iedn mirip dengan r6idn. Ini menawarkan tingkat EBS yang ditingkatkan, jaringan yang ditingkatkan, dan penyimpanan instans NVMe SSD yang serupa, tetapi dengan vCPU-to-RAM rasio 1:32 untuk beban kerja memori tinggi dan kuantitas CPU yang rendah (biaya lisensi SQL Server yang lebih rendah).
- x2iezn - Atribut “z” di x2iezn menunjukkan jenis instance ini difokuskan pada kinerja prosesor yang tinggi. Prosesor Cascade Lake memiliki frekuensi turbo all-core hingga 4, 5 GHz. Kami menyarankan Anda menggunakan instans EC2 ini, ditambah dengan vCPU-to-RAM rasio 1:32, dalam skenario di mana Anda ingin menjaga kuantitas vCPU tetap rendah. Ini, pada gilirannya, dapat menjaga biaya lisensi SQL Server tetap rendah.
- r7iz — Atribut “z” di r7iz menunjukkan jenis instance ini difokuskan pada kinerja prosesor yang tinggi. Prosesor Sapphire rapids memiliki frekuensi turbo all-core hingga 3, 9. GHz Seperti contoh x2iezn, r7iz memprioritaskan kinerja prosesor frekuensi tinggi tetapi dengan rasio 1:8. vCPU-to-RAM

Sumber daya tambahan

- Instans Amazon EC2 tujuan umum (dokumentasi)AWS
- Alat perbandingan (Vantage)
- Lisensi - SQL Server (dokumentasi)AWS

Mengkonsolidasikan contoh

Bagian ini berfokus pada teknik pengoptimalan biaya menggabungkan beberapa instance SQL Server ke server yang sama untuk meminimalkan biaya lisensi dan memaksimalkan pemanfaatan sumber daya.

Ikhtisar

Membuat instance adalah bagian dari proses untuk menginstal SQL Server Database Engine. SQL Server instance adalah instalasi lengkap, berisi file server sendiri, login keamanan, dan database sistem (master, model, msdb, dan tempdb). Karena sebuah instans memiliki semua file dan layanannya sendiri, Anda dapat menginstal beberapa instance SQL Server pada sistem operasi yang sama tanpa instance mengganggu satu sama lain. Namun, karena semua instance diinstal pada server yang sama, mereka semua berbagi sumber daya perangkat keras yang sama, seperti komputasi, memori, dan jaringan.

Biasanya hanya menggunakan satu instance SQL Server per server di lingkungan produksi sehingga instance “sibuk” tidak terlalu sering menggunakan sumber daya perangkat keras bersama. Memberikan setiap instance SQL Server sistem operasinya sendiri, dengan sumber dayanya sendiri, adalah batas yang lebih baik daripada mengandalkan tata kelola sumber daya. Hal ini terutama berlaku untuk beban kerja SQL Server berkinerja tinggi yang membutuhkan sejumlah besar RAM dan sumber daya CPU.

Namun, tidak semua beban kerja SQL Server menggunakan sejumlah besar sumber daya. Misalnya, beberapa organisasi menetapkan masing-masing pelanggan mereka sendiri instance SQL Server khusus mereka untuk tujuan kepatuhan atau keamanan. Untuk klien yang lebih kecil atau klien yang biasanya tidak aktif, itu berarti menjalankan instance SQL Server dengan sumber daya minimal.

Seperti disebutkan dalam Microsoft SQL Server 2019: Panduan lisensi, setiap server yang menjalankan SQL Server harus memperhitungkan minimal empat lisensi CPU. Ini berarti bahwa bahkan jika Anda menjalankan server dengan hanya dua vCPUs, Anda masih harus melisensikan SQL Server untuk empat vCPUs. Berdasarkan harga SQL Server publik Microsoft yang selisih $3.945 jika Anda menggunakan edisi Standar SQL Server. Untuk organisasi yang menjalankan beberapa server dengan instance SQL Server tunggal yang menggunakan sumber daya minimal, biaya gabungan karena harus melisensikan sumber daya yang tidak digunakan dapat menjadi besar.

Skenario pengoptimalan biaya

Bagian ini mengeksplorasi skenario contoh yang membandingkan perbedaan antara menjalankan empat server Windows Server, masing-masing dengan satu instance SQL Server, ke satu server Windows Server yang lebih besar yang menjalankan beberapa instance SQL Server secara bersamaan.

Jika setiap instance SQL Server hanya membutuhkan dua v CPUs dan 8 GB RAM, total biaya per server adalah $7.890 untuk lisensi SQL Server selain biaya komputasi per jam sebesar $0,096.

Instans EC2	v CPUs	RAM	Harga	v CPUs untuk lisensi	Total biaya lisensi SQL Server
m6i.large	2	8	0,096	4	$7.890

Memperluas ini ke empat server, total biaya adalah $31.560 untuk lisensi SQL Server dengan biaya komputasi per jam $0.384.

Instans EC2	v CPUs	RAM	Harga	v CPUs untuk lisensi	Total biaya lisensi SQL Server
4x m6i.besar	2	32	0,384	16	$31.560

Jika Anda menggabungkan keempat instans SQL Server ke satu instans EC2, jumlah total sumber daya komputasi dan komputasi tetap sama. Namun, dengan menghapus biaya lisensi SQL Server yang tidak perlu, Anda dapat mengurangi total biaya untuk menjalankan beban kerja sebesar $15.780.

Instans EC2	v CPUs	RAM	Harga	v CPUs untuk lisensi	Total biaya lisensi SQL Server
m6i.2xlarge	8	32	0,384	8	$15.780

> **Note**
>
> Pada tabel sebelumnya, biaya komputasi menunjukkan harga sesuai permintaan per jam untuk server Amazon EC2 yang menjalankan Windows Server di Wilayah. `us-east-1` Biaya lisensi SQL Server Standard Edition mengacu pada harga SQL Server publik Microsoft.

Rekomendasi optimisasi biaya

Jika Anda mempertimbangkan untuk mengkonsolidasikan instance SQL Server, kekhawatiran terbesar adalah konsumsi sumber daya untuk setiap instance yang ingin Anda konsolidasikan. Penting untuk mendapatkan metrik kinerja dalam waktu lama untuk mendapatkan pemahaman yang lebih baik tentang pola beban kerja di setiap server. Beberapa alat umum untuk pemantauan konsumsi sumber daya adalah Amazon CloudWatch, Windows Performance Monitor (perfmon), dan alat pemantauan asli SQL Server.

Kami menyarankan Anda mempertimbangkan pertanyaan-pertanyaan berikut saat menganalisis apakah beban kerja SQL Server Anda dapat digabungkan untuk menggunakan sumber daya server yang sama tanpa mengganggu satu sama lain:

- Sumber daya apa (CPU, memori, dan bandwidth jaringan) yang dikonsumsi selama kondisi tunak Anda?
- Sumber daya apa (CPU, memori, dan bandwidth jaringan) yang dikonsumsi selama lonjakan?
- Seberapa sering paku terjadi? Apakah paku konsisten?
- Apakah lonjakan sumber daya dari satu server bertepatan dengan lonjakan sumber daya server lain?
- Apa IOPS penyimpanan dan throughput yang digunakan oleh SQL Server?

Jika Anda ingin melanjutkan dengan rencana untuk menggabungkan instans SQL Server, lihat Jalankan beberapa instance SQL Server pada satu posting instans Amazon EC2 di Blog Operasi & Migrasi Cloud. AWS Posting ini memberikan petunjuk tentang cara membuat perubahan konfigurasi di SQL Server untuk menambahkan instance tambahan. Sebelum Anda memulai, pertimbangkan perbedaan kecil ketika beberapa instance diinstal pada server yang sama:

- Contoh database SQL Server default diberi nama `MSSQLSERVER` dan menggunakan port 1433.
- Setiap instance tambahan yang diinstal pada server yang sama adalah instance database “bernama”.
- Setiap instance bernama memiliki nama instance yang unik dan port yang unik.
- Browser SQL Server harus dijalankan untuk mengoordinasikan lalu lintas ke instance bernama.
- Setiap instance dapat menggunakan lokasi terpisah untuk file data database dan login terpisah.
- Pengaturan memori server maks SQL Server harus dikonfigurasi sesuai dengan kebutuhan kinerja setiap instance, dengan total gabungannya juga meninggalkan memori yang cukup untuk sistem operasi yang mendasarinya.

- Anda dapat menggunakan kemampuan backup dan restore asli SQL Server atau AWS DMSuntuk migrasi atau konsolidasi.

Sumber daya tambahan

- Lembar Data Lisensi SQL Server (Blog Operasi & MigrasiAWS Cloud)
- SQL Server Beberapa posting blog pengaturan instance (Blog Operasi & Migrasi AWS Cloud)

Bandingkan edisi SQL Server

Ikhtisar

Lisensi Microsoft SQL Server adalah salah satu biaya terbesar untuk lingkungan beban kerja Windows. Biaya lisensi untuk SQL Server dapat dengan mudah melampaui biaya komputasi untuk menjalankan beban kerja. Jika Anda memilih edisi yang salah, Anda dapat membayar untuk fitur yang tidak Anda gunakan atau bahkan tidak perlu. Bagian ini membandingkan edisi SQL Server berikut, termasuk fitur dan biaya relatifnya:

- Enterprise — SQL Server Enterprise edition menyediakan kemampuan pusat data dengan kinerja tinggi, virtualisasi tak terbatas, dan beberapa alat intelijen bisnis (BI).
- Standar — SQL Server Standard edition menyediakan manajemen data dasar dan intelijen bisnis untuk organisasi dan departemen yang lebih kecil.
- Web — SQL Server Web edition cocok untuk perusahaan yang merupakan web hoster atau penyedia nilai tambah web ()VAPs. Edisi ini menawarkan total biaya kepemilikan yang rendah, dan menyediakan kemampuan skalabilitas dan pengelolaan untuk properti web skala kecil hingga besar.

> **Important**
>
> Anda dapat menggunakan edisi Web SQL Server untuk mendukung hanya halaman web publik dan internet yang dapat diakses, situs web, aplikasi web, dan layanan web. Anda tidak dapat menggunakan edisi Web SQL Server untuk mendukung line-of-business aplikasi (seperti manajemen hubungan pelanggan atau aplikasi manajemen sumber daya perusahaan).

- Pengembang - SQL Server Developer edition mencakup semua fungsi edisi Enterprise, tetapi ditujukan untuk tujuan pengembangan saja.
- Express — SQL Server Express edition adalah database gratis dan dapat digunakan untuk belajar atau untuk membangun aplikasi desktop. Anda dapat memperbarui edisi Express ke edisi lain.

> **Note**
>
> Edisi Evaluasi SQL Server tersedia untuk masa percobaan 180 hari.

Dampak biaya

Anda dapat membeli lisensi SQL Server dari pengecer Microsoft dan membawanya AWS dengan Jaminan Perangkat Lunak. Atau, Anda dapat menggunakan lisensi SQL Server dengan pay-as-you-go model yang memiliki Amazon EC2 yang disertakan lisensi. AMIs

Jika Anda membeli lisensi SQL Server dari Microsoft reseller, lisensi inti dijual dalam paket dua dan Anda harus melisensikan minimal empat core per server. Tabel berikut menunjukkan perbandingan biaya antara edisi Enterprise dan Standard.

Versi	Edisi SQL Server Enterprise (paket 2 core)	SQL Server edisi Standar (paket 2 core)	Tabungan
2022	$15.123	$3.945	74%
2019	$13.748	$3.586	74%

> **Note**
>
> Harga di tabel sebelumnya didasarkan pada harga publik Microsoft untuk SQL Server 2022 dan SQL Server 2019.

Perbandingan biaya berikut menunjukkan hosting edisi SQL Server yang berbeda dengan Amazon EC2 yang disertakan lisensi. AMIs Dalam perbandingan ini, SQL Server di-host di r6i.xlarge (4 vCPU) di Wilayah. `us-east-1`

Instans	Biaya komputasi	Biaya lisensi Windows	Biaya lisensi SQL Server	Total
R6i.xLarge (Linux)	$183,96	–	–	$183,96
R6i.xLarge + Windows	$183,96	$134,32	–	$318.28
R6i.xLarge + SQL Server edisi Web	$183,96	$134,32	$49,35	$367,63
R6i.xLarge + SQL Server edisi Standar	$183,96	$134,32	$350.4	$668,68
Edisi R6i.xLarge + SQL Enterprise	$183,96	$134,32	$1.095	$1,413.28

Anda dapat menghemat hingga 95 persen biaya lisensi SQL Server dengan memilih edisi SQL Server yang tepat untuk beban kerja Anda. Tabel berikut membandingkan biaya lisensi SQL Server pada instance r6i.xlarge.

Edisi	% penghematan
Standar dibandingkan dengan Enterprise	68%
Web dibandingkan dengan Standard	86%
Web dibandingkan dengan Enterprise	95%

Dalam sebagian besar skenario, organisasi beralih dari edisi Enterprise ke Standard, tetapi ada beberapa kasus di mana beralih dari edisi Standar atau Enterprise ke edisi Web dimungkinkan.

Rekomendasi optimisasi biaya

Anda dapat memilih edisi terbaik untuk beban kerja Anda berdasarkan batas penskalaan, ketersediaan tinggi, kinerja, dan keamanan. Tabel berikut menunjukkan fitur yang didukung di seluruh edisi SQL Server. Ini dapat membantu Anda memutuskan edisi mana yang akan digunakan. Perbandingan ini berlaku untuk SQL Server 2016 SP1 dan versi yang lebih baru.

Batas penskalaan

Tabel berikut membandingkan batas penskalaan edisi SQL Server yang berbeda.

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
Kapasitas komputasi maksimum yang digunakan oleh satu instance SQL Server Database Engine, SQL Server Analysis Services (SSAS), atau SQL Server Reporting Services (SSRS)	Sistem operasi maksimum	Terbatas untuk kurang dari 4 soket atau 24 core	Terbatas untuk kurang dari 4 soket atau 16 core	Terbatas untuk kurang dari 4 soket atau 4 core
Memori maksimum untuk kumpulan buffer per instance SQL Server Database Engine	Sistem operasi maksimum	128 GB	64 GB	1410 MB

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
Kapasitas maksimum untuk ekstensi kumpulan buffer per instance SQL Server Database Engine	32 kali memori maks dikonfigurasi	4 kali memori maks dikonfigurasi	N/A	N/A
Ukuran database relasional maksimum	524 PB	524 PB	524 PB	10 GB
Memori maksimum untuk cache Columnstore atau data yang dioptimalkan memori	Sistem operasi maksimum	32 GB	16 GB	352 MB

Jika aplikasi Anda membutuhkan kurang dari 16 core (32 vCPUs) dan 64 GB RAM, maka Anda dapat mulai mengevaluasi dari SQL Server Web edition. Jika beban kerja Anda membutuhkan memori lebih dari 64 GB atau opsi ketersediaan tinggi lainnya, maka Anda harus meningkatkan ke edisi Standar SQL Server.

Anda dapat menggunakan edisi Web SQL Server untuk mendukung halaman web publik dan internet yang dapat diakses, situs web, aplikasi web, dan layanan web, tetapi Anda tidak dapat menggunakan edisi Web SQL Server untuk mendukung lini aplikasi bisnis. Untuk informasi selengkapnya tentang kasus penggunaan untuk edisi Web SQL Server, hubungi Microsoft Licensing Support atau reseller Microsoft Anda.

Anda dapat menggunakan edisi Standar SQL Server untuk beban kerja hingga 24 core (48 vCPUs) dan memori 128 GB. Namun, Anda dapat menggunakan ekstensi kumpulan buffer untuk mengaktifkan edisi Standar SQL Server untuk memanfaatkan penyimpanan instans lokal, seperti

yang ada di instans EC2 r6id. Ini memperluas memori hingga ukuran empat kali konfigurasi memori maksimum. Kombinasi fitur ini dapat menunda server dari keharusan untuk meningkatkan ke edisi Enterprise ketika kebutuhan memori mulai meningkat.

Anda dapat mengidentifikasi pemanfaatan memori dengan menemukan halaman database di kumpulan buffer dan penghitung harapan hidup halaman. Harapan hidup halaman memberi tahu Anda berapa lama halaman berada dalam memori sebelum dibuang kembali ke disk. Nilai default penghitung ini adalah 300. Jika halaman berada dalam memori selama berjam-jam atau berhari-hari, maka ada kemungkinan mengurangi memori yang dialokasikan.

Ketersediaan tinggi

Tabel berikut membandingkan kemampuan ketersediaan tinggi dari edisi SQL Server yang berbeda.

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
Dukungan inti server 1	Ya	Ya	Ya	Ya
Pengiriman log	Ya	Ya	Ya	Tidak
Pencerminan basis data	Ya	Mode keamanan PENUH	Hanya sebagai saksi	Hanya sebagai saksi
Kompresi Backup	Ya	Ya	Tidak	Tidak
Selalu Pada instance cluster failover	16 node	2 node	Tidak	Tidak
Selalu Aktif pada grup ketersediaan	Hingga 8 replika sekunder, termasuk 2 replika sekunder sinkron	Tidak	Tidak	Tidak

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
Grup ketersediaan dasar	Tidak	2 node	Tidak	Tidak
Halaman online dan pemulihan file	Ya	Tidak	Tidak	Tidak
Pengindeksan online	Ya	Tidak	Tidak	Tidak
Perubahan skema online	Ya	Tidak	Tidak	Tidak
Pemulihan cepat	Ya	Tidak	Tidak	Tidak
Pencadangan cermin	Ya	Tidak	Tidak	Tidak
Tambahkan memori dan CPU panas	Ya	Tidak	Tidak	Tidak
Cadangan terenkripsi	Ya	Ya	Tidak	Tidak
Pencadangan hybrid ke Microsoft Azure (cadangan ke URL)	Ya	Ya	Tidak	Tidak
Server failover untuk pemulihan bencana	Ya	Ya	Tidak	Tidak

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
Server failover untuk ketersediaan tinggi	Ya	Ya	Tidak	Tidak

Fitur umum lainnya

Tabel berikut membandingkan fitur yang paling umum dari edisi SQL Server yang berbeda. Untuk daftar fitur yang lengkap, lihat Edisi dan fitur yang didukung SQL Server 2019 di dokumentasi Microsoft.

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
(Kinerja) Gubernur sumber daya	Ya	Tidak	Tidak	Tidak
(Keamanan) Enkripsi Database Transparan (TDE)	Ya	Ya	Tidak	Tidak
(Keamanan) Manajemen kunci yang dapat diperluas (EKM)	Ya	Tidak	Tidak	Tidak
(Replikasi) Publikasi Oracle	Ya	Tidak	Tidak	Tidak
(Replikasi) Replikasi	Ya	Tidak	Tidak	Tidak

Fitur	Edisi perusahaan	Edisi standar	Edisi web	Edisi ekspres
transaksional peer to peer				
Tangkapan data perubahan	Ya	Ya	Tidak	Tidak

Edisi Pengembang SQL Server

Semua beban kerja non-produksi, seperti pengembangan, QA, pengujian, pementasan, dan lingkungan UAT, dapat menggunakan edisi Pengembang SQL Server untuk menghemat 100 persen biaya lisensi SQL Server. Setelah mengunduh SQL Server, Anda dapat menginstal edisi SQL Server Developer pada instans EC2 dengan menggunakan penyewaan bersama. Infrastruktur khusus tidak diperlukan untuk edisi SQL Server Developer. Untuk informasi selengkapnya, lihat rekomendasi panduan ini untuk edisi SQL Server Developer.

Mengganti edisi

Untuk beban kerja yang ada, beralih dari satu edisi ke edisi lain memerlukan pengujian ekstensif. Ini adalah praktik terbaik untuk memeriksa beban kerja yang berjalan pada edisi Enterprise atau Standar untuk melihat apakah fitur khusus edisi digunakan dan apakah ada solusi alternatif untuk fitur tersebut. Misalnya, jika Anda ingin melihat apakah database Anda menggunakan fitur tingkat Enterprise, Anda dapat menjalankan tampilan manajemen dinamis (DMV) pada semua database seperti yang ditunjukkan oleh perintah contoh berikut.

```
SELECT feature_name FROM sys.dm_db_persisted_sku_features; GO
```

Ada beberapa fitur edisi Enterprise yang tidak dapat ditangkap di T-SQL, seperti pengindeksan ulang online sebagai bagian dari pekerjaan pemeliharaan SQL. Ini harus diverifikasi secara manual.

Pertimbangan migrasi

Bagaimana Anda melisensikan SQL Server akan menentukan opsi Anda untuk beralih edisi. AMI, termasuk AMI SQL Server, memiliki biaya lisensi yang termasuk dalam harga instans EC2 — biaya lisensi terikat pada AMI. Anda dapat menggunakan kode AWS penagihan untuk memverifikasi versi SQL Server yang disertakan dalam AMI Anda. Untuk instance yang AWS disertakan lisensi, mengubah edisi SQL Server di dalam sistem operasi tidak akan mengubah penagihan yang terkait

dengan AMI. Anda harus memigrasikan database Anda ke instans EC2 baru dengan AMI yang menjalankan edisi baru SQL Server.

Jika Anda membawa lisensi Anda sendiri, maka Anda memiliki lebih banyak fleksibilitas. Biasanya masih disarankan untuk bermigrasi ke instans EC2 lain yang menjalankan versi baru. Ini memungkinkan kegagalan kembali dengan mudah jika sesuatu tidak berjalan sesuai rencana. Namun, jika Anda harus menggunakan server yang ada, Anda masih dapat melakukan side-by-side instalasi SQL Server dan memigrasikan database antar instance. Untuk langkah-langkah lebih rinci tentang penurunan side-by-side versi edisi, lihat Upgrade dan Downgrade Edisi di SQL Server di situs web. MSSQLTips

Sumber daya tambahan

- Edisi dan fitur yang didukung SQL Server 2022 (Microsoft Learn)
- sys.dm_db_persisted_sku_features (Transaksi-SQL) (Microsoft Learn)
- Versi SQL Server Mana yang Harus Anda Gunakan? (Brent Ozar Tidak Terbatas)
- AWS Kalkulator Harga (AWS)

Evaluasi edisi Pengembang SQL Server

Ikhtisar

SQL Server Developer edition adalah edisi gratis SQL Server yang berisi semua fitur edisi Enterprise dan dapat digunakan dalam lingkungan non-produksi. Di cloud, di mana lisensi Microsoft Developer Network (MSDN) tidak dapat digunakan, edisi SQL Server Developer adalah cara yang baik untuk menghemat biaya tanpa harus memberikan lisensi untuk pengembangan dan pengujian beban kerja. Hal ini terutama berlaku untuk tim yang menjalankan pengembangan besar dan pengujian lingkungan dan berusaha untuk mengurangi biaya yang tidak perlu.

Lingkungan produksi didefinisikan sebagai lingkungan yang diakses oleh pengguna akhir aplikasi (seperti situs web internet) dan digunakan untuk lebih dari mengumpulkan umpan balik atau pengujian penerimaan aplikasi itu. Skenario lain yang merupakan lingkungan produksi meliputi:

- Lingkungan yang terhubung ke database produksi
- Lingkungan yang mendukung pemulihan bencana atau cadangan untuk lingkungan produksi
- Lingkungan yang digunakan untuk produksi setidaknya beberapa waktu, seperti server yang diputar ke produksi selama periode puncak aktivitas

Untuk informasi perizinan selengkapnya, lihat Amazon Web Services dan Microsoft: Pertanyaan yang Sering Diajukan dalam AWS dokumentasi.

Dampak biaya

Jika Anda menggunakan edisi SQL Server Developer untuk beban kerja non-produksi, Anda dapat menghemat 100 persen dari biaya lisensi SQL Server Anda saat ini untuk lingkungan pengembangan dan pengujian.

Versi SQL Server	Edisi SQL Server Enterprise (paket 2 core)	SQL Server edisi Standar (paket 2 core)	Edisi Pengembang SQL Server
2022	$15.123	$3.945	Kosong
2019	$13.748	$3.586	Kosong

> **Note**
>
> Harga di tabel sebelumnya didasarkan pada harga publik Microsoft untuk SQL Server 2022 dan SQL Server 2019.

Tabel berikut membandingkan biaya edisi SQL Server yang berbeda berjalan dengan 4 v CPUs dan menggunakan harga sesuai permintaan di Wilayah. `us-east-2` Ini berlaku untuk skenario yang bergantung pada instance yang disertakan lisensi dari. AWS

Instans EC2	AMI	Harga komputasi	Harga lisensi Windows	Harga lisensi SQL Server	Harga total
r5.xlarge	Linux (harga komputasi)	$183,96	–	–	$183,96
r5.xlarge	Edisi Pengembang Linux + SQL Server	$183,96	$0	$0	$183,96

Instans EC2	AMI	Harga komputasi	Harga lisensi Windows	Harga lisensi SQL Server	Harga total
r5.xlarge	Server Windows (LI)	$183,96	$134,32	–	$318.28
r5.xlarge	Edisi Pengembang Windows + SQL Server	$183,96	$134,32	$0	$318.28
r5.xlarge	Windows + SQL Server edisi Web (LI)	$183,96	$134,32	$49,64	$367,92
r5.xlarge	Windows + SQL Server edisi Standar (LI)	$183,96	$134,32	$350.4	$668,68
r5.xlarge	Edisi Windows + SQL Server Enterprise (LI)	$183,96	$134,32	$1095	$1413,28

Skenario pengoptimalan biaya

Setelah perusahaan integritas data melakukan akuisisi baru, ia ingin memigrasikan beban kerja yang baru diperoleh dari lokasi saat ini pada penyedia hosting terkelola untuk dikonsolidasikan dengan beban kerja lainnya di. AWS Cloud Harga awal menunjukkan bahwa beban kerja SQL Server perusahaan akan menelan biaya 60 persen lebih banyak AWS daripada pada penyedia layanan terkelola saat ini. Sebuah UKM MACO mengevaluasi estimasi dan menemukan bahwa pelanggan benar-benar membayar untuk lisensi SQL Server di penyedia hosting terkelola untuk pengembangan dan pengujian lingkungan mereka. Dengan mengalihkan beban kerja non-produksi ke edisi SQL Server Developer selama migrasi, perusahaan mengurangi lisensi SQL Server mereka sebesar 40 persen.

Lisensi SQL Server disertakan di Amazon EC2

Jika Anda memiliki SQL Server pada instans EC2 yang menggunakan lisensi yang disertakan AMIs, tidak mungkin melakukan konversi langsung dari edisi Enterprise ke edisi Pengembang. Biaya lisensi untuk instans yang termasuk lisensi terikat pada AMI. Bahkan jika SQL Server dihapus dari dalam sistem operasi, instans EC2 masih dikenakan biaya untuk biaya lisensi.

Untuk mengonversi ke edisi Pengembang, Anda harus mengunduh edisi SQL Server Developer, menginstalnya pada instans EC2 baru, dan kemudian memigrasikan database Anda. Anda dapat memigrasikan database SQL Server antara instans EC2 dengan menggunakan berbagai metode. Untuk informasi selengkapnya, lihat metode migrasi database SQL Server di Migrasi database Microsoft SQL Server ke panduan. AWS Cloud Anda juga dapat menggunakan solusi Pengembang SQL Server Otomatis untuk menyiapkan instance baru yang Anda rencanakan untuk dimigrasi.

SQL Server BYOL di Amazon EC2

Jika Anda memiliki instance SQL Server yang menggunakan BYOL, Anda dapat memilih dari opsi konversi atau penurunan versi di tempat berikut: side-by-side

- Unduh edisi Pengembang SQL Server dari situs web Microsoft. Untuk petunjuk penginstalan manual atau otomatis, lihat posting Automating SQL Server Developer Developer di Blog. AWS
- Gunakan cadangan asli SQL Server dan pulihkan untuk memigrasikan database atau detach/attach database dari satu instance SQL ke instance SQL lainnya.
- Gunakan alat otomatisasi untuk penyebaran massal.

> **Note**
>
> SQL Server Developer edition hanya untuk lingkungan non-produksi.

Sumber daya tambahan

- Mengotomatiskan penerapan SQL Server Developer untuk menyebarkan SQL Server Developer Edition di EC2 (Blog)AWS
- Harga SQL 2022 (Microsoft)
- Harga SQL 2019 (Microsoft)

- Opsi lisensi (SQL Server di Amazon EC2)
- AWS Kalkulator Harga(SQL Server pada dokumentasi Amazon EC2)
- Panduan Lisensi Microsoft SQL Server 2019 (unduh dari Microsoft)
- SQL Server 2022 Edisi pengembang (unduh dari Microsoft)

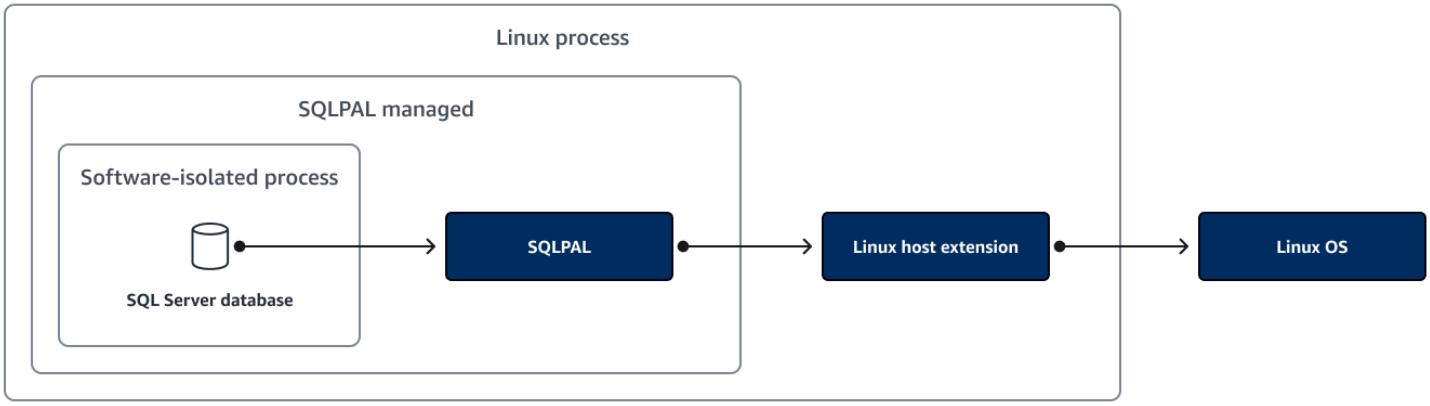
Mengevaluasi SQL Server di Linux

Ikhtisar

Sejak SQL Server 2017, dimungkinkan untuk menginstal SQL Server pada sistem operasi Linux. SQL Server di Linux siap untuk perusahaan dan menawarkan fleksibilitas, kinerja tinggi, fitur keamanan, pengurangan TCO, HA/DR fitur, dan pengalaman pengguna yang luar biasa. Anda dapat beralih dari SQL Server di Windows Server ke SQL Server di Linux untuk menghemat biaya lisensi Windows Server.

Untuk Linux, SQL Server tersedia untuk digunakan di Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Ubuntu, dan Amazon Linux 2. Mesin database SQL Server berjalan dengan cara yang sama pada Windows Server dan Linux, tetapi ada beberapa perubahan mendasar pada tugas-tugas tertentu saat menggunakan Linux. Salah satu perbedaan utama antara menjalankan aplikasi SQL Server Always On di Linux dan Windows terkait dengan pengelompokan failover. Jika Anda menyebarkan grup ketersediaan Selalu Aktif pada host Windows Server, Anda dapat memanfaatkan Windows Server Failover Clustering (WSFC) dan Active Directory sebagai fitur bawaan yang mendukung pengelompokan failover. Namun, baik WSFC maupun Active Directory tidak tersedia untuk mendukung pengelompokan failover di Linux. Jika Anda ingin meluncurkan failover clustering untuk SQL Server di Linux, Anda dapat menggunakannya AWS Launch Wizarduntuk menyederhanakan pengaturan cluster dan instalasi SQL pada instance Linux dengan menggunakan Pacemaker. ClusterLabs

SQL Server di Windows dan Linux berbagi basis kode yang sama. Artinya, mesin inti SQL Server belum diubah, sama sekali, untuk berjalan di Linux. SQL Server memperkenalkan Platform Abstraction Layer (SQLPAL), seperti yang ditunjukkan pada diagram berikut.



SQLPAL bertanggung jawab untuk abstraksi panggilan dan komunikasi antara SQL Server dan sistem operasi yang mendasarinya. Ekstensi host hanyalah aplikasi Linux asli. Fungsi sistem operasi tingkat rendah adalah panggilan asli untuk mengoptimalkan penggunaan I/O, memori, dan CPU. Ketika ekstensi host dimulai, ia memuat dan menginisialisasi SQLPAL, yang kemudian memunculkan SQL Server. SQLPAL meluncurkan proses perangkat lunak terisolasi yang menyediakan terjemahan yang diperlukan untuk sisa kode. Menambahkan lapisan baru ini ke arsitektur SQL Server berarti bahwa fitur inti tingkat perusahaan yang sama dan manfaat yang telah membuat SQL Server begitu kuat pada Windows tersedia terlepas dari sistem operasi.

Dampak biaya

Untuk contoh r5.2xlarge, pengurangan biaya lisensi Windows Server adalah sekitar $268 di setiap skenario. Pengurangan ini adalah persentase yang lebih tinggi dari total biaya server dibandingkan dengan menggunakan edisi SQL Server yang lebih murah. Tabel berikut menunjukkan penghematan biaya.

Instans	Edisi	Biaya bulanan SQL Server di Windows	Biaya bulanan SQL Server di Linux	Tabungan
r5.2xlarge	Web	$735	$466	37%
r5.2xlarge	Standar	$1,337	$1.068	20%
r5.2xlarge	Perusahaan	$2.826	$2.558	10%

> **Note**
>
> Estimasi harga pada tabel sebelumnya didasarkan pada harga sesuai permintaan di `us-east-1` Wilayah dan dapat dilihat langsung di. AWS Kalkulator Harga

Pertimbangkan contoh skenario di mana pelanggan ISV di segmen SMB ingin menghemat biaya pada lingkungan pengembangan mereka. Mereka sudah menggunakan edisi SQL Server Developer pada satu set server Windows. Dengan beralih dari Windows dengan edisi SQL Server Developer ke Linux dengan edisi SQL Server Developer, pelanggan ISV dapat menghemat 33 persen pada beban kerja pengembangan mereka. Tabel berikut menunjukkan perkiraan biaya berikut untuk skenario ini.

Perkiraan	Biaya bulanan
Windows+SQL Server	$9.307,72
Linux + Server SQL	$6.218.36
Perkiraan penghematan biaya	$3.089,36 (33%)

Dalam skenario contoh lain, perusahaan memigrasikan instance SQL Server EC2 yang disertakan lisensi dari Windows ke Linux. Perusahaan menghemat total $300.000 per tahun untuk biaya lisensi Windows Server — sekitar 20 persen dari total tagihan mereka. AWS

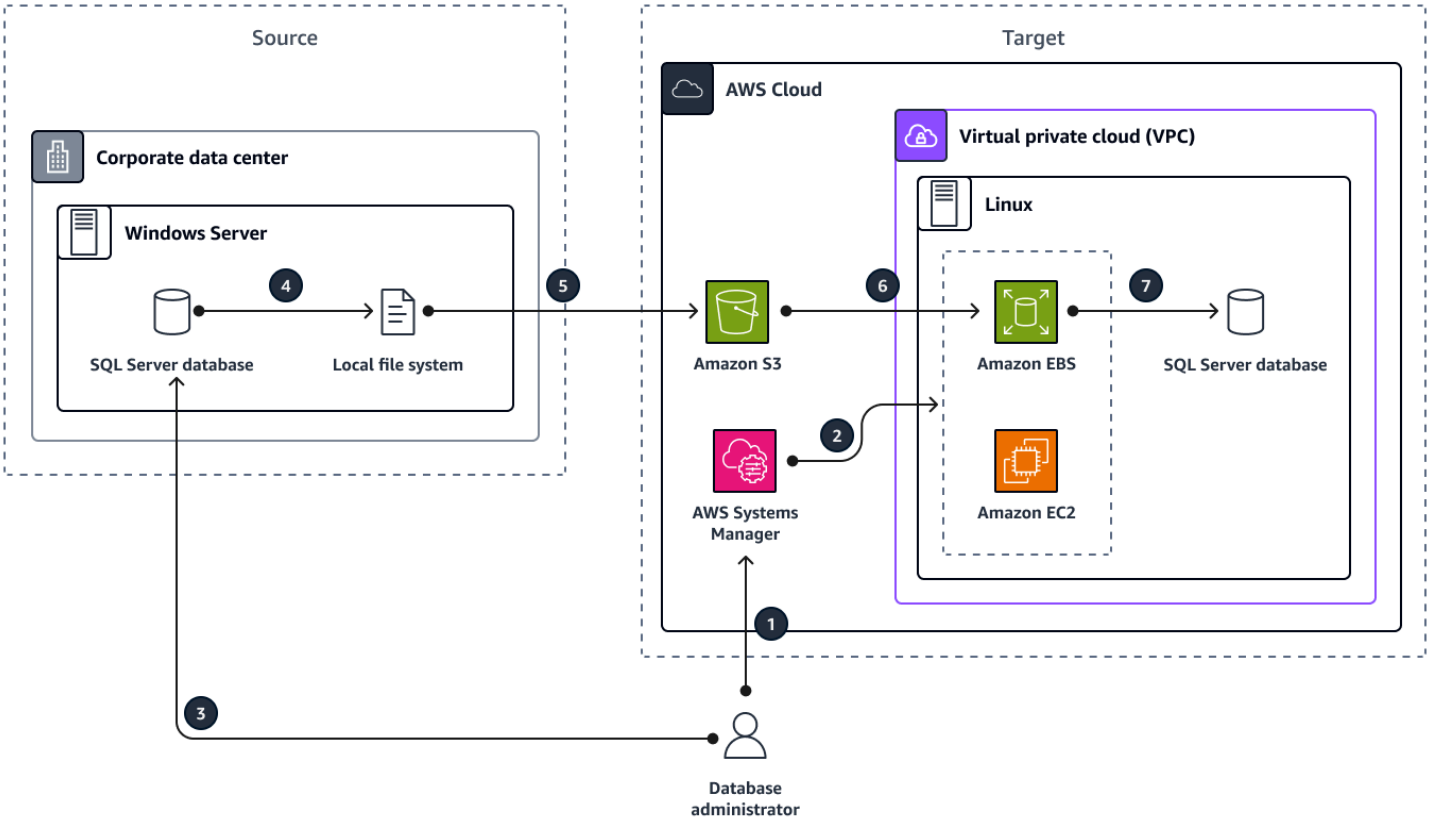
Rekomendasi optimisasi biaya

Kami menyarankan Anda mempertimbangkan hal-hal berikut:

- SQL Server di Linux didukung dimulai dengan SQL Server 2017.
- Untuk membantu beralih, Anda dapat menggunakan asisten replatforming Windows ke Linux untuk Microsoft SQL Server Databases. Asisten replatforming adalah alat skrip yang dapat membantu Anda memindahkan beban kerja SQL Server yang ada dari sistem operasi Windows ke Linux dengan memeriksa ketidakcocokan umum, mengekspor database dari host Windows, dan kemudian mengimpor database ke instans EC2 yang menjalankan Microsoft SQL Server 2017 di Ubuntu 16.04.
- Anda juga dapat menggunakan kemampuan cadangan dan pemulihan di SQL Server untuk beralih dari SQL Server di Windows ke Linux.

- Anda dapat dengan mudah dan cepat menyebarkan ke SQL Server di Linux atau Ubuntu dengan menggunakan file. AWS Launch Wizard Launch Wizard dapat menyebarkan SQL Server di Linux atau Ubuntu dalam skenario mandiri dan ketersediaan tinggi berdasarkan kebutuhan aplikasi Anda. Untuk informasi selengkapnya, lihat Menerapkan ke SQL Server Selalu di Linux dengan AWS Launch Wizard posting di Microsoft Workloads di blog. AWS

Diagram berikut menunjukkan arsitektur untuk solusi yang menggunakan asisten replatforming Windows ke Linux untuk Microsoft SQL Server Databases.



Sumber daya tambahan

- Ikhtisar SQL Server di Linux (Microsoft Learn)
- Panduan instalasi untuk SQL Server di Linux (Microsoft Learn)
- Menyebarkan ke SQL Server Selalu di Linux dengan (AWS Launch Wizard Microsoft Workloads di Blog) AWS
- SQL Server yang Sangat Tersedia di Linux (Blog Sumber AWS Terbuka)

Optimalkan strategi pencadangan SQL Server

Ikhtisar

Sebagian besar organisasi mencari solusi yang tepat untuk melindungi data mereka di SQL Server di Amazon EC2 untuk memenuhi persyaratan mereka saat ini untuk tujuan titik pemulihan (RPO), jumlah waktu maksimum yang dapat diterima sejak pencadangan terakhir, dan tujuan waktu pemulihan (RTO), penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan. Jika Anda menjalankan SQL Server pada instans EC2, Anda memiliki beberapa opsi untuk membuat cadangan data Anda dan memulihkan data Anda. Strategi Backup untuk melindungi data SQL Server di Amazon EC2 meliputi:

- Pencadangan tingkat server menggunakan snapshot Amazon Elastic Block Store (Amazon EBS) yang diaktifkan Windows Volume Shadow Copy Service (VSS) atau AWS Backup
- Pencadangan tingkat basis data menggunakan cadangan dan pemulihan asli di SQL Server

Anda memiliki opsi penyimpanan berikut untuk cadangan asli tingkat basis data:

- Cadangan lokal dengan volume Amazon EBS
- Cadangan sistem file jaringan dengan Amazon FSx untuk Windows File Server atau Amazon FSx untuk NetApp ONTAP
- Pencadangan jaringan ke Amazon Simple Storage Service (Amazon S3) menggunakan AWS Storage Gateway
- Pencadangan langsung ke Amazon S3 untuk SQL Server 2022

Bagian ini melakukan hal berikut:

- Menyoroti fitur untuk membantu Anda menghemat ruang penyimpanan
- Membandingkan biaya antara opsi penyimpanan backend yang berbeda
- Menyediakan tautan ke dokumentasi mendalam untuk membantu mengimplementasikan rekomendasi ini

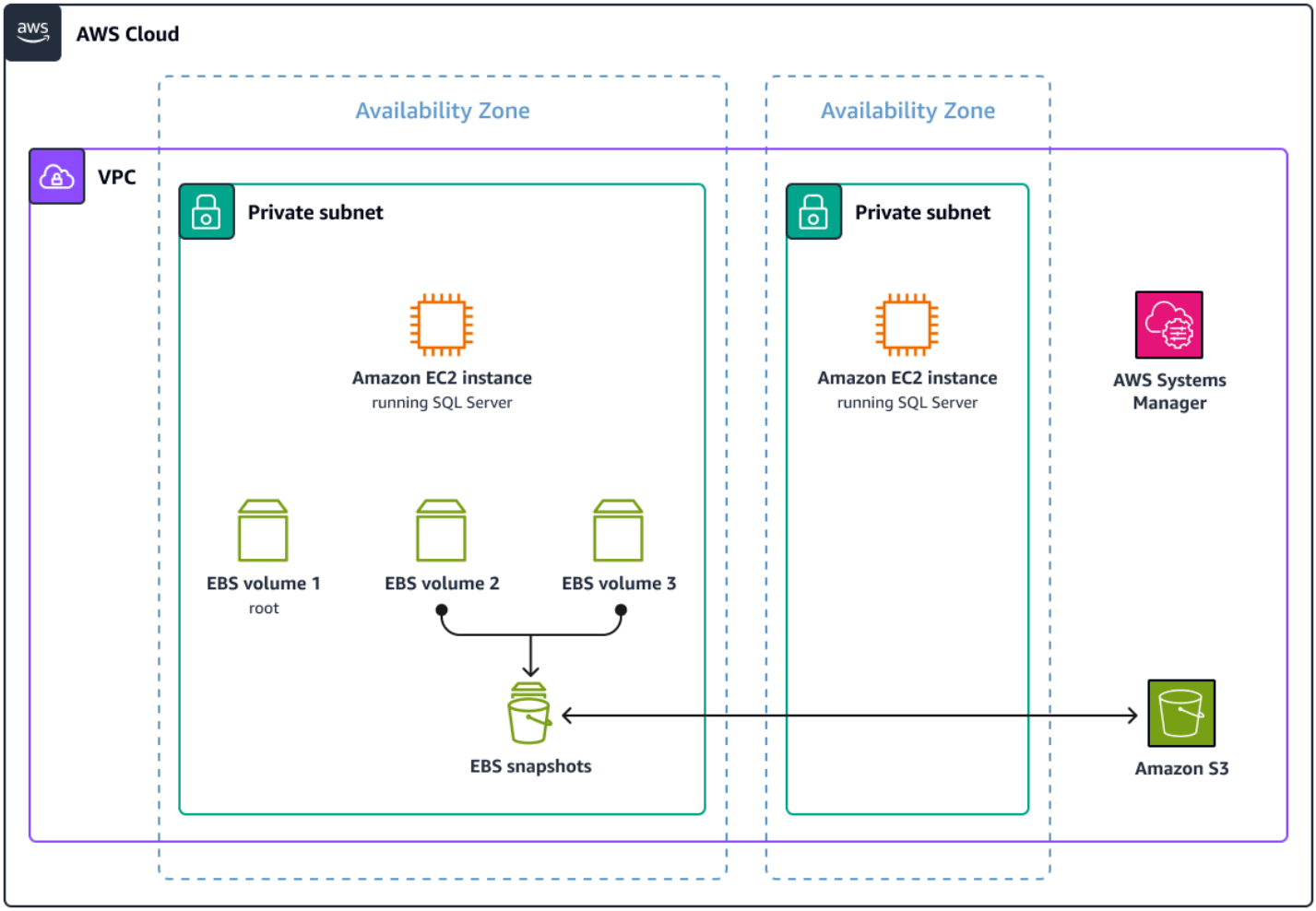
Pencadangan tingkat server menggunakan snapshot berkemampuan VSS

Arsitektur snapshot berkemampuan VSS menggunakan AWS Systems Manager Run Command untuk menginstal agen VSS pada instance SQL Server Anda. Anda juga dapat menggunakan Run

Command untuk memanggil seluruh alur kerja sistem operasi pembilasan dan buffer aplikasi ke disk, menjeda I/O operasi, mengambil point-in-time snapshot dari volume EBS, dan kemudian melanjutkan I/O.

Perintah Jalankan ini membuat snapshot otomatis dari semua volume EBS yang dilampirkan ke instance target. Anda juga memiliki opsi untuk mengecualikan volume root, karena file database pengguna biasanya disimpan pada volume lain. Jika Anda melakukan stripe beberapa volume EBS untuk membuat satu sistem file untuk file SQL Server, Amazon EBS juga mendukung snapshot multivolume yang konsisten dengan crash menggunakan satu perintah API. Untuk informasi selengkapnya tentang snapshot EBS berkemampuan VSS yang konsisten dengan aplikasi, lihat Membuat snapshot yang konsisten dengan aplikasi VSS dalam dokumentasi Amazon EC2.

Diagram berikut menunjukkan arsitektur untuk pencadangan tingkat server menggunakan snapshot berkemampuan VSS.



Pertimbangkan manfaat berikut menggunakan snapshot berkemampuan VSS:

- Snapshot pertama instans DB berisi data untuk instans DB penuh. Snapshot berikutnya dari instans DB yang sama bersifat inkremental, yang berarti hanya data yang telah berubah setelah snapshot terbaru Anda disimpan.
- Snapshot EBS memberikan point-in-time pemulihan.
- Anda dapat mengembalikan ke instans SQL Server EC2 baru dari snapshot.
- Jika instance dienkripsi menggunakan Amazon EBS atau jika database dienkripsi dalam instance menggunakan TDE, instance atau database tersebut akan dipulihkan secara otomatis dengan enkripsi yang sama.
- Anda dapat menyalin backup Lintas wilayah otomatis Anda.
- Saat Anda mengembalikan volume EBS dari snapshot, volume EBS akan segera tersedia bagi aplikasi untuk mengaksesnya. Ini berarti Anda dapat segera membawa SQL Server online setelah memulihkan satu atau lebih volume EBS yang mendasarinya dari snapshot.
- Secara default, volume yang dipulihkan mengambil blok yang mendasari dari Amazon S3 saat pertama kali aplikasi mencoba membacanya. Ini berarti bahwa mungkin ada lag dalam kinerja setelah volume EBS dipulihkan dari snapshot. Volume akhirnya menyusul kinerja nominal. Namun, Anda dapat menghindari kelambatan itu dengan menggunakan snapshot snapshot-restore (FSR) cepat.
- Anda dapat menggunakan manajemen siklus hidup untuk snapshot EBS.

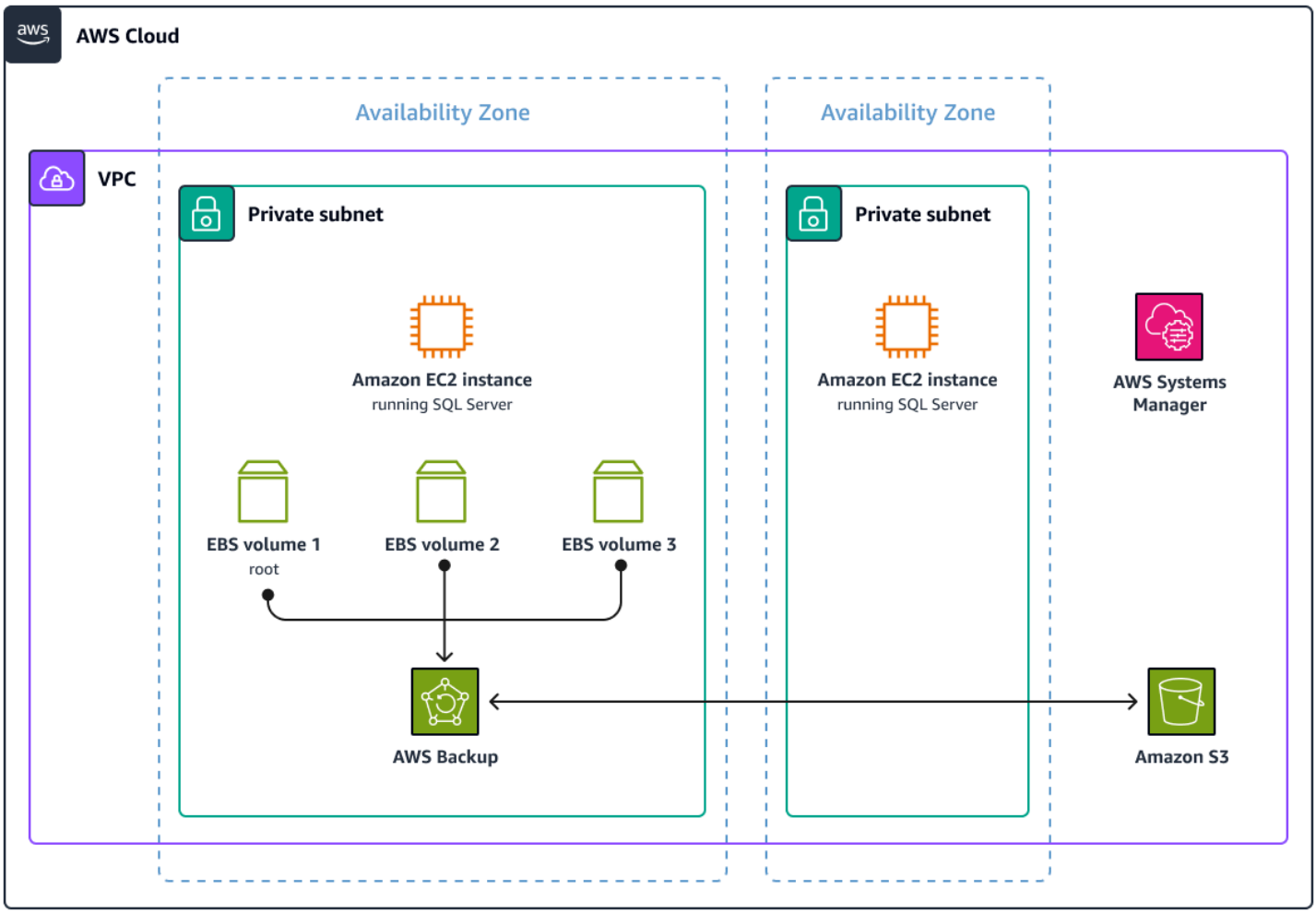
Pertimbangkan batasan penggunaan snapshot berkemampuan VSS berikut ini:

- Anda tidak dapat melakukan point-in-time pemulihan lintas wilayah dengan snapshot terenkripsi untuk instance SQL Server.
- Anda tidak dapat membuat snapshot terenkripsi dari instance yang tidak terenkripsi.
- Anda tidak dapat memulihkan database individual karena snapshot diambil pada tingkat volume EBS.
- Anda tidak dapat mengembalikan instance ke dirinya sendiri.
- Sebuah snapshot dari instans DB harus dienkripsi dengan menggunakan kunci AWS Key Management Service (AWS KMS) yang sama dengan instans DB.
- Penyimpanan I/O ditangguhkan selama sepersekian detik (sekitar 10 milidetik) selama proses pencadangan snapshot.

Pencadangan SQL Server menggunakan AWS Backup

Anda dapat menggunakan AWS Backupuntuk memusatkan dan mengotomatiskan perlindungan data di seluruh. Layanan AWS AWS Backup menawarkan solusi berbasis kebijakan yang hemat biaya, dikelola sepenuhnya, yang menyederhanakan perlindungan data dalam skala besar. AWS Backup juga membantu Anda mendukung kewajiban kepatuhan terhadap peraturan Anda dan memenuhi tujuan kelangsungan bisnis Anda. Bersama dengan AWS Organizations, AWS Backup Anda dapat menerapkan kebijakan perlindungan data (backup) secara terpusat untuk mengonfigurasi, mengelola, dan mengatur aktivitas pencadangan di seluruh organisasi dan sumber daya Anda. Akun AWS

Diagram berikut menunjukkan arsitektur solusi cadangan dan pemulihan untuk SQL Server pada EC2 dengan menggunakan. AWS Backup



Pertimbangkan manfaat berikut dari membuat cadangan SQL Server dengan menggunakan: AWS Backup

- Anda dapat mengotomatiskan penjadwalan cadangan, manajemen retensi, dan manajemen siklus hidup.
- Anda dapat memusatkan strategi pencadangan di seluruh organisasi Anda, mencakup beberapa akun dan. Wilayah AWS
- Anda dapat memusatkan pemantauan aktivitas pencadangan dan peringatan di seluruh. Layanan AWS
- Anda dapat menerapkan backup lintas wilayah untuk perencanaan pemulihan bencana.
- Solusinya mendukung pencadangan lintas akun.
- Anda dapat melakukan backup aman dengan enkripsi cadangan sekunder.
- Semua backup mendukung enkripsi dengan menggunakan kunci AWS KMS enkripsi.
- Solusinya bekerja dengan TDE.
- Anda dapat mengembalikan ke titik pemulihan tertentu dari AWS Backup konsol.
- Anda dapat mencadangkan seluruh instance SQL Server, yang mencakup semua database SQL Server.

Pencadangan tingkat basis data

Pendekatan ini menggunakan fungsionalitas cadangan Microsoft SQL Server asli. Anda dapat mengambil backup database individu pada instance SQL Server dan mengembalikan database individual.

Masing-masing opsi ini untuk cadangan dan pemulihan SQL Server asli juga mendukung yang berikut:

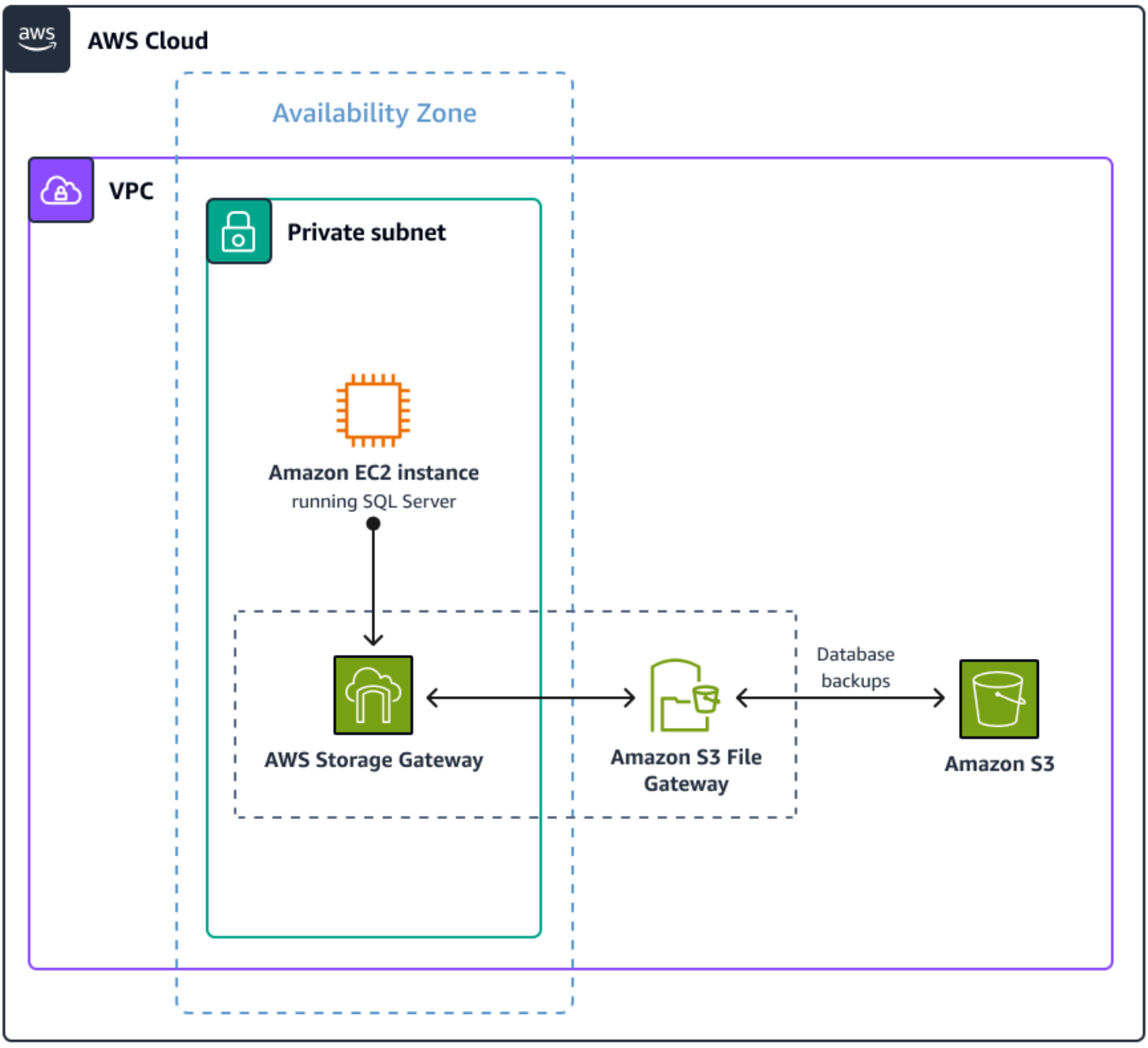
- Kompresi dan cadangan beberapa file
- Pencadangan penuh, diferensial, dan T-log
- Database terenkripsi TDE

Pencadangan asli SQL Server dan pulihkan ke Amazon S3

SQL Server di Amazon EC2 mendukung pencadangan dan pemulihan asli untuk database SQL Server. Anda dapat mengambil cadangan database SQL Server Anda dan kemudian memulihkan file cadangan ke database yang ada atau ke instans SQL Server EC2 baru, Amazon RDS for SQL Server, atau server lokal.

Storage Gateway adalah layanan penyimpanan cloud hybrid yang menyediakan aplikasi lokal dengan akses ke penyimpanan cloud yang hampir tidak terbatas. Anda dapat menggunakan Storage Gateway untuk mencadangkan database Microsoft SQL Server langsung ke Amazon S3, mengurangi jejak penyimpanan lokal dan menggunakan Amazon S3 untuk penyimpanan yang tahan lama, terukur, dan hemat biaya.

Diagram berikut menunjukkan arsitektur solusi backup dan restore asli yang menggunakan Storage Gateway dan Amazon S3.



Pertimbangkan manfaat berikut menggunakan cadangan SQL Server asli dengan Storage Gateway:

- Anda dapat memetakan gateway penyimpanan sebagai berbagi file Server Message Block (SMB) pada instans EC2 dan mengirim cadangan ke Amazon S3.
- Cadangan langsung masuk ke bucket S3 atau melalui cache file Storage Gateway.
- Backup multi-file didukung.

Pertimbangkan batasan cadangan asli berikut menggunakan Storage Gateway:

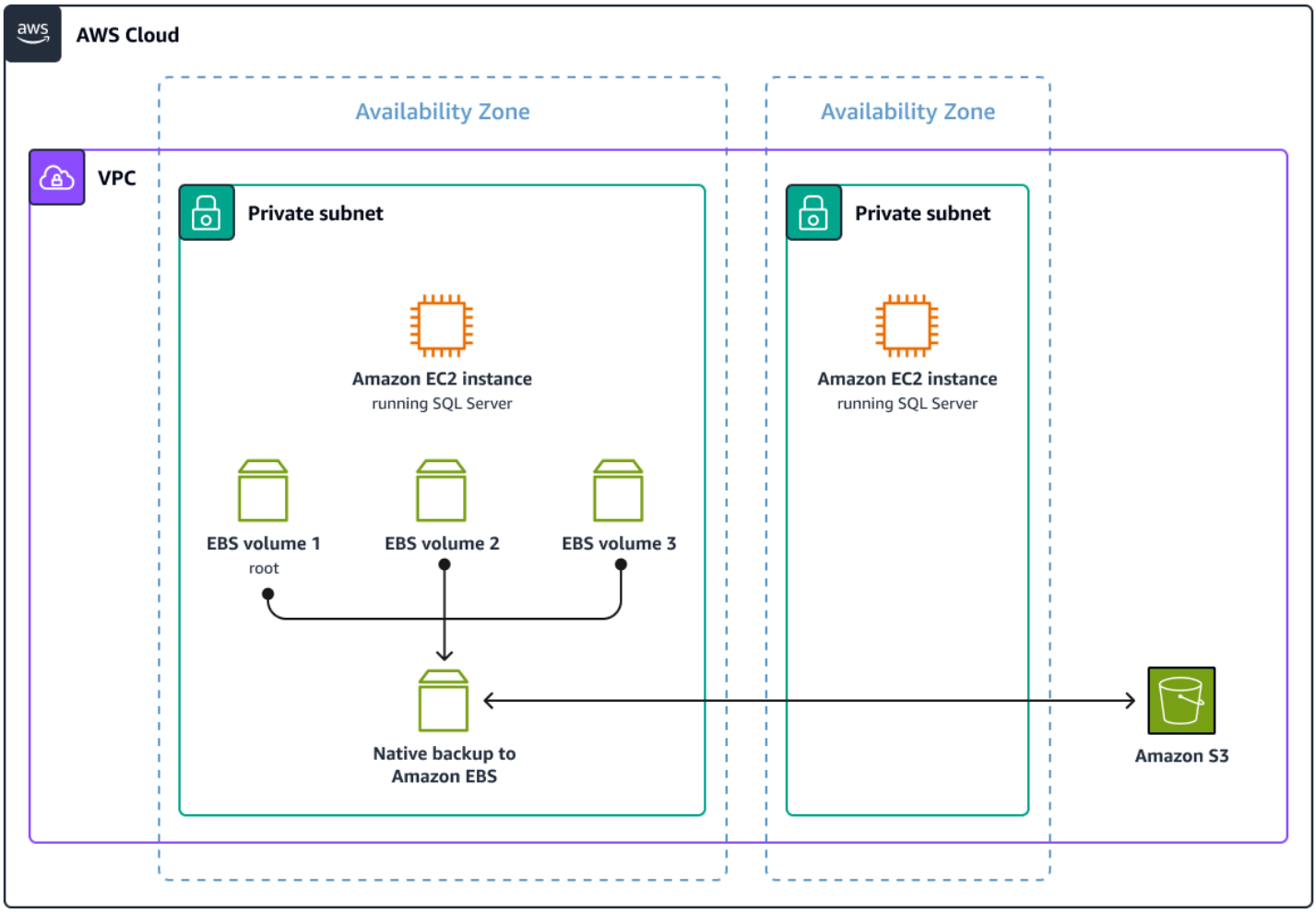
- Anda harus mengatur cadangan dan pemulihan untuk setiap database individu.
- Anda harus mengelola kebijakan siklus hidup Amazon S3 untuk file cadangan.

Untuk informasi selengkapnya tentang cara mengatur Storage Gateway, lihat backup Store SQL Server di Amazon S3 menggunakan AWS Storage Gateway postingan di Blog. AWS

Pencadangan asli SQL Server ke volume EBS

Anda dapat mengambil cadangan asli database SQL Server Anda dan menyimpan file dalam volume Amazon EBS. Amazon EBS adalah layanan penyimpanan blok berkinerja tinggi. Volume EBS elastis, yang mendukung enkripsi. Mereka dapat dilepas dan dilampirkan ke instance EC2. Anda dapat mencadangkan SQL Server pada instans EC2 pada jenis volume EBS yang sama atau pada jenis volume EBS yang berbeda. Salah satu keuntungan dari membuat cadangan ke volume EBS yang berbeda adalah penghematan biaya.

Diagram berikut menunjukkan arsitektur cadangan asli ke volume EBS.



Pertimbangkan manfaat berikut menggunakan cadangan asli SQL Server untuk volume EBS:

- Anda dapat mengambil backup database individual pada instance SQL Server EC2 dan mengembalikan database individual alih-alih harus mengembalikan instance lengkap.
- Backup multi-file didukung.
- Anda dapat menjadwalkan pekerjaan cadangan dengan menggunakan SQL Server Agent dan mesin kerja SQL Server.
- Anda bisa mendapatkan manfaat kinerja melalui pilihan perangkat keras Anda. Misalnya, Anda dapat menggunakan volume penyimpanan st1 untuk mencapai throughput yang lebih tinggi.

Pertimbangkan batasan berikut menggunakan cadangan asli untuk volume EBS:

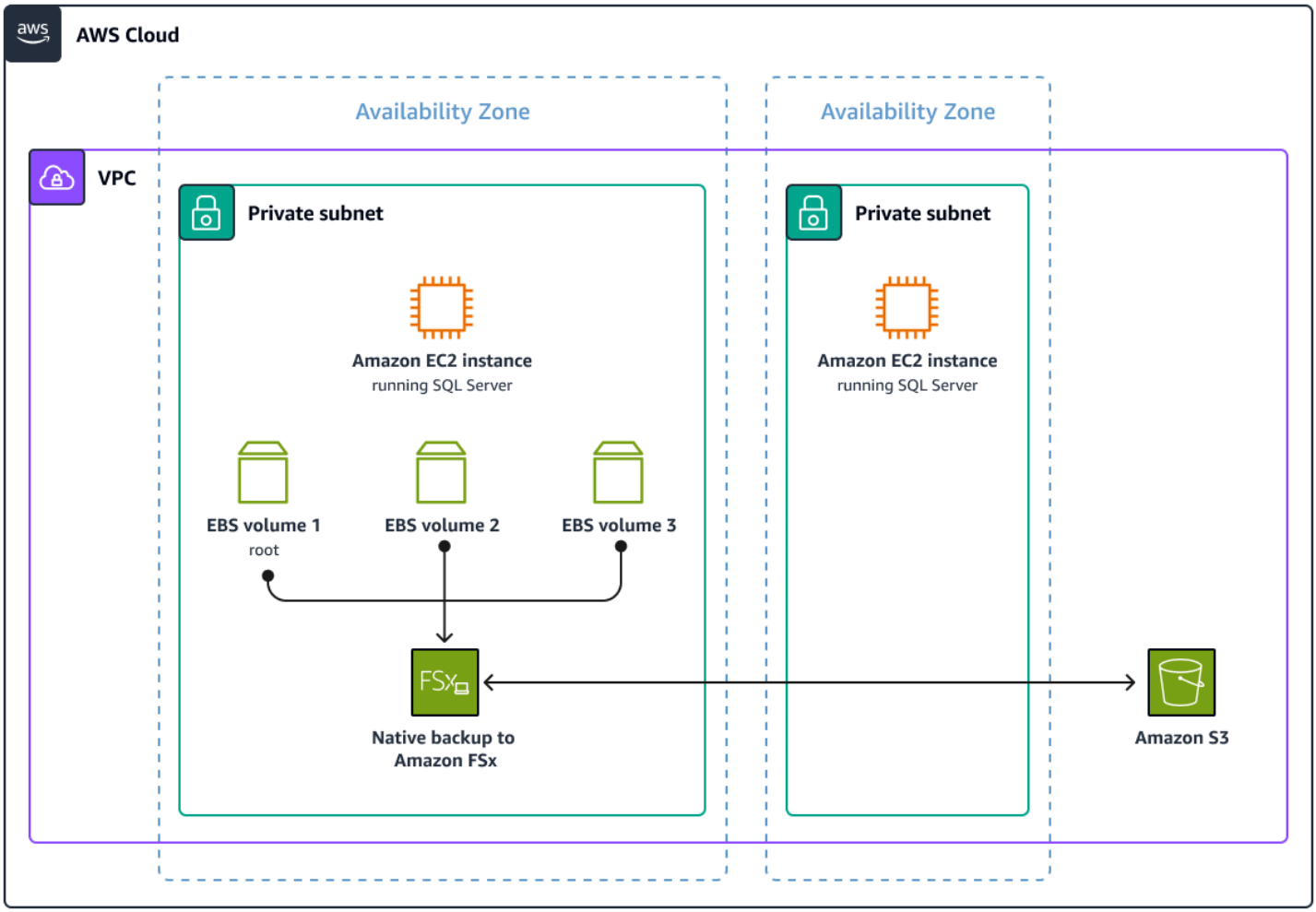
- Anda harus memindahkan cadangan secara manual ke Amazon S3 dari volume EBS.
- Untuk cadangan besar, Anda harus mengelola ruang disk di Amazon EC2.

- Pada instans EC2, throughput Amazon EBS dapat menjadi hambatan.
- Penyimpanan tambahan diperlukan untuk menyimpan cadangan di Amazon EBS.

Pencadangan asli SQL Server ke Amazon FSx untuk Windows File Server

Amazon FSx untuk Windows File Server adalah sistem file Windows asli yang dikelola sepenuhnya yang menawarkan penyimpanan hingga 64 TB yang dirancang untuk memberikan kinerja yang cepat, dapat diprediksi, dan konsisten. AWS memperkenalkan dukungan asli untuk penyebaran sistem file Multi-AZ FSx untuk Windows File Server. Dukungan asli membuatnya lebih mudah untuk menyebarkan penyimpanan file Windows AWS dengan ketersediaan tinggi dan redundansi di beberapa Availability Zone. AWS juga memperkenalkan dukungan untuk berbagi file SMB Continuously Available (CA). Anda dapat menggunakan FSx untuk Windows File Server sebagai penyimpanan cadangan untuk database SQL Server.

Diagram berikut menunjukkan arsitektur cadangan SQL Server asli FSx untuk Windows File Server.



Pertimbangkan manfaat berikut menggunakan cadangan SQL Server asli FSx untuk Windows File Server:

- Anda dapat mencadangkan database SQL Server Anda ke berbagi FSx file Amazon.
- Anda dapat mengambil backup database individual pada instance SQL Server dan mengembalikan database individual daripada harus mengembalikan instance lengkap.
- Cadangan multi-bagian didukung.
- Anda dapat menjadwalkan pekerjaan cadangan dengan menggunakan SQL Server Agent dan mesin pekerjaan.
- Instans memiliki bandwidth jaringan yang lebih tinggi dibandingkan dengan Amazon EBS.

Pertimbangkan batasan berikut menggunakan cadangan SQL Server asli FSx untuk Windows File Server:

- Anda harus memindahkan cadangan secara manual ke Amazon S3 dari FSx Amazon AWS Backup dengan menggunakan atau. AWS DataSync
- Pencadangan besar mungkin memerlukan overhead tambahan untuk manajemen ruang disk di Amazon. FSx
- Throughput jaringan instans EC2 dapat menjadi hambatan.
- Penyimpanan tambahan diperlukan untuk menyimpan cadangan FSx untuk Windows File Server.

Pencadangan SQL Server ke Amazon FSx untuk NetApp ONTAP

Snapshot dengan FSx untuk ONTAP selalu konsisten crash, tetapi mereka mengharuskan Anda untuk diam (atau menjeda) database Anda untuk membuat snapshot yang konsisten dengan aplikasi. I/O Anda dapat menggunakan NetApp SnapCenter (alat orkestrasi dengan plug-in untuk aplikasi tertentu, termasuk SQL Server) dengan ONTAP FSx untuk membuat snapshot yang konsisten aplikasi dan melindungi, mereplikasi, dan mengkloning database Anda tanpa biaya tambahan.

NetApp SnapCenter

NetApp SnapCenter adalah platform terpadu untuk perlindungan data yang konsisten aplikasi. SnapCenter mengacu pada snapshot sebagai cadangan. Panduan ini mengadopsi konvensi penamaan yang sama. SnapCenter menyediakan satu panel kaca untuk mengelola pencadangan, pemulihan, dan klon yang konsisten aplikasi. Anda menambahkan SnapCenter plug-in untuk aplikasi database spesifik Anda untuk membuat backup yang konsisten aplikasi. SnapCenter Plug-in untuk

SQL Server menyediakan fungsionalitas berikut yang menyederhanakan alur kerja perlindungan data Anda.

- Backup dan restore pilihan dengan granularitas untuk backup penuh dan log
- Pemulihan dan pemulihan di tempat ke lokasi alternatif

Untuk informasi selengkapnya SnapCenter, lihat Melindungi beban kerja SQL Server Anda menggunakan dengan NetApp SnapCenter Amazon FSx untuk NetApp ONTAP posting di Blog Penyimpanan. AWS

Optimalisasi biaya untuk backup

Opsi berikut dapat membantu Anda mengurangi biaya penyimpanan cadangan SQL Server. AWS

- Aktifkan kompresi SQL Server selama pembuatan file cadangan dan kirim file sekecil mungkin ke penyimpanan. Misalnya, rasio kompresi 3:1 menunjukkan bahwa Anda menghemat sekitar 66 persen pada ruang disk. Untuk query pada kolom ini, Anda dapat menggunakan pernyataan Transact-SQL berikut:. `SELECT backup_size/compressed_backup_size FROM msdb..backupset;`
- Untuk cadangan yang masuk ke bucket S3, aktifkan kelas penyimpanan Amazon S3 Intelligent-Tiering untuk mengurangi biaya penyimpanan hingga 30 persen.
- Untuk cadangan untuk Windows File Server atau FSx FSx untuk ONTAP, gunakan Availability Zone tunggal untuk penghematan biaya 50 persen (dibandingkan dengan menggunakan beberapa Availability Zone). Untuk informasi harga, lihat Harga Amazon FSx untuk Windows File Server dan Amazon FSx untuk Harga NetApp ONTAP.
- Opsi paling efisien untuk SQL Server 2022 adalah pencadangan langsung ke Amazon S3. Anda dapat menghemat biaya tambahan dengan menghindari Storage Gateway.

Hasil tes benchmark untuk backup

Bagian ini membandingkan opsi berikut dari sudut pandang biaya dan kinerja untuk database sampel 1 TB, berdasarkan hasil pengujian benchmark kinerja pada solusi cadangan yang tercakup dalam panduan ini.

- Spesifikasi instans EC2 - r5d.8xlarge dengan Windows Server 2019 dan edisi Pengembang SQL Server 2019
- Spesifikasi basis data - ukuran 1 TB dengan TDE dinonaktifkan

Pengujian dilakukan dengan instance r5d.8xlarge dan database SQL Server 1 TB sebagai sumbernya. Sistem sumber dikonfigurasi sesuai dengan praktik terbaik, dan database sumber berisi empat file data (masing-masing 250 GB) dan satu file log (50 GB) yang tersebar di volume gp3 yang terpisah. `BACKUP`Perintah asli SQL Server mencakup penulisan ke 10 file cadangan, menggunakan kompresi untuk mengoptimalkan kinerja cadangan dan mengurangi jumlah data yang dikirim ke seluruh jaringan dan ditulis ke target. Dalam semua kasus uji, kinerja penyimpanan adalah hambatan.

Ada berbagai kemungkinan konfigurasi yang hampir tak ada habisnya untuk jenis pengujian ini. Tes ini berfokus pada pengoptimalan kinerja, biaya, skalabilitas, dan kasus penggunaan dunia nyata. Tabel berikut menunjukkan metrik kinerja yang diambil untuk opsi target cadangan.

Opsi Backup	Tingkat	Durasi lari (Appx)	Tingkat Backup	Biaya USD per bulan*
Cadangan asli ke HDD EBS st1 lokal, 2 TB	Basis Data	00:30:46 mnt	554,7 Mbps	$92,16
Cadangan asli ke EBS SSD gp3 lokal, 2 TB	Basis Data	00:22:00 mnt	512 Mbps	$193,84
Cadangan asli FSx untuk HDD Server File Windows, throughput 2 TB @512 Mbps	Basis Data	00:20:58 mnt	814,0 Mbps	$1.146
Cadangan asli FSx untuk Windows File Server SSD, throughput 2 TB @512 Mbps	Basis Data	00:20:00 mnt	814,0 Mbps	$1,326
Cadangan asli ke S3	Basis Data	00:23:20 mnt	731,5 Mbps	$470.42

Opsi Backup	Tingkat	Durasi lari (Appx)	Tingkat Backup	Biaya USD per bulan*
File Gateway m6i.4xlarge (16 vCPU, 64 GB) dengan 2 TB gp3				
Cuplikan EBS VSS	Volume EBS	00:00:02 dtk 00:00:53 dtk	Cuplikan N/A	$51
AWS Backup (Cadangan AMI)	AMI	00:00:04 dtk 00:08:00 mnt	Cuplikan N/A	$75
Pencadangan SQL Server asli langsung ke Amazon S3 (SQL Server 2022)	Basis Data	00:12:00 mnt	731,5 Mbps	50 TB/Bulan Pertama, $0,023 per GB $23,55 per bulan
Pencadangan asli FSx untuk ONTAP (menggunakan SnapCenter)	Basis Data	–	–	$440,20

Tabel sebelumnya mengasumsikan hal berikut:

- Transfer data dan biaya Amazon S3 tidak termasuk.
- Harga penyimpanan sudah termasuk dalam harga instans.
- Biaya berbasis di `us-east-1` Wilayah.
- Throughput dan IOPS tumbuh sebesar 10 persen dengan beberapa cadangan yang memiliki tingkat perubahan keseluruhan 10 persen selama sebulan.

Hasil pengujian menunjukkan bahwa opsi tercepat adalah cadangan database SQL Server asli FSx untuk Windows File Server. Cadangan ke Storage Gateway dan volume EBS yang terpasang secara lokal adalah opsi yang lebih hemat biaya tetapi memiliki kinerja yang lebih lambat. Untuk pencadangan tingkat server (AMI), sebaiknya AWS Backup gunakan untuk kinerja, biaya, dan pengelolaan yang optimal.

Rekomendasi optimisasi biaya

Memahami solusi yang mungkin untuk mencadangkan SQL Server di Amazon EC2 adalah kunci untuk melindungi data Anda, memastikan bahwa Anda memenuhi kebutuhan cadangan, dan menyiapkan rencana untuk memulihkan dari peristiwa penting. Berbagai cara untuk mencadangkan dan memulihkan instans SQL Server dan database yang dieksplorasi di bagian ini dapat membantu Anda menyusun strategi pencadangan dan pemulihan yang melindungi data Anda dan memenuhi persyaratan organisasi Anda.

Bagian ini mencakup opsi cadangan berikut:

- Kompresi
- Amazon S3 Intelligent-Tiering
- Zona Ketersediaan Tunggal
- Backup ke URL

Panduan yang diberikan untuk masing-masing opsi ini adalah tingkat tinggi. Jika Anda ingin menerapkan salah satu rekomendasi ini di organisasi Anda, kami sarankan Anda menghubungi tim akun Anda. Tim kemudian dapat terlibat dengan Microsoft Specialist SA untuk memimpin percakapan. Anda juga dapat menghubungi dengan mengirim email ke optimize-microsoft@amazon.com.

Singkatnya, kami merekomendasikan yang berikut:

- Jika Anda menggunakan SQL Server 2022, mencadangkan ke Amazon S3 adalah opsi yang paling hemat biaya.
- Jika Anda menggunakan SQL Server 2019 dan edisi SQL Server sebelumnya, pertimbangkan untuk membuat cadangan ke Storage Gateway yang didukung oleh Amazon S3 sebagai opsi yang paling hemat biaya.

Kompresi

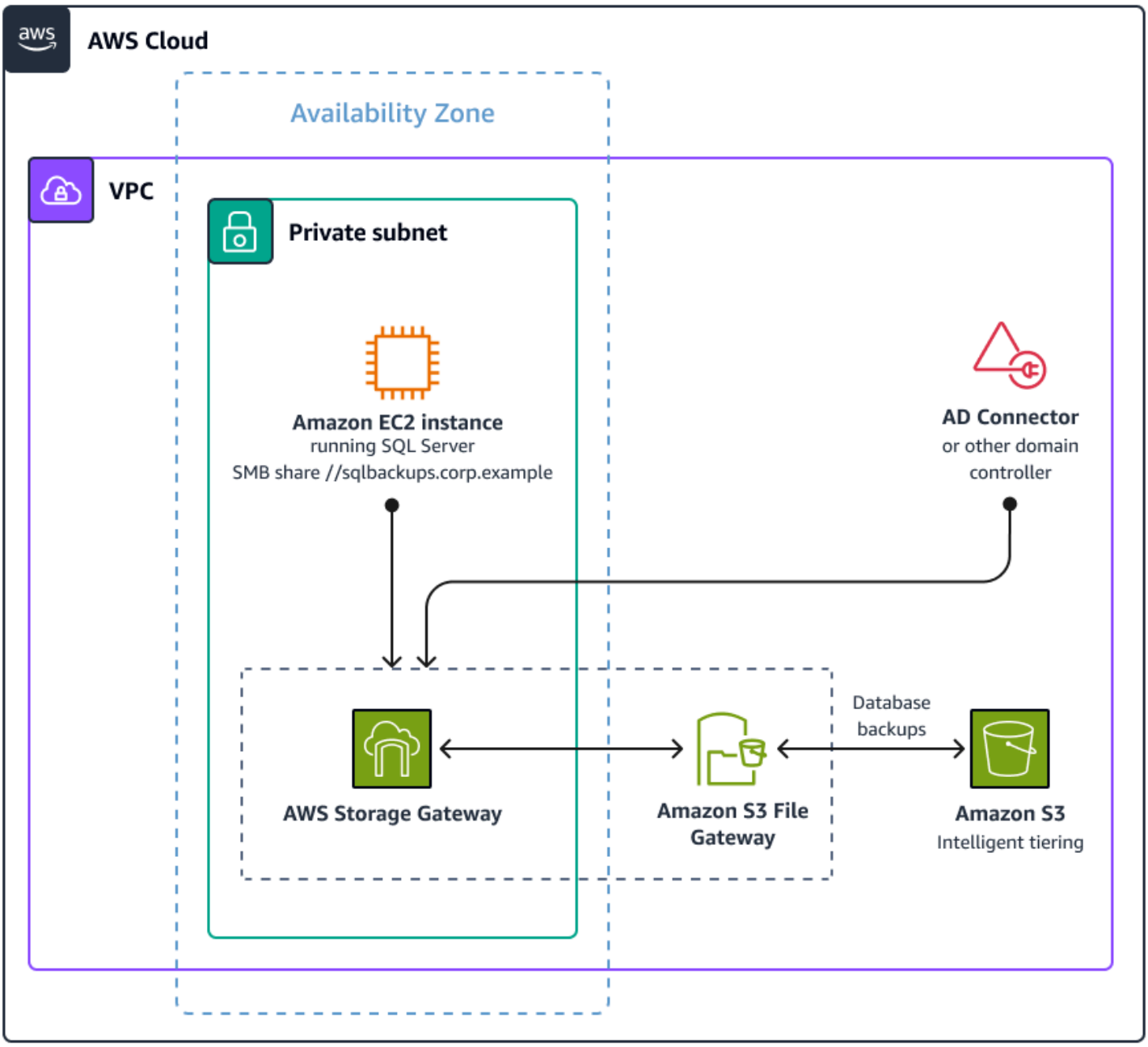
Tujuan kompresi adalah untuk memiliki lebih sedikit penyimpanan yang dikonsumsi oleh setiap cadangan, yang bermanfaat untuk berbagai opsi penyimpanan. Anda harus mengaktifkan kompresi untuk cadangan SQL Server pada tingkat instance SQL Server. Contoh berikut menunjukkan cara menambahkan kata kunci kompresi dengan database cadangan:

```
BACKUP DATABASE <database_name> TO DISK WITH COMPRESSION (ALGORITHM =
QAT_DEFLATE)
```

Amazon S3 Intelligent-Tiering

Untuk pencadangan yang masuk ke bucket Amazon S3, Anda dapat mengaktifkan Amazon S3 Intelligent-Tiering sebagai kelas penyimpanan Amazon S3 File Gateway Anda. Ini dapat mengurangi biaya penyimpanan hingga 30 persen. Anda kemudian memasang S3 File Gateway ke server SQL Anda dengan menggunakan berbagi file SMB yang dapat diintegrasikan dengan domain Active Directory Anda. Ini memberi Anda kontrol akses untuk berbagi Anda, kemampuan untuk memanfaatkan akun layanan yang ada, dan akses ke Amazon S3 menggunakan protokol file fokus Microsoft yang umum. Untuk akun yang mungkin tidak memiliki konektivitas langsung ke pengontrol domain, Anda dapat menggunakan Konektor Direktori Aktif untuk memfasilitasi komunikasi dengan Active Directory lokal atau di cloud. Untuk mengkonfigurasi pengaturan Direktori Aktif pada gateway, Anda harus menentukan Konektor Direktori Aktif IPs untuk pengontrol domain untuk permintaan proxy ke Active Directory.

Diagram berikut menunjukkan arsitektur untuk solusi berdasarkan S3 Intelligent-Tiering.



Secara default, file cadangan yang ditulis ke bucket S3 menggunakan tingkat Standar. Untuk mengonversi file cadangan dari tingkat Standar ke S3 Intelligent-Tiering, Anda harus membuat aturan siklus hidup. Anda juga dapat menggunakan Konsol Manajemen AWSuntuk mengaktifkan S3 Intelligent-Tiering. Untuk informasi selengkapnya, lihat Memulai Menggunakan Amazon S3 Intelligent-Tiering dalam dokumentasi. AWS

Zona Ketersediaan Tunggal

Untuk membuat sistem file Zona Ketersediaan Tunggal, pilih opsi Single-AZ saat Anda membuat sistem file FSx untuk Windows File Server. Amazon FSx juga mengambil cadangan yang sangat tahan lama (disimpan di Amazon S3) dari sistem file Anda setiap hari menggunakan Windows Volume Shadow Copy Service, dan memungkinkan Anda untuk mengambil cadangan tambahan kapan saja. Ingatlah beberapa masalah dengan menggunakan Zona Ketersediaan Tunggal. Misalnya, berbagi file SMB menjadi tidak dapat diakses jika Availability Zone yang terpengaruh di mana sistem file disediakan turun selama berjam-jam pada suatu waktu. Jika Anda memerlukan akses ke data, Anda harus memulihkannya dari cadangan di Availability Zone yang tersedia di dalam Wilayah sumber. Untuk informasi selengkapnya, lihat bagian Gunakan Zona Ketersediaan tunggal pada panduan ini.

Backup ke URL

Untuk SQL Server 2022, fitur backup ke URL memungkinkan pencadangan langsung ke Amazon S3. Ini adalah pendekatan pencadangan ideal untuk SQL Server 2022 yang berjalan AWS saat Anda mendapatkan set fitur lengkap Amazon S3 di lapisan penyimpanan dan menghapus biaya alat yang diperlukan dalam versi sebelumnya untuk memfasilitasi fungsionalitas ini. AWS Storage Gateway Ada dua biaya utama yang perlu dipertimbangkan saat menerapkan fitur ini: biaya transfer data dan kelas penyimpanan S3 yang dipilih. Jika Anda menginginkan kemampuan pemulihan bencana asli Amazon S3, maka Anda harus memperhitungkan bahwa Replikasi Lintas Wilayah menimbulkan biaya keluar data lintas wilayah. Untuk mempelajari selengkapnya tentang cara mengonfigurasi opsi ini, lihat Backup database SQL Server ke posting Amazon S3 di Microsoft Workloads di blog. AWS

Sumber daya tambahan

- Opsi pencadangan dan pemulihan untuk SQL Server di Amazon EC2 AWS (Panduan Preskriptif)
- Point-in-time pemulihan dan pencadangan berkelanjutan untuk Amazon RDS dengan AWS Backup (Blog AWS Penyimpanan)
- Lindungi beban kerja SQL Server Anda menggunakan NetApp SnapCenter Amazon FSx untuk NetApp ONTAP (AWS Blog Penyimpanan)
- Memulai Menggunakan Amazon S3 Intelligent-Tiering (Memulai Pusat Sumber Daya)AWS
- Strategi Backup dan Restore untuk Amazon RDS for SQL Server AWS (Blog Database)
- Memigrasi database Microsoft SQL Server lokal ke Amazon EC2 (Panduan Preskriptif)AWS
- Praktik Terbaik untuk Menyebarkan Microsoft SQL Server di Amazon AWS EC2 (Whitepaper)

Memodernisasi database SQL Server

Ikhtisar

Jika Anda memulai perjalanan menuju modernisasi database lama untuk skalabilitas, kinerja, dan pengoptimalan biaya, Anda mungkin menghadapi tantangan dengan database komersial seperti SQL Server. Database komersial mahal, mengunci pelanggan, dan menawarkan persyaratan lisensi hukuman. Bagian ini memberikan ikhtisar tingkat tinggi tentang opsi untuk bermigrasi dan memodernisasi dari SQL Server ke database sumber terbuka dan informasi tentang memilih opsi terbaik untuk beban kerja Anda.

Anda dapat memfaktorkan ulang database SQL Server Anda ke database sumber terbuka seperti Amazon Aurora PostgreSQL untuk menghemat biaya lisensi Windows dan SQL Server. Database modern cloud-native seperti Aurora menggabungkan fleksibilitas dan biaya rendah database open-source dengan fitur database komersial kelas enterprise yang kuat. Jika Anda memiliki beban kerja variabel atau beban kerja multi-tenant, Anda juga dapat bermigrasi ke Aurora tanpa server V2. Ini dapat mengurangi biaya hingga 90 persen, tergantung pada karakteristik beban kerja. Selain itu, AWS menawarkan kemampuan seperti Babelfish untuk Aurora PostgreSQL, alat seperti AWS Schema Conversion Tool (AWS SCT), dan layanan seperti AWS Database Migration Service (AWS DMS) untuk menyederhanakan migrasi dan modernisasi database SQL Server. AWS

Penawaran basis data

Bermigrasi dari SQL Server di Windows ke database open-source seperti Amazon Aurora, Amazon RDS for MySQL, atau Amazon RDS for PostgreSQL dapat menawarkan penghematan biaya yang signifikan tanpa mengorbankan kinerja atau fitur. Pertimbangkan hal berikut:

- Beralih dari edisi SQL Server Enterprise di Amazon EC2 ke Amazon RDS untuk PostgreSQL atau Amazon RDS untuk MySQL dapat menghasilkan penghematan biaya hingga 80 persen.
- Beralih dari edisi SQL Server Enterprise di Amazon EC2 ke Amazon Aurora PostgreSQL Compatible Edition atau Amazon Aurora MySQL Compatible Edition dapat menghasilkan penghematan biaya hingga 70 persen.

Untuk beban kerja database tradisional, Amazon RDS for PostgreSQL dan Amazon RDS untuk persyaratan alamat MySQL dan memberikan solusi hemat biaya untuk database relasional. Aurora menambahkan banyak ketersediaan dan fitur kinerja yang sebelumnya terbatas pada vendor komersial yang mahal. Fitur ketahanan di Aurora adalah biaya tambahan. Namun, dibandingkan

dengan fitur serupa oleh vendor komersial lainnya, biaya ketahanan Aurora masih lebih murah daripada biaya perangkat lunak komersial untuk jenis fitur yang sama. Arsitektur Aurora dioptimalkan untuk memberikan peningkatan kinerja yang signifikan dibandingkan dengan penerapan MySQL dan PostgreSQL standar.

Karena Aurora kompatibel dengan database PostgreSQL dan MySQL open-source, ada manfaat tambahan portabilitas. Apakah opsi terbaik adalah Amazon RDS untuk PostgreSQL, Amazon RDS for MySQL, atau Aurora bermuara pada pemahaman persyaratan bisnis dan memetakan fitur yang diperlukan ke opsi terbaik.

Perbandingan Amazon RDS dan Aurora

Tabel berikut merangkum perbedaan utama antara Amazon RDS dan Amazon Aurora.

Kategori	Amazon RDS untuk PostgreSQL atau Amazon RDS untuk MySQL	Aurora PostgreSQL atau Aurora MySQL
Performa	Performa bagus	3x atau kinerja yang lebih baik
Failover	Biasanya 60—120 detik*	Biasanya 30 detik
Skalabilitas	Hingga 5 replika baca Lag dalam hitungan detik	Hingga 15 replika baca Lag dalam milidetik
Penyimpanan	Hingga 64 TB	Hingga 128 TB
Penyimpanan HA	Multi-AZ dengan satu atau dua siaga, masing-masing dengan salinan database	6 salinan data di 3 Availability Zone secara default
Pencadangan	Cuplikan harian dan cadangan log	Pencadangan asinkron berkelanjutan ke Amazon S3
Inovasi dengan Aurora	TA	100 GB Kloning basis data cepat

Kategori	Amazon RDS untuk PostgreSQL atau Amazon RDS untuk MySQL	Aurora PostgreSQL atau Aurora MySQL
	Replika baca penskalaan otomatis	
	Manajemen rencana kueri	
	Aurora Serverless	
	Replika Lintas Wilayah dengan Database Global	
	Manajemen cache klaster**	
	Kueri Paralel	
	Aliran aktivitas basis data	

* Transaksi besar dapat meningkatkan waktu failover

** Tersedia di Aurora PostgreSQL

Tabel berikut menunjukkan perkiraan biaya bulanan dari berbagai layanan database yang tercakup dalam bagian ini.

Layanan basis data	Biaya USD per bulan*	AWS Kalkulator Harga (membutuhkan Akun AWS)
Amazon RDS for SQL Server edisi Enterprise	$3.750	Estimasi
Amazon RDS for SQL Server edisi Standar	$2.318	Estimasi
Edisi SQL Server Enterprise di Amazon EC2	$2.835	Estimasi

Layanan basis data	Biaya USD per bulan*	AWS Kalkulator Harga (membutuhkan Akun AWS)
SQL Server edisi Standar di Amazon EC2	$1,345	Estimasi
Amazon RDS for PostgreSQL	$742	Estimasi
Amazon RDS for MySQL	$712	Estimasi
Aurora PostgreSQL	$1.032	Estimasi
Aurora MySQL	$1.031	Estimasi

* Harga penyimpanan sudah termasuk dalam harga instans. Biaya didasarkan pada `us-east-1` Wilayah. Throughput dan IOPS adalah asumsi. Perhitungannya adalah untuk instance r6i.2xlarge dan r6g.2xlarge.

Rekomendasi optimisasi biaya

Migrasi database heterogen biasanya memerlukan konversi skema database dari sumber ke mesin database target dan migrasi data dari sumber ke database target. Langkah pertama menuju migrasi adalah mengevaluasi dan mengonversi skema server SQL dan objek kode ke mesin basis data target.

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengevaluasi dan menilai database untuk kompatibilitas dengan berbagai opsi database sumber terbuka target seperti Amazon RDS for MySQL atau Amazon RDS for PostgreSQL, Aurora MySQL, dan PostgreSQL. Anda juga dapat menggunakan alat Kompas Babelfish untuk menilai kompatibilitas dengan Babelfish untuk Aurora PostgreSQL. Hal ini membuat AWS SCT dan Compass alat yang ampuh untuk memahami pekerjaan di muka yang terlibat sebelum memutuskan strategi migrasi. Jika Anda memutuskan untuk melanjutkan, AWS SCT mengotomatiskan perubahan yang diperlukan pada skema. Filosofi inti di balik Babelfish Compass adalah memungkinkan database SQL untuk pindah ke Aurora tanpa, atau sangat sedikit, modifikasi. Kompas akan mengevaluasi database SQL yang ada untuk menentukan apakah ini dapat dicapai. Dengan cara ini, hasilnya diketahui sebelum upaya apa pun dihabiskan untuk memigrasi data dari SQL Server ke Aurora.

AWS SCT mengotomatiskan konversi dan migrasi skema database dan kode ke mesin database target. Anda dapat menggunakan Babelfish for Aurora PostgreSQL untuk memigrasikan database dan aplikasi Anda dari SQL Server ke Aurora PostgreSQL tanpa perubahan skema atau minimal. Ini dapat mempercepat migrasi Anda.

Setelah skema dimigrasikan, Anda dapat menggunakannya AWS DMS untuk memigrasikan data. AWS DMS dapat melakukan pemuatan data penuh dan mereplikasi perubahan untuk melakukan migrasi dengan waktu henti minimal.

Bagian ini mengeksplorasi alat-alat berikut secara lebih rinci:

- AWS Schema Conversion Tool
- Babelfish for Aurora PostgreSQL
- Kompas Babelfish
- AWS Database Migration Service

AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengevaluasi database SQL Server yang ada dan menilai kompatibilitas dengan Amazon RDS atau Aurora. Untuk menyederhanakan proses migrasi, Anda juga dapat menggunakan AWS SCT untuk mengonversi skema dari satu mesin database ke mesin database lainnya dalam migrasi database heterogen. Anda dapat menggunakan AWS SCT untuk mengevaluasi aplikasi Anda dan mengonversi kode aplikasi tertanam untuk aplikasi yang ditulis C #, C ++, Java, dan bahasa lainnya. Untuk informasi selengkapnya, lihat Mengonversi aplikasi SQL menggunakan AWS SCT dalam dokumentasi. AWS SCT

AWS SCT adalah AWS alat gratis yang mendukung banyak sumber database. Untuk menggunakannya AWS SCT, Anda mengarahkannya ke database sumber dan kemudian menjalankan penilaian. Kemudian, AWS SCTevaluasi skema dan menghasilkan laporan penilaian. Laporan penilaian mencakup ringkasan eksekutif, kompleksitas dan upaya migrasi, mesin basis data target yang sesuai, dan rekomendasi untuk konversi. Untuk mengunduh AWS SCT, lihat Menginstal, memverifikasi, dan memperbarui AWS SCT dalam AWS SCT dokumentasi.

Tabel berikut menunjukkan contoh Ringkasan Eksekutif yang dihasilkan oleh AWS SCT untuk menunjukkan kompleksitas yang terlibat dengan mengubah database ke platform target yang berbeda.

Platform target	Perubahan otomatis atau minimal			Tindakan kompleks			
	Objek penyimpanan	Objek kode	Tindakan konversi	Objek penyimpanan		Objek kode	
Amazon RDS for MySQL	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56
Edisi yang Kompatibel dengan Amazon Aurora MySQL	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56
Amazon RDS for PostgreSQL	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Amazon RDS for MariaDB	60 (98%)	7 (30%)	42	1 (2%)	1	16 (70%)	58

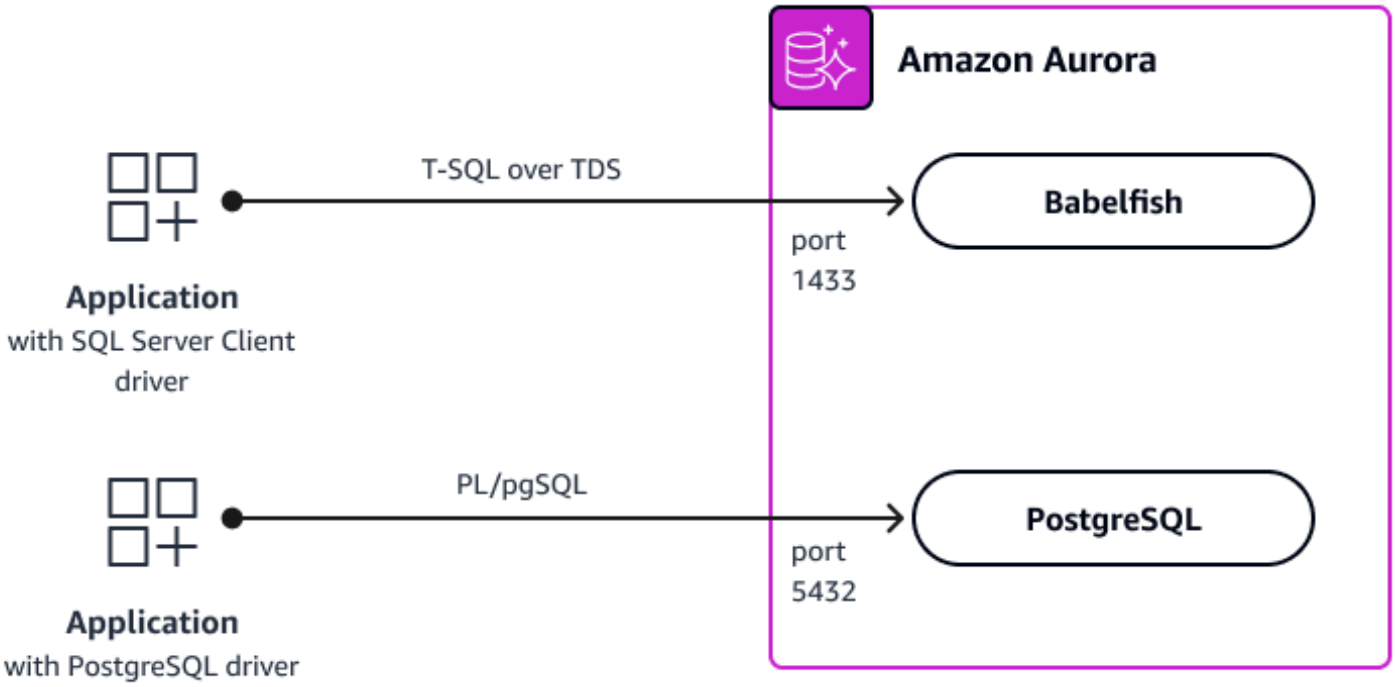
Amazon Redshift	61 (100%)	9 (39%)	124	0 (0%)	0	14 (61%)	25
AWS Glue	0 (0%)	17 (100%)	0	0 (0%)	0	0 (0%)	0
Babelfish	59 (97%)	10 (45%)	20	2 (3%)	2	12 (55%)	30

AWS SCT Laporan juga memberikan rincian tentang elemen skema yang tidak dapat dikonversi secara otomatis. Anda dapat menutup celah AWS SCT konversi dan mengoptimalkan skema target dengan merujuk ke buku pedoman AWS migrasi. Ada banyak buku pedoman migrasi database untuk membantu migrasi heterogen.

Babelfish for Aurora PostgreSQL

Babelfish untuk Aurora PostgreSQL memperluas Aurora PostgreSQL dengan kemampuan untuk menerima koneksi database dari klien SQL Server. Babelfish memungkinkan aplikasi yang awalnya dibangun untuk SQL Server untuk bekerja secara langsung dengan Aurora PostgreSQL, dengan sedikit perubahan kode dan tanpa mengubah driver database. Babelfish mengubah Aurora PostgreSQL bilingual sehingga Aurora PostgreSQL dapat bekerja dengan T-SQL dan bahasa. PL/pgSQL Babelfish meminimalkan upaya untuk bermigrasi dari SQL Server ke Aurora PostgreSQL. Ini mempercepat migrasi, meminimalkan risiko, dan mengurangi biaya migrasi secara signifikan. Anda dapat terus menggunakan migrasi posting T-SQL, tetapi ada juga opsi untuk menggunakan alat asli PostgreSQL untuk pengembangan.

Diagram berikut menggambarkan bagaimana aplikasi yang menggunakan T-SQL terhubung ke port default 1433 di SQL Server dan menggunakan penerjemah Babelfish untuk berkomunikasi dengan database Aurora PostgreSQL, sedangkan aplikasi menggunakan PL/PGSQL dapat langsung dan bersamaan terhubung ke database Aurora PostgreSQL menggunakan port default 5432 di Aurora PostgreSQL PostGresQL.



Babelfish tidak mendukung fitur SQL Server T-SQL tertentu. Untuk alasan ini, Amazon menyediakan alat penilaian untuk melakukan line-by-line analisis pernyataan SQL Anda dan menentukan apakah ada yang tidak didukung oleh Babelfish.

Ada dua opsi untuk penilaian Babelfish. AWS SCT dapat menilai kompatibilitas database SQL Server Anda dengan Babelfish. Pilihan lain adalah alat Babelfish Compass, yang merupakan solusi yang direkomendasikan karena alat Kompas diperbarui sejalan dengan rilis baru Babelfish untuk Aurora PostgreSQL.

Kompas Babelfish

Babelfish Compass adalah alat yang dapat diunduh gratis yang selaras dengan rilis terbaru Babelfish untuk Aurora PostgreSQL. Sebaliknya, AWS SCT akan mendukung versi Babelfish yang lebih baru setelah beberapa waktu. Babelfish Compass dijalankan terhadap skema database SQL Server. Anda juga dapat mengekstrak skema database SQL Server sumber dengan menggunakan alat seperti SQL Server Management Studio (SSMS). Kemudian, Anda dapat menjalankan skema melalui Babelfish Compass. Ini menghasilkan laporan yang merinci kompatibilitas skema SQL Server dengan Babelfish dan jika ada perubahan yang diperlukan sebelum bermigrasi. Alat Kompas Babelfish juga dapat mengotomatiskan banyak perubahan ini dan pada akhirnya mempercepat migrasi Anda.

Setelah penilaian dan perubahan selesai, Anda dapat memigrasikan skema ke Aurora PostgreSQL dengan menggunakan alat asli SQL Server seperti SSMS atau sqlcmd. Untuk petunjuk, lihat posting Migrasi dari SQL Server ke Amazon Aurora menggunakan Babelfish di Blog Database. AWS

AWS Database Migration Service

Setelah skema dimigrasikan, Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data AWS dengan waktu henti minimal. AWS DMS tidak hanya memuat data penuh, tetapi juga mereplikasi perubahan dari sumber ke tujuan saat sistem sumber aktif dan berjalan. Setelah database sumber dan target disinkronkan, aktivitas cutover dapat terjadi di mana aplikasi diarahkan ke database target yang menyelesaikan migrasi. AWS DMS saat ini hanya melakukan pemuatan data penuh dengan Babelfish untuk target Aurora PostgreSQL dan tidak mereplikasi perubahan. Untuk informasi selengkapnya, lihat Menggunakan Babelfish sebagai target AWS Database Migration Service dalam dokumentasi. AWS DMS

AWS DMS dapat melakukan migrasi homogen (di mesin database yang sama) dan heterogen (di seluruh mesin database yang berbeda). AWS DMS mendukung banyak mesin basis data sumber dan tujuan. Untuk informasi selengkapnya, lihat Memigrasi database SQL Server Anda ke Amazon RDS for SQL Server AWS DMS menggunakan postingan di Blog Database. AWS

Sumber daya tambahan

- Selamat tinggal Microsoft SQL Server, Halo Babelfish (Blog Berita)AWS
- Mengkonversi skema database dan aplikasi SQL menggunakan AWS Schema Conversion Tool CLI (Database Blog)AWS
- Migrasikan SQL Server ke Amazon Aurora PostgreSQL menggunakan praktik terbaik dan pelajaran yang dipetik dari lapangan (Blog Database)AWS
- Validasi objek database pasca-migrasi dari Microsoft SQL Server ke Amazon RDS for PostgreSQL dan Amazon Aurora PostgreSQL (Blog Database)AWS

Optimalkan penyimpanan untuk SQL Server

Ikhtisar

Bagian ini berfokus pada pengoptimalan biaya untuk penyimpanan SSD Amazon Elastic Block Store (Amazon EBS) untuk SQL Server pada beban kerja EC2.

Anda memiliki berbagai macam opsi penyimpanan untuk menyebarkan dan menjalankan beban kerja SQL Server. AWS Memilih penyimpanan yang tepat harus didasarkan pada tujuan, arsitektur, daya tahan, kinerja, kapasitas, dan biaya. AWS Pelanggan yang menjalankan beban kerja SQL Server biasanya menggunakan kombinasi penyimpanan Amazon EBS, Amazon, FSx dan NVMe Amazon Simple Storage Service (Amazon S3).

Amazon EBS adalah penyimpanan terpasang jaringan yang terhubung ke instans komputasi EC2 dan digunakan untuk menyimpan dan memproses sistem operasi umum, aplikasi, database, dan file cadangan. Penyimpanan solid state drive (SSD) Amazon EBS mencakup General Purpose SSD (gp2 dan gp3) dan Provisioned IOPS SSD (io1, io2, dan IO2bx). Pertimbangkan hal berikut:

- Beberapa instans EC2, seperti r5d, memiliki lokal yang melekat NVMe SSDs secara fisik ke instance host. Volume ini menyediakan penyimpanan tingkat blok yang biasa digunakan untuk SQL Server tempdb atau ekstensi kumpulan buffer.
- Amazon FSx untuk Windows File Server adalah layanan penyimpanan file yang dikelola sepenuhnya, sedangkan Amazon FSx NetApp untuk ONTAP adalah penyimpanan bersama yang dikelola sepenuhnya yang dibangun NetApp di atas sistem file ONTAP yang populer. Amazon FSx sering digunakan untuk menjalankan beban kerja SQL Server dalam konfigurasi SQL Server Failover Clustered Instance (FCI) dengan ketersediaan tinggi. Solusi ini menampung data SQL Server dan file log, yang mengurangi persyaratan kinerja EBS pada instans EC2.
- Amazon S3 adalah layanan penyimpanan objek yang menawarkan skalabilitas, ketersediaan data, keamanan, dan kinerja terdepan di industri. Anda dapat menyimpan file cadangan asli SQL Server AMIs, snapshot EBS, log aplikasi, dan lainnya di Amazon S3.

Jenis penyimpanan SSD, kinerja, dan biaya untuk Amazon EBS

Biaya penyimpanan SSD untuk Amazon EBS umumnya meningkat seiring dengan peningkatan daya tahan dan kinerja. Penyimpanan saat ini hadir dalam lima jenis volume, masing-masing dengan metrik kinerja uniknya sendiri. Untuk ringkasan kasus penggunaan dan karakteristik volume yang didukung SSD, lihat tabel di bagian volume Solid state drive (SSD) pada dokumentasi Amazon EBS.

Anda dapat menggunakan Amazon CloudWatch untuk memantau kinerja SSD, menangkap data yang sedang tren, dan mengatur alarm saat ambang batas tertentu terpenuhi. Jika Anda menjalankan beban kerja SQL Server AWS, pertimbangkan untuk mengaktifkan pemantauan terperinci dan menerapkan metrik CloudWatch khusus untuk menangkap metrik kinerja volume terperinci seperti latensi disk, IOPS, throughput, panjang antrian disk, kapasitas bekas vs. gratis, dan banyak lagi. Anda dapat menggunakan metrik CloudWatch kinerja ini untuk mengidentifikasi penyimpanan yang

kurang disediakan dan disediakan secara berlebihan serta menyediakan titik data historis untuk menentukan persyaratan penyimpanan secara akurat.

Biaya penyimpanan SSD untuk Amazon EBS juga bervariasi berdasarkan kapasitas yang dialokasikan. Tabel di bawah ini menunjukkan perbandingan jenis volume yang berbeda. Semua jenis volume memiliki kapasitas 1 TB dan konfigurasi kinerja yang serupa.

Tipe volume	IOPS Maks (16 KiB I/O)	Throughput maks (128 KiB I/O)	Harga per 1TB	Penghematan biaya persen
gp2	3.000	250	$102,40	
gp3	3.000	250	$86,92	15%
io1	16.000	500	$1.168	
io2	16.000	500	$1.168	
gp3	16.000	500	$146,92	87%
io2bx	16.000	4.000	$1.168	
gp3	16.000	1.000	$181,92	84%

> **Note**
>
> Metrik kinerja dan biaya pada tabel sebelumnya adalah per volume, berdasarkan perkiraan dari. AWS Kalkulator Harga An Akun AWS diperlukan untuk mengakses perkiraan di AWS Kalkulator Harga.

Volume Amazon EBS SSD gp3 memberikan kinerja luar biasa dengan biaya rendah. Anda dapat menghemat hingga 87 persen jika Anda memilih volume gp3 di atas volume io1 atau io2 untuk beban kerja yang membutuhkan kurang dari 16.000 IOPS dan 500 throughput. MiBps

Volume io2 Block Express (IO2bx) menawarkan peningkatan kinerja dibandingkan volume io2 biasa. Pada 16.000 IOPS, volume io1 atau io2 hanya mampu mencapai 500 MiBps throughput, sedangkan volume IO2bx dapat dikonfigurasi hingga 4.000 throughput. MiBps Dibandingkan dengan

volume io1 dan io2, volume IO2bx memberikan lebih dari empat kali throughput antara 16.000 hingga 64.000 IOPS, dengan harga yang sama persis. Volume io2 reguler dapat dikonversi ke volume IO2bx dengan melampirkannya ke instans EC2 yang didukung IO2BX. Untuk daftar instans EC2 yang didukung IO2BX, lihat Volume SSD IOPS yang disediakan dalam dokumentasi Amazon EBS. Sebelum menggunakan penyimpanan baru, Anda dapat menggunakannya AWS Kalkulator Hargauntuk memperkirakan biaya bulanan Anda dan memahami dampaknya terhadap biaya berdasarkan trade-off antara daya tahan, kinerja, dan kapasitas.

Optimalisasi biaya SSD umum untuk Amazon EBS

Kami menyarankan Anda mengevaluasi apa yang Anda simpan dan memastikan bahwa Anda menggunakan jenis dan kelas penyimpanan yang tepat. Misalnya, Amazon S3 menyediakan titik harga yang bagus, kebijakan siklus hidup bawaan, dan opsi replikasi yang ideal untuk pencadangan SQL Server. SQL Server 2022 memiliki kemampuan untuk membuat cadangan langsung ke Amazon S3, sementara versi SQL Server sebelumnya mengandalkan cadangan lokal asli. Jika Anda menjalankan SQL Server versi lama, pertimbangkan untuk membuat cadangan ke volume HDD Amazon EBS dan kemudian menyalin cadangan ke Amazon S3. Solusi ini dapat menghemat 53 persen dibandingkan dengan menggunakan volume gp3 untuk cadangan.

Tabel berikut menunjukkan perbedaan harga untuk penyimpanan 1 TB di Amazon EBS gp3, Amazon EBS HDD st1, dan Amazon S3.

Tipe penyimpanan	Kapasitas	Harga pm
EBS gp3 500 MiBps	1 TB	$96,92
EBS st1 meledak 500 MiBps		$46,08
S3 Standard		$23,55
Standar S3 (akses jarang)		$12,80
S3 Glacier Deep Archive		$1,03

> **Note**
>
> Metrik biaya pada tabel sebelumnya didasarkan pada perkiraan di. AWS Kalkulator Harga An Akun AWS diperlukan untuk mengakses perkiraan di AWS Kalkulator Harga.

Kami menyarankan Anda mempertimbangkan hal-hal berikut:

- Aktifkan pemantauan terperinci dan CloudWatch terapkan metrik khusus untuk menangkap persyaratan kinerja penyimpanan mereka secara akurat.
- Tingkatkan penyimpanan Amazon EBS dari gp2 ke gp3 untuk mengurangi biaya, meningkatkan fleksibilitas, dan meningkatkan kinerja.
- Tingkatkan penyimpanan Amazon EBS dari io1 ke io2 untuk meningkatkan daya tahan dan fleksibilitas kinerja.
- Gunakan IO2bx sebagai pengganti io1 atau io2 jika memungkinkan untuk meningkatkan daya tahan dan kinerja.
- Pertimbangkan mix-and-match pendekatan ketika memilih penyimpanan untuk membantu mengurangi kebutuhan kapasitas dan biaya volume kinerja tinggi. Misalnya, Anda dapat menggunakan volume gp3 murah untuk volume root Anda (sistem operasi), instalasi SQL Server, database sistem (tidak termasuk tempdb), dan database pengguna berkinerja rendah. Ini dapat membantu mengurangi kapasitas dan biaya volume io2, yang dapat didedikasikan untuk database pengguna berkinerja tinggi.
- Jika Anda menghosting database SQL Server AWS, kami sarankan Anda menggunakan beberapa file data SQL Server per database. Hal ini memungkinkan kesempatan untuk mendistribusikan read/write beban kerja di beberapa volume, mengurangi kinerja dan persyaratan kapasitas per volume dan akibatnya mengurangi biaya.
- Bahkan jika beban kerja produksi memerlukan penyimpanan berkinerja lebih tinggi, seperti io1 atau IO2/IO2bx, pertimbangkan volume gp3 untuk beban kerja non-produksi untuk membantu mengurangi biaya.
- Lacak dan tren pemanfaatan penyimpanan dari waktu ke waktu untuk dengan mudah mengidentifikasi lonjakan penggunaan dan biaya tak terduga.
- Gunakan AWS Compute Optimizeruntuk rekomendasi tentang penskalaan volume EBS naik atau turun berdasarkan pemanfaatan aktual.
- Gunakan elastisitas AWS untuk menyesuaikan kinerja dan kebutuhan kapasitas volume SSD Anda untuk Amazon EBS. Tidak seperti lingkungan lokal, Anda tidak perlu menyediakan kinerja dan kapasitas penyimpanan yang berlebihan untuk beban kerja di masa mendatang. Anda dapat memigrasikan beban kerja SQL Server yang ada ke AWS dan menyesuaikan kinerja atau kapasitas sesuai kebutuhan, sambil menjaga database tetap online.

Sumber daya tambahan

- Jenis volume Amazon EBS (dokumentasi Amazon EBS)
- Amazon Elastic Block Store (Amazon EBS) (dokumentasi Amazon EBS)
- Volume IOPS SSD yang disediakan (dokumentasi Amazon EBS)
- Volume penyimpanan instans SSD (dokumentasi Amazon EC2)
- CloudWatch Metrik Amazon untuk Amazon EBS (dokumentasi Amazon EBS)
- Spesifikasi untuk instans yang dioptimalkan untuk penyimpanan Amazon EC2 (dokumentasi Amazon EC2)
- Lindungi beban kerja SQL Server Anda menggunakan NetApp SnapCenter Amazon FSx untuk NetApp ONTAP (AWS Blog Penyimpanan)
- FAQ Amazon EC2 (halaman produk)AWS

Optimalkan lisensi SQL Server dengan menggunakan Compute Optimizer

Panduan tentang cara mengoptimalkan lisensi untuk SQL Server dengan menggunakan. AWS Compute Optimizer

Ikhtisar

AWS Compute Optimizerdapat merekomendasikan peluang pengoptimalan lisensi untuk beban kerja Microsoft SQL Server di Amazon Elastic Compute Cloud (Amazon EC2). Compute Optimizer dapat memberikan rekomendasi otomatis untuk mengurangi biaya lisensi. Rekomendasi dari Compute Optimizer tercantum di samping setiap instans EC2 Anda dengan lisensi Microsoft SQL Server. Informasi yang diberikan mencakup peluang penghematan yang disarankan, harga On-Demand instans EC2, dan harga lisensi Anda sendiri (BYOL) per jam. Informasi ini dapat membantu Anda memutuskan apakah Anda harus menurunkan versi edisi lisensi Anda.

Compute Optimizer secara otomatis menemukan instans SQL Server Anda di Amazon EC2 berdasarkan jenis beban kerja yang disimpulkan. Untuk melihat rekomendasi lisensi, Anda dapat memilih instance SQL Server di Compute Optimizer dan kemudian mengautentikasi dengan CloudWatch Amazon Application Insights menggunakan kredenal database hanya-baca. Compute Optimizer menganalisis jika Anda menggunakan fitur edisi SQL Server Enterprise apa pun. Jika

tidak ada fitur edisi Enterprise yang digunakan, Compute Optimizer merekomendasikan agar Anda menurunkan versi ke edisi Standar untuk mengurangi biaya lisensi.

Anda juga dapat menggunakan Compute Optimizer untuk membuat rekomendasi ukuran untuk instans Amazon EC2 yang menjalankan beban kerja SQL Server. Untuk informasi selengkapnya, lihat Optimalkan ukuran SQL Server menggunakan Compute Optimizer dalam panduan ini.

Rekomendasi optimisasi biaya

Rekomendasi lisensi di Compute Optimizer dapat membantu Anda mengevaluasi fitur yang Anda gunakan di Microsoft SQL Server dan memilih edisi yang paling hemat biaya untuk beban kerja Anda. Edisi SQL Server Enterprise secara signifikan lebih mahal daripada edisi Standar. Untuk informasi selengkapnya, lihat Bandingkan edisi SQL Server dalam panduan ini dan lihat harga SQL Server 2022 di situs web Microsoft. Menginvestasikan waktu untuk mengonfigurasi Compute Optimizer untuk mengevaluasi armada SQL Server Anda dan memberikan rekomendasi dapat secara dramatis mengurangi biaya lisensi Anda.

Halaman detail Lisensi memberikan informasi berikut:

- Gunakan tabel untuk membandingkan setelan lisensi Anda saat ini (seperti edisi, model, dan jumlah inti instans) dengan rekomendasi Compute Optimizer.
- Gunakan grafik pemanfaatan untuk meninjau jumlah fitur edisi Enterprise yang digunakan selama periode analisis.

Untuk informasi selengkapnya, lihat Melihat detail rekomendasi lisensi perangkat lunak komersial di dokumentasi Compute Optimizer.

Konfigurasikan Compute Optimizer

Compute Optimizer menganalisis lisensi perangkat lunak komersial dengan menggunakan metrik. `mssql_enterprise_features_used` Untuk informasi selengkapnya tentang metrik ini, lihat Metrik untuk lisensi perangkat lunak komersial.

1. Pastikan Anda memiliki izin yang sesuai untuk ikut serta dalam Compute Optimizer. Untuk informasi selengkapnya, lihat berikut ini:
   - Kebijakan untuk ikut serta dalam Compute Optimizer
   - Kebijakan untuk memberikan akses ke Compute Optimizer untuk mandiri Akun AWS
   - Kebijakan untuk memberikan akses ke Compute Optimizer untuk akun manajemen organisasi

2. Lampirkan peran dan kebijakan instans yang diperlukan untuk CloudWatch Application Insights. Untuk petunjuk, lihat Kebijakan untuk mengaktifkan rekomendasi lisensi perangkat lunak komersial.
3. Aktifkan Wawasan CloudWatch Aplikasi dengan menggunakan kredensil database Microsoft SQL Server Anda. Untuk petunjuk, lihat Mengatur aplikasi untuk pemantauan dalam CloudWatch dokumentasi.

> **Note**
>
> Untuk menghasilkan rekomendasi untuk lisensi perangkat lunak komersial, setidaknya 30 jam berturut-turut data CloudWatch metrik diperlukan. Untuk informasi selengkapnya, lihat persyaratan CloudWatch metrik.

4. Gunakan kueri SQL berikut untuk mengonfigurasi akses hak istimewa terkecil untuk Wawasan Aplikasi. CloudWatch

```
GRANT VIEW SERVER STATE TO [LOGIN];
GRANT VIEW ANY DEFINITION TO [LOGIN];
```

   Ini memungkinkan layanan baru, PrometheusSqlExporter SQL.

5. Dari akun manajemen target Akun AWS atau organisasi, pilih Compute Optimizer. Untuk petunjuk, lihat Memilih di akun Anda.

> **Note**
>
> Setelah Anda ikut serta, temuan dan rekomendasi pengoptimalan dapat memakan waktu hingga 24 jam untuk dihasilkan.

6. Di konsol Compute Optimizer, pilih Lisensi di panel navigasi.
7. Di kolom Temuan, cari instance yang memiliki temuan metrik tidak mencukupi. Compute Optimizer mengembalikan temuan ini jika mendeteksi CloudWatch bahwa Application Insights tidak diaktifkan atau memiliki izin yang tidak mencukupi. Untuk informasi lebih lanjut, lihat Menemukan alasan. Lakukan hal berikut untuk menyelesaikan temuan ini:
   a. Pilih instance.
   b. Tambahkan rahasia.
   c. Konfirmasikan peran instans dan kebijakan terlampir.

d. Pilih Aktifkan rekomendasi lisensi.

8. Di kolom Temuan, cari contoh apa pun yang memiliki temuan Tidak dioptimalkan. Compute Optimizer mengembalikan temuan ini jika mendeteksi bahwa infrastruktur Amazon EC2 Anda tidak menggunakan salah satu fitur lisensi Microsoft SQL Server yang Anda bayar. Untuk informasi lebih lanjut, lihat Menemukan alasan. Lakukan hal berikut untuk menyelesaikan temuan ini:
   a. Pilih instance.
   b. Bandingkan edisi lisensi saat ini dengan edisi yang direkomendasikan.
   c. Tinjau grafik pemanfaatan lisensi saat ini.
   d. Jika Anda ingin menurunkan versi lisensi, pilih Implementasikan rekomendasi.
   e. Tinjau persyaratan dan ikuti instruksi untuk menurunkan versi lisensi. Jika Anda ingin mengotomatiskan proses, lihat Downgrade SQL Server Enterprise edition menggunakan AWS Systems Manager Dokumen untuk mengurangi biaya (AWS Blog).

Sumber daya tambahan

- Mengurangi biaya lisensi Microsoft SQL Server dengan AWS Compute Optimizer(Blog)AWS
- Apa itu AWS Compute Optimizer? (AWS dokumentasi)
- Melihat rekomendasi lisensi perangkat lunak komersial (AWS dokumentasi)
- Turunkan versi edisi Microsoft SQL Server Anda (dokumentasi)AWS
- Microsoft SQL Server di AWS ()AWS
- Lisensi Microsoft pada AWS ()AWS
- Harga Microsoft SQL Server 2019 (Microsoft)
- Harga Microsoft SQL Server 2022 (Microsoft)

Optimalkan ukuran SQL Server dengan menggunakan Compute Optimizer

Ikhtisar

AWS Compute Optimizermembantu administrator database (DBA) menemukan beban kerja Microsoft SQL Server di Amazon Elastic Compute Cloud (Amazon EC2) dan instans EC2 ukuran kanan untuk mengurangi biaya lisensi hingga 25%. Fitur tipe beban kerja yang disimpulkan di Compute Optimizer menggunakan machine learning (ML) dan secara otomatis mendeteksi aplikasi yang

mungkin berjalan pada resource Anda. AWS Compute Optimizer mencakup dukungan untuk SQL Server sebagai tipe beban kerja yang disimpulkan. Dengan menggunakan fitur tipe beban kerja yang disimpulkan, Anda dapat menentukan peluang penghematan biaya berdasarkan beban kerja tertentu yang berjalan pada instans Amazon EC2 Anda.

Dengan fitur ini, Anda dapat mengkategorikan peluang penghematan biaya dengan jenis beban kerja yang disimpulkan yang didukung, seperti SQL Server. Compute Optimizer dapat secara otomatis menemukan instans SQL Server EC2 yang disediakan secara berlebihan. Anda dapat beralih ke konsol EC2 untuk mengurangi ukuran instans, yang membantu mengurangi biaya lisensi dan infrastruktur.

Anda juga dapat menggunakan Compute Optimizer untuk membuat rekomendasi lisensi SQL Server. Untuk informasi selengkapnya, lihat Optimalkan lisensi SQL Server menggunakan Compute Optimizer dalam panduan ini.

Konfigurasikan Compute Optimizer

Untuk petunjuk penggunaan Compute Optimizer dengan beban kerja yang disimpulkan SQL Server, lihat Mengoptimalkan kinerja dan mengurangi biaya lisensi: Memanfaatkan instans SQL Server Amazon EC2 (AWS Compute Optimizer Blog).AWS Anda dapat memilih akun mandiri, akun yang merupakan anggota organisasi, dan akun manajemen organisasi. Untuk akun mandiri dan anggota, memilih untuk mengaktifkan Compute Optimizer hanya untuk akun tersebut. Untuk akun manajemen organisasi, Anda dapat memilih apakah akan mengaktifkan Compute Optimizer di akun tersebut saja atau untuk semua akun anggota organisasi.

Proses keikutsertaan Compute Optimizer secara otomatis membuat AWS Identity and Access Management peran terkait layanan (IAM). Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk. AWS Compute Optimizer

Compute Optimizer menganalisis sumber daya Anda berdasarkan metrik CloudWatch Amazon, seperti penggunaan CPU, I/O, jaringan, dan Amazon Elastic Block Store (Amazon EBS). Untuk menghasilkan rekomendasi, setidaknya 30 jam berturut-turut data CloudWatch metrik diperlukan dalam 14 hari terakhir. Jika Anda mengaktifkan fitur metrik infrastruktur yang disempurnakan, itu memperluas metrik pemanfaatan hingga 93 hari. Untuk informasi selengkapnya, lihat persyaratan CloudWatch metrik dan Metrik infrastruktur yang disempurnakan dalam dokumentasi Compute Optimizer.

Compute Optimizer menyediakan opsi dan penghematan yang terkait dengan setiap opsi, berdasarkan vCPU, memori, penyimpanan, jaringan, risiko, dan upaya migrasi. Anda dapat

menggunakan dasbor CloudWatch metrik untuk menganalisis data yang digunakan untuk membuat rekomendasi. Dengan data ini, Anda dapat mengukur instans EC2 yang menjalankan beban kerja SQL Server. Untuk informasi selengkapnya tentang cara mengubah jenis instans, lihat Mengubah jenis instans di dokumentasi Amazon EC2.

Sumber daya tambahan

- AWS Compute Optimizer mengidentifikasi dan menyaring beban kerja Microsoft SQL Server ()AWS
- Mengoptimalkan kinerja dan mengurangi biaya lisensi: Memanfaatkan instans AWS Compute Optimizer SQL Server Amazon EC2 (Blog)AWS
- Apa itu AWS Compute Optimizer? (AWS dokumentasi)
- Melihat rekomendasi instans EC2 (AWS dokumentasi)

Meninjau Trusted Advisor rekomendasi untuk beban kerja SQL Server

Ikhtisar

AWS Trusted Advisormemberikan rekomendasi yang membantu Anda mengikuti praktik AWS terbaik. Dengan menganalisis penggunaan, konfigurasi, dan pengeluaran Anda, Trusted Advisor memberikan rekomendasi yang dapat ditindaklanjuti untuk mengurangi biaya, meningkatkan ketersediaan dan kinerja sistem, atau membantu menutup celah keamanan. Bagian ini berfokus pada Trusted Advisor pemeriksaan yang dapat membantu Anda mengurangi biaya pengoperasian beban kerja SQL Server di. AWS Cloud

Rekomendasi optimisasi biaya

Trusted Advisor memberikan rekomendasi yang membantu Anda mengoptimalkan beban kerja SQL Server di Amazon Elastic Compute Cloud (Amazon EC2). Pemeriksaan memeriksa beban kerja SQL Server Anda dan secara otomatis mencantumkan instance yang memerlukan pengoptimalan. Trusted Advisor Rekomendasi operasionalisasi dapat mengurangi biaya dan meningkatkan postur keamanan organisasi Anda.

Berikut ini adalah Trusted Advisor pemeriksaan yang berfokus pada Microsoft SQL Server:

- Instans Amazon EC2 disediakan secara berlebihan untuk Microsoft SQL Server — Pemeriksaan ini menganalisis instans Amazon EC2 Anda yang menjalankan SQL Server dan memberi tahu Anda jika instans melebihi batas vCPU perangkat lunak SQL Server. Misalnya, sebuah instance dengan SQL Server Standard edition dapat menggunakan hingga 48 vCPUs. Sebuah instance dengan SQL Server Web dapat menggunakan hingga 32 vCPUs.

Edisi	vCPU Min.	vCPU Maks.
Web	4	32
Standar	4	48
Perusahaan	4	Batas OS

- Konsolidasi instans Amazon EC2 untuk Microsoft SQL Server — Pemeriksaan ini menganalisis instans Amazon EC2 Anda dan memberi tahu Anda jika instans Anda memiliki kurang dari jumlah minimum lisensi SQL Server. Anda dapat mengkonsolidasikan instance SQL Server yang lebih kecil untuk membantu menurunkan biaya. Jika Anda memiliki banyak instance SQL Server kecil yang disertakan lisensi, maka pertimbangkan untuk mengkonsolidasikan. Menurut panduan lisensi Microsoft SQL Server 2019, SQL Server memerlukan minimal 4 lisensi vCPU per instans. Jika Anda mengkonsolidasikan database ini, Anda dapat menghemat biaya lisensi. Anda dapat membuat keputusan berdasarkan jumlah database pada instance, ukuran database maksimum, dan ukuran total database. Konsolidasi didukung untuk edisi Web, Standar, dan Perusahaan SQL Server. Untuk informasi selengkapnya, lihat Mengkonsolidasikan Database SQL Server (posting blog Microsoft).

  AWS tidak merekomendasikan menempatkan database produksi besar hanya pada satu server. Namun, Anda dapat mengkonsolidasikan yang lebih kecil yang digunakan untuk lingkungan non-produksi, seperti untuk pengembangan, pengujian, dan pementasan. Ini tergantung pada penggunaan SQL Server Anda saat ini; jika Anda memiliki database penggunaan rendah, Anda dapat mengkonsolidasikan pada satu server.

Konfigurasikan Trusted Advisor

Lakukan hal berikut untuk mengevaluasi pemeriksaan terfokus SQL Server. Trusted Advisor

1. Masuk ke Konsol Manajemen AWS.
2. Buka konsol AWS Trusted Advisor.

3. Di panel navigasi, di bawah Rekomendasi, pilih Optimalisasi biaya.
4. Dalam daftar pemeriksaan pengoptimalan biaya, tinjau status konsolidasi instans Amazon EC2 untuk instans Microsoft SQL Server dan Amazon EC2 yang disediakan secara berlebihan untuk pemeriksaan Microsoft SQL Server.
   - Simbol centang hijau menunjukkan bahwa instans Amazon EC2 Anda dikonfigurasi secara optimal.
   - Simbol peringatan oranye menunjukkan bahwa ada peluang untuk perbaikan.
5. Pilih cek untuk melihat detail dan rekomendasinya.
6. Ikuti petunjuk yang diberikan oleh pemeriksaan untuk mengoptimalkan instans Amazon EC2 yang menjalankan beban kerja SQL Server.
7. Pantau instans Anda secara teratur, dan segarkan pemeriksaan secara berkala.

Sumber daya tambahan

- Trusted Advisor periksa referensi (AWS dokumentasi)
- Microsoft SQL Server di AWS ()AWS
- Lisensi Microsoft pada AWS ()AWS
- Harga SQL Server 2019 (Microsoft)
- AWS Launch Wizard untuk SQL Server (AWS dokumentasi)

Wadah

Modernisasi adalah perjalanan transformasional yang menawarkan banyak opsi, termasuk menguraikan monolit ke layanan mikro, merancang ulang aplikasi agar didorong oleh peristiwa dengan menggunakan fungsi tanpa server (),AWS Lambda dan repurposing database dari SQL Server ke Amazon Aurora atau database terkelola yang dibuat khusus. Jalur modernisasi untuk memplatform ulang aplikasi.NET Framework ke wadah Linux dan Windows membutuhkan lebih sedikit usaha daripada opsi modernisasi lainnya. Wadah menawarkan manfaat berikut:

- Mempercepat inovasi — Pindah ke kontainer memudahkan untuk mengotomatisasi tahapan siklus hidup pengembangan yang mencakup pembuatan, pengujian, dan penerapan aplikasi. Dengan mengotomatiskan proses ini, tim pengembangan dan operasi memiliki lebih banyak waktu untuk fokus pada inovasi.
- Mengurangi total biaya kepemilikan (TCO) — Pindah ke kontainer juga dapat mengurangi ketergantungan Anda pada manajemen lisensi dan alat perlindungan titik akhir. Karena kontainer adalah unit komputasi sementara, Anda dapat mengotomatiskan dan menyederhanakan tugas manajemen seperti menambal, menskalakan, dan mencadangkan dan memulihkan. Ini mengurangi TCO dalam mengelola dan mengoperasikan beban kerja berbasis kontainer. Terakhir, kontainer lebih efisien dibandingkan dengan mesin virtual karena Anda dapat menggunakan wadah untuk memaksimalkan penempatan aplikasi Anda dengan memberikan isolasi yang lebih baik. Ini meningkatkan pemanfaatan sumber daya infrastruktur aplikasi Anda.
- Tingkatkan pemanfaatan sumber daya — Kontainer lebih efisien dibandingkan dengan mesin virtual karena Anda dapat menggunakan wadah untuk memaksimalkan penempatan aplikasi Anda. Ini meningkatkan pemanfaatan sumber daya infrastruktur aplikasi Anda dengan memberikan isolasi yang lebih baik.
- Tutup kesenjangan keterampilan — AWS menawarkan hari perendaman untuk meningkatkan keterampilan tim pengembangan Anda tentang teknologi dan DevOps praktik kontainer.

Bagian ini mencakup topik-topik berikut:

- Pindahkan aplikasi Windows ke wadah
- Optimalkan biaya untuk AWS Fargate tugas di Amazon ECS
- Dapatkan visibilitas ke biaya Amazon EKS Anda
- Replatform aplikasi Windows dengan App2Container

Untuk informasi perizinan, lihat bagian LisensiAmazon Web Services dan Microsoft: Pertanyaan yang Sering Diajukan atau email pertanyaan Anda ke microsoft@amazon.com.

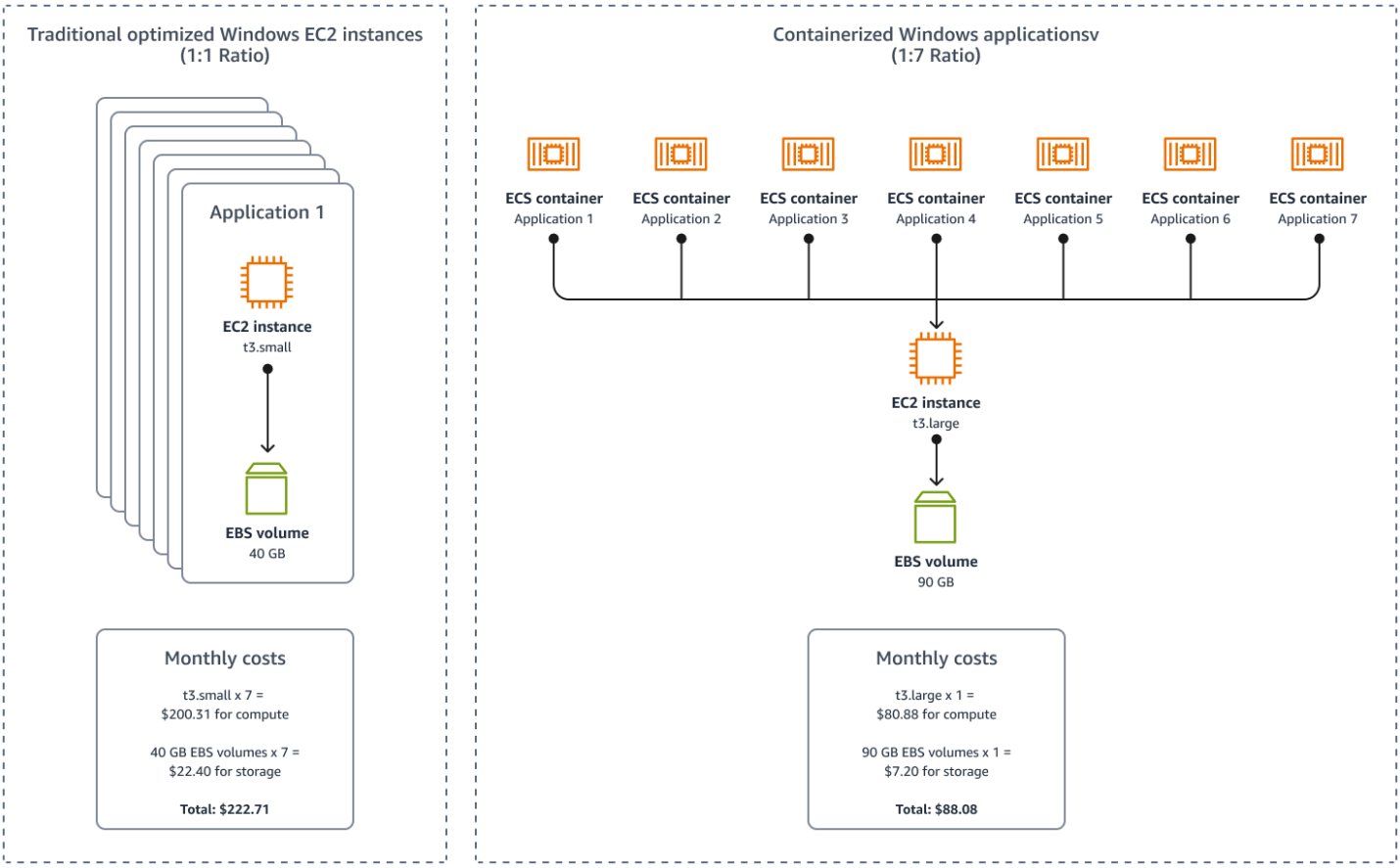
Pindahkan aplikasi Windows ke wadah

Ikhtisar

Menurut Survei Tahunan CNCF 2021, 96 persen organisasi menggunakan atau mengevaluasi wadah untuk memodernisasi infrastruktur mereka. Ini karena kontainer dapat membantu organisasi Anda mengurangi risiko, meningkatkan efisiensi dan kecepatan operasional, dan memungkinkan kelincahan. Anda juga dapat menggunakan kontainer untuk mengurangi biaya menjalankan aplikasi Anda. Bagian ini menawarkan rekomendasi untuk menjalankan kontainer yang hemat biaya di seluruh layanan AWS kontainer, termasuk Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), dan. AWS Fargate

Manfaat biaya

Infografis berikut menunjukkan penghematan biaya yang dapat dicapai bisnis dengan mengkonsolidasikan aplikasi ASP.NET Framework mereka ke instans Amazon Elastic Compute Cloud (Amazon EC2) berdasarkan AWS rekomendasi Optimization and Licensing Assessment (OLA).AWS Infografis berikut menunjukkan penghematan tambahan apa yang dapat dicapai dengan memindahkan aplikasi ke wadah Windows.



AWS OLA merekomendasikan agar bisnis melakukan lift dan beralih ke instans t3.small individual. Bisnis dapat mencapai penghematan ini dengan menjalankan tujuh aplikasi ASP.NET di server lokal, seperti yang ditunjukkan oleh analisis pemanfaatan kinerja berikut.

		Not sharable ↓ Storage	Nearing end of life ↓ Operating system	Approximately 15% overhead ↓ On-premises CPU AVG utilization	On-premises CPU peak utilization	The OS accounts for 1.2 GB per instance ↓ On-premises RAM (GB)	On-premises RAM AVG utilization (GB)	On-premises RAM peak utilization (GB)	Optimized for EC2 ↓ Instance size	vCPU	RAM (GB)
1	AppServer01	60	Windows Server 2012	7.00%	17.00%	8	13.50%	17.10%	t3.small	2	2
2	AppServer02	39	Windows Server 2012	20.07%	22.00%	16	7.50%	12.40%	t3.small	2	2
3	AppServer03	39	Windows Server 2012	24.00%	25.50%	16	8.80%	11.90%	t3.small	2	2
4	AppServer04	4	Windows Server 2012	21.40%	24.00%	16	7.80%	10.70%	t3.small	2	2
5	AppServer05	40	Windows Server 2012	21.30%	23.00%	16	8.20%	12.00%	t3.small	2	2
6	AppServer06	39	Windows Server 2012	21.50%	23.50%	16	7.90%	10.90%	t3.small	2	2
7	AppServer07	39	Windows Server 2012	21.60%	22.90%	16	8.40%	11.50%	t3.small	2	2

Analisis lebih lanjut mengungkapkan bahwa bisnis dapat menghemat lebih banyak biaya dengan menjalankan beban kerjanya pada kontainer. Container mengurangi overhead sistem operasi pada sumber daya sistem seperti CPU, RAM, dan penggunaan disk (dijelaskan di bagian selanjutnya). Dalam skenario ini, bisnis dapat mengkonsolidasikan ketujuh aplikasi ke dalam satu instance t3.large dan masih memiliki RAM 3 GB untuk cadangan. Migrasi ke kontainer dapat membantu bisnis

mencapai penghematan biaya rata-rata 64 persen di seluruh komputasi dan penyimpanan dengan menggunakan kontainer alih-alih Amazon EC2.

Rekomendasi optimisasi biaya

Bagian berikut menawarkan rekomendasi untuk mengoptimalkan biaya dengan mengkonsolidasikan aplikasi dan menggunakan wadah.

Kurangi jejak Windows Anda di Amazon EC2

Kontainer Windows dapat mengurangi jejak Windows Anda di Amazon EC2 dengan memungkinkan Anda untuk mengkonsolidasikan lebih banyak aplikasi ke instans EC2 yang lebih sedikit. Misalnya, asumsikan bahwa Anda memiliki 500 aplikasi ASP.NET. Jika Anda menjalankan satu inti per satu aplikasi untuk Windows di Amazon EC2, itu sama dengan 500 instance Windows (t3.small). Jika Anda mengasumsikan rasio 1:7 (yang dapat meningkat secara signifikan tergantung pada jenis/ukuran instans EC2) untuk menggunakan wadah Windows (dengan t3.large), maka Anda hanya memerlukan sekitar 71 instance Windows. Itu mewakili penurunan 85,8 persen pada jejak Windows Anda di Amazon EC2.

Mengurangi biaya lisensi Windows

Jika Anda melisensikan instance Windows, Anda tidak perlu melisensikan kontainer yang berjalan pada instance itu. Akibatnya, mengkonsolidasikan aplikasi ASP.NET Anda menggunakan wadah Windows dapat secara signifikan mengurangi biaya lisensi Windows Anda.

Kurangi jejak penyimpanan Anda

Setiap kali Anda meluncurkan instans EC2 baru, Anda membuat dan membayar volume Amazon Elastic Block Store (Amazon EBS) baru untuk menampung sistem operasi. Saat skala ini, biaya berskala dengannya. Jika Anda menggunakan kontainer, Anda dapat mengurangi biaya penyimpanan karena semua kontainer berbagi sistem operasi dasar yang sama. Selain itu, container menggunakan konsep layer untuk menggunakan kembali bagian immutable dari image container untuk semua container yang berjalan berdasarkan image tersebut. Dalam skenario contoh sebelumnya, semua kontainer menjalankan .NET Framework dan oleh karena itu semua berbagi lapisan kerangka ASP.NET menengah dan abadi.

Migrasikan end-of-support server ke kontainer

Support untuk Windows Server 2012 dan Windows Server 2012 R2 berakhir pada 10 Oktober 2023. Anda dapat memigrasikan aplikasi yang berjalan di Windows Server 2012 atau versi sebelumnya

dengan memasukkannya ke dalam wadah untuk berjalan di sistem operasi baru. Dengan cara ini, Anda menghindari menjalankan aplikasi pada sistem operasi yang tidak sesuai sambil memanfaatkan efisiensi biaya, pengurangan risiko, efisiensi operasional, kecepatan, dan kelincahan yang disediakan kontainer.

Peringatan yang perlu dipertimbangkan dengan pendekatan ini adalah jika aplikasi Anda memerlukan spesifik APIs terkait dengan versi sistem operasi yang saat ini digunakan (COM Interop, misalnya). Dalam hal ini, Anda harus menguji memindahkan aplikasi Anda ke versi Windows yang lebih baru. Wadah Windows menyelaraskan gambar wadah dasarnya (misalnya, Windows Server 2019) dengan sistem operasi host kontainer (misalnya, Windows Server 2019). Menguji dan pindah ke kontainer dapat memungkinkan peningkatan sistem operasi yang lebih mudah di masa depan dengan mengubah gambar dasar dalam Dockerfile Anda dan menyebarkan ke set host baru yang menjalankan versi terbaru Windows.

Hapus alat dan lisensi manajemen pihak ketiga

Mengelola armada server Anda memerlukan penggunaan beberapa alat operasi sistem pihak ketiga untuk patching dan manajemen konfigurasi. Ini dapat membuat manajemen infrastruktur menjadi kompleks dan Anda sering dikenakan biaya lisensi pihak ketiga. Jika Anda menggunakan kontainer aktif AWS, Anda tidak perlu mengelola apa pun di sisi sistem operasi. Runtime kontainer mengelola kontainer. Ini berarti inang yang mendasarinya bersifat fana dan dapat dengan mudah diganti. Anda dapat menjalankan kontainer Anda tanpa perlu mengelola host kontainer secara langsung. Selain itu, Anda dapat menggunakan alat gratis seperti AWS Systems Manager Session Manager mengakses host dengan mudah dan memecahkan masalah.

Meningkatkan kontrol dan portabilitas

Container memberi Anda kontrol yang lebih terperinci atas sumber daya server seperti CPU dan RAM daripada yang Anda miliki atas instans EC2. Untuk instans EC2, Anda dapat mengontrol CPU dan RAM dengan memilih keluarga instans, jenis instans, dan opsi CPU. Namun, dengan kontainer, Anda dapat menentukan dengan tepat berapa banyak CPU atau RAM yang ingin Anda alokasikan ke wadah dalam definisi tugas ECS Anda atau ke pod di Amazon EKS. Bahkan, kami merekomendasikan untuk menentukan CPU dan memori tingkat kontainer untuk wadah Windows. Tingkat granularitas ini membawa manfaat biaya. Perhatikan contoh kode berikut:

```
json
{
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/demo-
service:1",
```

```
    "containerDefinitions": [
        {
            "name": "demo-service",
            "image": "mcr.microsoft.com/dotnet/framework/samples:aspnetapp-
windowsservercore-ltsc2019",
            "cpu": 512,
            "memory": 512,
            "links": [],
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 0,
                    "protocol": "tcp"
                }
            ],
```

Mempercepat inovasi

Pindah ke kontainer memudahkan untuk mengotomatisasi tahapan siklus hidup pengembangan yang mencakup pembuatan, pengujian, dan penerapan aplikasi. Jika Anda mengotomatiskan proses ini, maka Anda memberi tim pengembangan dan operasi Anda lebih banyak waktu untuk fokus berinovasi.

Kurangi TCO

Pindah ke kontainer sering mengurangi ketergantungan pada manajemen lisensi dan alat perlindungan titik akhir. Karena kontainer adalah unit komputasi sementara, Anda dapat mengotomatiskan dan menyederhanakan tugas manajemen seperti menambal, menskalakan, dan mencadangkan dan memulihkan. Ini dapat mengurangi TCO dalam mengelola dan mengoperasikan beban kerja berbasis kontainer. Container lebih efisien dibandingkan dengan mesin virtual karena memungkinkan Anda memaksimalkan penempatan aplikasi Anda sehingga Anda dapat meningkatkan pemanfaatan sumber daya infrastruktur aplikasi Anda.

Tutup kesenjangan keterampilan

AWS menawarkan program dan hari perendaman untuk meningkatkan keterampilan tim pengembangan pelanggan pada wadah dan DevOps teknologi. Ini termasuk konsultasi langsung dan pemberdayaan.

Refactor ke .NET 5+ dan gunakan wadah Linux

Meskipun Anda dapat mengurangi biaya dengan memindahkan aplikasi.NET Framework Anda ke container, Anda dapat mewujudkan penghematan biaya lebih lanjut ketika Anda memfaktorkan ulang aplikasi.NET lama ke alternatif cloud-native. AWS

Hapus biaya lisensi

Refactoring aplikasi Anda dari .NET Framework di Windows ke .NET Core di Linux menghasilkan penghematan biaya sekitar 45 persen.

Akses perangkat tambahan terbaru

Refactoring aplikasi Anda dari .NET Framework di Windows ke .NET Core di Linux memberi Anda akses ke perangkat tambahan terbaru seperti Graviton2. Graviton2 menawarkan harga 40 persen lebih baik untuk kinerja dibandingkan contoh yang sebanding.

Meningkatkan keamanan dan kinerja

Memfaktorkan ulang aplikasi Anda dari .NET Framework di Windows ke .NET Core pada wadah Linux membawa peningkatan keamanan dan kinerja. Ini karena Anda mendapatkan patch keamanan terbaru, manfaat dari isolasi kontainer, dan memiliki akses ke fitur-fitur baru.

Gunakan wadah Windows alih-alih menjalankan banyak aplikasi pada satu contoh IIS

Pertimbangkan keuntungan berikut menggunakan wadah Windows alih-alih menjalankan beberapa aplikasi pada satu instance EC2 Windows dengan Internet Information Services (IIS):

- Keamanan — Kontainer memberikan tingkat keamanan di luar kotak yang tidak dicapai melalui isolasi di tingkat IIS. Jika satu situs web atau aplikasi IIS dikompromikan, semua situs yang dihosting lainnya terbuka dan rentan. Pelarian kontainer jarang terjadi dan kerentanan yang lebih sulit untuk dieksploitasi daripada mendapatkan kendali server melalui kerentanan web.
- Fleksibilitas — Kemampuan untuk menjalankan kontainer dalam isolasi proses dan memiliki instance sendiri memungkinkan opsi jaringan yang lebih terperinci. Container juga menawarkan metode distribusi yang kompleks di banyak instans EC2. Anda tidak mendapatkan manfaat ini ketika Anda mengkonsolidasikan aplikasi pada satu instance IIS.
- Manajemen overhead — Server Name Indication (SNI) menciptakan overhead yang membutuhkan manajemen dan otomatisasi. Selain itu, Anda harus bergulat dengan operasi manajemen sistem

operasi yang khas seperti menambal, memecahkan masalah BSOD (jika penskalaan otomatis tidak ada), perlindungan titik akhir, dan sebagainya. Mengkonfigurasi situs IIS sesuai dengan praktik terbaik keamanan adalah aktivitas yang memakan waktu dan berkelanjutan. Anda bahkan mungkin perlu mengatur tingkat kepercayaan, yang juga menambah overhead manajemen. Wadah dirancang untuk menjadi tanpa kewarganegaraan dan tidak dapat diubah. Pada akhirnya, penerapan Anda lebih cepat, lebih aman, dan dapat diulang jika Anda menggunakan wadah Windows sebagai gantinya.

Langkah selanjutnya

Berinvestasi dalam infrastruktur modern untuk menjalankan beban kerja lama Anda membawa manfaat besar bagi organisasi Anda. AWS Layanan kontainer memudahkan pengelolaan infrastruktur dasar Anda, baik di tempat maupun di cloud, sehingga Anda dapat fokus pada inovasi dan kebutuhan bisnis Anda. Hampir 80 persen dari semua kontainer di cloud berjalan AWS hari ini. AWS menyediakan serangkaian layanan kontainer yang kaya untuk hampir semua kasus penggunaan. Untuk memulai, lihat Kontainer di AWS.

Sumber daya tambahan

- Optimalkan biaya untuk beban kerja kontainer dengan penyedia kapasitas ECS dan Instans Spot EC2 (Blog)AWS
- Daftar Periksa Pengoptimalan Biaya untuk Amazon ECS dan AWS Fargate(Blog)AWS
- Amazon EKS di AWS Graviton2 umumnya tersedia: pertimbangan pada aplikasi multi-arsitektur (Blog)AWS
- Optimalisasi biaya untuk Kubernetes di AWS(Blog)AWS
- Mengoptimalkan biaya komputasi Kubernetes Anda dengan konsolidasi Karpenter (Blog)AWS

Optimalkan biaya untuk AWS Fargate tugas di Amazon ECS

Ikhtisar

AWS Fargate Tugas ukuran yang tepat merupakan langkah penting untuk optimalisasi biaya. Terlalu sering, aplikasi dibangun dengan ukuran sewenang-wenang untuk tugas Fargate dan tidak pernah ditinjau kembali. Hal ini dapat menyebabkan overprovisioning tugas Fargate dan pengeluaran yang tidak perlu. Bagian ini menunjukkan cara menggunakannya AWS Compute Optimizeruntuk

memberikan rekomendasi yang dapat ditindaklanjuti sehingga Anda dapat mengoptimalkan CPU tugas dan memori untuk layanan Amazon Elastic Container Service (Amazon ECS) yang berjalan di Fargate. Compute Optimizer juga mengukur dampak biaya dari mengadopsi rekomendasi ini. Ini memungkinkan Anda untuk memprioritaskan upaya pengoptimalan Anda berdasarkan ukuran peluang tabungan. Rekomendasi Compute Optimizer menyediakan konfigurasi CPU dan memori tingkat kontainer untuk tugas perampingan.

Manfaat biaya

Ukuran tugas Amazon ECS yang tepat di Fargate dapat mengurangi biaya hingga 30-70 persen untuk tugas yang berjalan lama. Tanpa meninjau metrik kinerja aplikasi untuk ukuran tugas yang tepat, Anda dapat menerapkan pola pikir yang sama yang digunakan pada instans komputasi EC2 ke ukuran wadah. Hal ini menyebabkan tugas Fargate yang terlalu besar yang meningkatkan biaya untuk sumber daya yang menganggur. Anda dapat menggunakan Compute Optimizer untuk memunculkan peluang ukuran yang tepat secara reaktif. Idealnya, pemilik aplikasi meninjau metrik kinerja aplikasi tertentu dan menghapus overhead sistem operasi untuk memastikan ukuran tugas yang tepat ditentukan. Untuk informasi selengkapnya, lihat bagian Pindahkan aplikasi Windows ke kontainer dari panduan ini.

Rekomendasi optimisasi biaya

Bagian ini menawarkan rekomendasi untuk menggunakan Compute Optimizer untuk mengukur ukuran Amazon ECS Anda dengan tepat pada tugas Fargate.

Sebagai bagian dari proses pengoptimalan biaya, kami menyarankan Anda melakukan hal berikut:

- Aktifkan Compute Optimizer
- Konsumsi hasil Compute Optimizer
- Tandai tugas agar berukuran tepat
- Aktifkan tag alokasi biaya untuk bekerja dengan alat AWS penagihan
- Menerapkan rekomendasi ukuran yang tepat
- Tinjau biaya sebelum dan sesudah di Cost Explorer

Aktifkan Compute Optimizer

Anda dapat mengaktifkan AWS Compute Optimizerdi tingkat organisasi atau akun tunggal di AWS Organizations. Konfigurasi seluruh organisasi menyediakan laporan berkelanjutan untuk instans baru

dan yang sudah ada di seluruh armada Anda untuk semua akun anggota. Ini memungkinkan ukuran yang tepat menjadi aktivitas berulang, bukan aktivitas. point-in-time

Tingkat organisasi

Untuk sebagian besar organisasi, cara paling efisien untuk menggunakan Compute Optimizer adalah di tingkat organisasi. Ini memberikan visibilitas multi-akun dan Multi-wilayah ke organisasi Anda dan memusatkan data menjadi satu sumber untuk ditinjau. Untuk mengaktifkan ini di tingkat organisasi, lakukan hal berikut:

1. Masuk ke akun AWS Organizations manajemen Anda dengan peran yang memiliki izin yang diperlukan dan pilih untuk memilih semua akun dalam organisasi ini. Organisasi Anda harus mengaktifkan semua fitur.
2. Setelah mengaktifkan akun manajemen, Anda dapat masuk ke akun, melihat semua akun anggota lainnya, dan menelusuri rekomendasi mereka.

> **Note**
>
> Ini adalah praktik terbaik untuk mengonfigurasi akun administrator yang didelegasikan untuk Compute Optimizer. Ini memungkinkan Anda untuk menggunakan prinsip hak istimewa paling sedikit, meminimalkan akses ke akun AWS Organizations manajemen sambil tetap menyediakan akses ke layanan di seluruh organisasi.

Tingkat akun tunggal

Jika Anda menargetkan akun dengan biaya tinggi tetapi tidak memiliki akses AWS Organizations, Anda masih dapat mengaktifkan Compute Optimizer untuk akun dan Wilayah tersebut. Untuk mempelajari tentang proses keikutsertaan, lihat Memulai dengan AWS Compute Optimizer.

> **Note**
>
> Rekomendasi disegarkan setiap hari dan dapat memakan waktu hingga 12 jam untuk menghasilkan. Perlu diingat bahwa Compute Optimizer memerlukan 24 jam metrik dalam 14 hari terakhir untuk menghasilkan rekomendasi Amazon ECS di Fargate. Untuk informasi selengkapnya, lihat Persyaratan untuk layanan Amazon ECS di Fargate dalam dokumentasi Compute Optimizer.

Compute Optimizer secara otomatis menganalisis metrik pemanfaatan Amazon dan CloudWatch Amazon ECS berikut untuk layanan Amazon ECS Anda di Fargate:

- `CPUUtilization`— Persentase kapasitas CPU yang digunakan dalam layanan.
- `MemoryUtilization`— Persentase memori yang digunakan dalam layanan.

Konsumsi hasil Compute Optimizer

Pertimbangkan contoh yang berfokus pada membuat perubahan ukuran yang tepat dalam satu akun dan satu Wilayah. Dalam contoh ini, Compute Optimizer diaktifkan di tingkat organisasi di semua akun. Perlu diingat bahwa ukuran yang tepat adalah proses yang mengganggu yang dalam banyak kasus dilakukan dengan presisi oleh pemilik aplikasi selama jendela pemeliharaan terjadwal selama beberapa minggu.

Jika Anda menavigasi ke Compute Optimizer dari dalam akun manajemen organisasi (seperti yang ditunjukkan pada langkah-langkah berikut), Anda dapat memilih akun yang ingin Anda selidiki. Dalam contoh ini, satu tugas berjalan dalam satu akun yang dilebih-lebihkan. `us-east-1` Fokusnya adalah mengubah ukuran ke ukuran yang disarankan untuk layanan Amazon ECS.

1. Buka konsol Compute Optimizer.
2. Pada halaman Dasbor, filter menurut findings=Over-provisioned untuk melihat semua layanan Amazon ECS di Fargate.
3. Untuk meninjau rekomendasi terperinci untuk layanan ECS yang disediakan secara berlebihan di Fargate, gulir ke bawah lalu pilih Lihat rekomendasi.
4. Pilih Ekspor dan simpan file untuk digunakan di masa mendatang.

> **Note**
>
> Untuk menyimpan rekomendasi untuk tinjauan masa depan, Anda harus memiliki bucket S3 yang tersedia untuk Compute Optimizer untuk ditulis di setiap Wilayah. Untuk informasi selengkapnya, lihat kebijakan bucket Amazon S3 untuk dokumentasi AWS Compute Optimizer Compute Optimizer.

Untuk melihat rekomendasi dari Compute Optimizer, lakukan hal berikut:

1. Di konsol Compute Optimizer, buka halaman Rekomendasi ekspor.

2. Untuk tujuan bucket S3, pilih bucket S3 Anda.
3. Di bagian Ekspor filter, untuk jenis sumber daya, pilih layanan ECS di Fargate.
4. Pada Rekomendasi untuk layanan ECS di halaman Fargate, telusuri salah satu layanan ECS di Fargate dan lihat rekomendasi CPU dan memori dari Compute Optimizer. Misalnya, tinjau rekomendasi di bagian Bandingkan setelan saat ini dengan merekomendasikan ukuran tugas dan Bandingkan pengaturan saat ini dengan ukuran wadah yang disarankan.

Untuk mendapatkan daftar layanan ECS untuk Fargate yang Anda butuhkan untuk ukuran yang tepat, lakukan hal berikut:

1. Buka konsol Amazon S3.
2. Di panel navigasi, pilih Bucket, lalu pilih bucket tempat Anda mengekspor hasil.
3. Pada tab Objek, pilih objek Anda dan pilih Unduh.
4. Dalam hasil yang Anda unduh, filter kolom pencarian untuk hanya menampilkan layanan Amazon ECS OVER_PROVISIONED di Fargate. Ini menunjukkan layanan Amazon ECS yang Anda rencanakan untuk ditargetkan untuk ukuran yang tepat.
5. Simpan definisi tugas dalam editor teks untuk digunakan nanti.

Tugas tag ukuran yang tepat

Menandai beban kerja Anda adalah alat yang ampuh untuk mengatur sumber daya Anda. AWS Anda dapat menggunakan tag untuk mendapatkan visibilitas halus ke biaya dan mengaktifkan tolak bayar. Ada banyak metode dan strategi untuk menambahkan tag ke AWS sumber daya untuk menangani tolak bayar dan otomatisasi. Untuk informasi selengkapnya, lihat AWS Whitepaper Best Practices for AWS Tagging Resources. Contoh berikut digunakan AWS CloudShelluntuk menandai semua tugas yang merupakan bagian dari layanan Amazon ECS dalam akun target dan Wilayah AWS.

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$( aws ecs list-clusters -query 'clusterArns' -output text)
for ClustersArn in $ClustersArns; do
 ServiceArns=$( aws ecs list-services -cluster $ClustersArn -query 'serviceArns' -
output text)
 for ServiceArn in $ServiceArns; do
```

```
  TasksArns=$( aws ecs list-tasks –cluster $ClustersArn –service-name $ServiceArn –
query 'taskArns' –output text)
  for TasksArn in $TasksArns; do
    aws ecs tag-resource –resource-arn $TasksArn –tags key=$TAG_KEY,value=$TAG_VALUE
  done
 done
done
```

Contoh kode berikut menunjukkan cara mengaktifkan propagasi tag ke semua layanan Amazon ECS.

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$(aws ecs list-clusters --query 'clusterArns' --output text)
for ClustersArn in $ClustersArns; do
 ServiceArns=$(aws ecs list-services --cluster $ClustersArn --query 'serviceArns' --
output text)
 for ServiceArn in $ServiceArns; do
  aws ecs update-service --cluster $ClustersArn --service $ServiceArn --propagate-tags
 SERVICE &>/dev/null
  aws ecs tag-resource --resource-arn $ServiceArn --tags key=$TAG_KEY,value=$TAG_VALUE
 done
done
```

Aktifkan tag alokasi biaya untuk bekerja dengan alat AWS penagihan

Sebaiknya aktifkan tag alokasi biaya yang ditentukan pengguna. Ini memungkinkan tag Rightsizing dikenali dan difilter di alat AWS penagihan (misalnya, dan). AWS Cost Explorer AWS Cost and Usage Report Jika Anda tidak mengaktifkan ini, opsi pemfilteran tag dan data tidak akan tersedia. Untuk informasi tentang penggunaan tag alokasi biaya, lihat Mengaktifkan tag alokasi biaya yang ditentukan pengguna dalam dokumentasi. AWS Manajemen Penagihan dan Biaya

Setelah menunggu selama 24 jam, Anda dapat melihat tag di Cost Explorer sebelum menerapkan rekomendasi ukuran yang tepat di bagian berikutnya. Untuk melakukan ini, cari tag Rightsizing di Cost Explorer.

Menerapkan rekomendasi ukuran yang tepat

Compute Optimizer akan memberikan rekomendasi ukuran tugas atau wadah. Untuk menerapkan rekomendasi ukuran yang tepat, lakukan hal berikut.

1. Buka konsol Amazon ECS.
2. Dari bilah navigasi, pilih Wilayah yang berisi ketentuan tugas Anda.
3. Di panel navigasi, pilih Definisi tugas.
4. Pada halaman Definisi tugas, pilih tugas, lalu pilih Buat revisi baru.
5. Pada halaman Buat revisi definisi tugas baru, buat perubahan. Untuk memperbarui rekomendasi ukuran kontainer, perbarui `cpu` dan `memory` di bawah blok ContainerDefinitions dalam definisi tugas ECS Anda. Contoh:

```
"containerDefinitions": [
 {
  "name": "your-container-name",
  "image": "your-image",
  "cpu": 1024,
  "memory": 2048,
 }
],
```

6. Verifikasi informasi, lalu pilih Buat.

Untuk memperbarui layanan Amazon ECS, lakukan hal berikut:

1. Buka konsol Amazon ECS.
2. Pada halaman Clusters, pilih cluster.
3. Pada halaman Ikhtisar cluster, pilih layanan, lalu pilih Perbarui.
4. Untuk ketentuan tugas, pilih keluarga ketentuan tugas dan revisi yang akan digunakan.

Untuk operator tingkat lanjut, Anda dapat menggunakan CloudShell untuk memperbarui layanan Amazon ECS. Contoh:

```
bash
#!/bin/bash
# Set variables
ClustersName="workshop-cluster"
ServiceName="lab7-fargate-service"
TaskDefinition="lab7-fargate-demo:3"
# update the service
aws ecs update-service --cluster $ClustersName --service $ServiceName --task-definition
 $TaskDefinition
```

Tinjau biaya sebelum dan sesudah

Setelah Anda mengukur sumber daya dengan benar, Anda dapat menggunakan Cost Explorer untuk menampilkan sebelum dan sesudah biaya dengan menggunakan tag Rightsizing. Ingatlah bahwa Anda dapat menggunakan tag sumber daya untuk melacak biaya. Dengan menggunakan beberapa lapisan tag, Anda dapat mencapai visibilitas granular ke dalam biaya Anda. Dalam contoh yang tercakup dalam panduan ini, tag Rightsizing digunakan untuk menerapkan tag generik ke semua instance yang ditargetkan. Kemudian, tag tim digunakan untuk mengatur sumber daya lebih lanjut. Langkah selanjutnya adalah memperkenalkan tag aplikasi untuk lebih menunjukkan dampak biaya untuk mengoperasikan aplikasi tertentu.

Pertimbangkan contoh pengurangan biaya yang dapat dicapai dengan menggunakan tag Rightsizing untuk satu tingkat akun. Dalam contoh ini, biaya operasional naik dari $30,26 per hari menjadi $7,56 per hari. Dengan asumsi 744 jam per bulan, biaya tahunan sebelum ukuran yang tepat adalah $11.044.9. Setelah ukuran yang tepat, biaya tahunan turun menjadi $2.759.4. Ini berarti penurunan 75 persen dalam biaya komputasi untuk akun ini. Bayangkan dampak dari ini di seluruh organisasi besar.

Sebelum memulai perjalanan ukuran yang tepat, pertimbangkan hal berikut:

- AWS menawarkan banyak pilihan untuk pengurangan biaya. Ini termasuk AWS OLA, tempat AWS meninjau instans lokal Anda sebelum pindah ke. AWS AWS OLA juga memberi Anda rekomendasi ukuran yang tepat dan panduan perizinan.
- Selesaikan semua ukuran yang tepat sebelum membeli Savings Plans. Ini dapat membantu Anda menghindari pembelian berlebihan pada komitmen Savings Plans Anda.

Langkah selanjutnya

Kami merekomendasikan langkah-langkah berikut:

1. Tinjau lanskap Anda yang ada dan pertimbangkan untuk mengonversi volume Amazon EBS gp2 menjadi volume gp3.
2. Tinjau Savings Plans.

Sumber daya tambahan

- Memulai dengan Compute AWS Optimizer (dokumentasi)

- Praktik Terbaik untuk Menandai AWS Sumber Daya (AWS Whitepaper)
- Wadah Windows aktif AWS (Studio AWS Bengkel)

Dapatkan visibilitas ke biaya Amazon EKS Anda

Ikhtisar

Pandangan holistik diperlukan untuk secara efektif memantau biaya penyebaran Kubernetes. Satu-satunya biaya tetap dan diketahui adalah untuk pesawat kontrol Amazon Elastic Kubernetes Service (Amazon EKS). Ini termasuk setiap komponen lain yang membentuk penyebaran, dari komputasi dan penyimpanan hingga jaringan, menjadi jumlah variabel berdasarkan kebutuhan aplikasi Anda.

Anda dapat menggunakan Kubecost untuk menganalisis biaya infrastruktur Kubernetes Anda mulai dari Namespace dan Services hingga ke masing-masing Pod, dan kemudian menampilkan data di dasbor. Kubecost memunculkan biaya in-cluster seperti komputasi dan penyimpanan serta biaya out-of-cluster seperti bucket Amazon Simple Storage Service (Amazon S3) dan instance Amazon Relational Database Service (Amazon RDS). Kubecost akan membuat rekomendasi ukuran yang tepat berdasarkan data ini dan menampilkan peringatan kritis yang dapat memengaruhi sistem. Kubecost dapat berintegrasi dengan AWS Cost and Usage Reportuntuk menunjukkan penghematan dari Compute Savings Plans, Instans Cadangan, dan program discount lainnya.

Manfaat biaya

Kubecost menyediakan laporan dan dasbor yang memvisualisasikan biaya penerapan Amazon EKS Anda. Hal ini memungkinkan Anda untuk menelusuri dari cluster ke masing-masing dari berbagai komponen seperti controller, layanan, node, pod, dan volume. Ini memberi Anda pandangan holistik dari aplikasi Anda yang berjalan di lingkungan Amazon EKS. Dengan mengaktifkan visibilitas ini, Anda dapat menindaklanjuti rekomendasi Kubecost atau melihat biaya setiap aplikasi pada tingkat granular. Ukuran yang tepat grup node Amazon EKS menawarkan potensi penghematan yang sama dengan instans EC2 standar. Jika Anda dapat mengukur wadah dan node dengan benar, Anda dapat menghapus compute bloat dari ukuran instance yang diperlukan untuk menjalankan container dan jumlah instans EC2 yang diperlukan dalam grup penskalaan otomatis.

Rekomendasi optimisasi biaya

Untuk memanfaatkan Kubecost, kami sarankan Anda melakukan hal berikut:

1. Terapkan Kubecost ke lingkungan Anda

2. Dapatkan rincian biaya terperinci dari aplikasi Windows
3. Node cluster ukuran yang tepat
4. Permintaan wadah ukuran yang tepat
5. Kelola node yang kurang dimanfaatkan
6. Memperbaiki beban kerja yang ditinggalkan
7. Bertindak berdasarkan rekomendasi
8. Perbarui node yang dikelola sendiri

Terapkan Kubecost ke lingkungan Anda

Amazon EKS Finhack Workshop mengajarkan Anda cara menerapkan lingkungan Amazon EKS yang dikonfigurasi untuk menggunakan Kubecost di akun yang dimiliki. AWS Ini memungkinkan Anda untuk mendapatkan pengalaman langsung dengan teknologi. Jika Anda tertarik untuk menjalankan lokakarya ini di organisasi Anda, hubungi tim akun Anda.

Untuk menerapkan Kubecost ke klaster Amazon EKS Anda menggunakan Helm, lihat AWS dan Kubecost berkolaborasi untuk memberikan pemantauan biaya untuk posting pelanggan EKS di Blog. AWS Atau, Anda dapat merujuk ke dokumentasi resmi Kubecost untuk petunjuk tentang menginstal dan mengkonfigurasi Kubecost. Untuk informasi tentang dukungan Kubecost untuk node Windows, lihat Dukungan Node Windows dalam dokumentasi Kubecost.

Dapatkan rincian biaya terperinci dari aplikasi Windows

Meskipun Anda dapat mencapai penghematan biaya yang signifikan dengan menggunakan Instans Spot Amazon EC2, Anda juga bisa mendapatkan keuntungan dari kenyataan bahwa beban kerja Windows cenderung stateful. Penggunaan Instans Spot bergantung pada aplikasi, dan kami mendorong Anda untuk memverifikasi apakah instans tersebut akan berlaku untuk kasus penggunaan Anda.

Untuk mendapatkan rincian biaya terperinci dari aplikasi Windows Anda, masuk ke Kubecost. Di halaman navigasi, pilih Tabungan.

Node cluster ukuran yang tepat

Di Kubecost, pilih Savings dari navigation bar, lalu pilih Right-size node cluster Anda.

Pertimbangkan contoh di mana Kubecost melaporkan bahwa cluster disediakan secara berlebihan baik dalam hal vCPU dan RAM. Tabel berikut menunjukkan rincian dan rekomendasi dari Kubecost.

	Saat ini	Rekomendasi: Sederhana	Rekomendasi: Kompleks
Jumlah total	US $3462.57 per bulan	US $137,24 per bulan	US $303.68 per bulan
Jumlah simpul	4	5	4
CPU	74 VCPUs	10 VCPUs	8 VCPUs
RAM	152 GB	20 GB	18 GB
Kerusakan instans	2 c5.xlarge + 2 lebih	5 t3a.sedang	2 c5n.large + 1 lagi

Seperti yang dijelaskan dalam posting blog Kubecost Temukan kumpulan node yang optimal untuk klaster Kubernetes, opsi sederhana menggunakan satu grup node, sedangkan yang kompleks menggunakan pendekatan grup multi-node. Tombol Pelajari cara mengadopsi dapat melakukan pengubahan ukuran klaster sekali klik. Hal ini membutuhkan instalasi Kubecost Cluster Controller.

Jika Anda menggunakan node Windows yang dikelola sendiri yang tidak dibuat oleh eksctl, lihat Memperbarui grup node yang dikelola sendiri yang ada. Petunjuk ini menunjukkan cara mengubah jenis instans di template peluncuran Amazon EC2 yang digunakan oleh grup Auto Scaling.

Permintaan wadah ukuran yang tepat

Di Kubecost, pilih Penghematan dari bilah navigasi, dan buka halaman Permintaan rekomendasi ukuran kanan. Halaman ini menunjukkan efisiensi pod, rekomendasi ukuran yang tepat, dan perkiraan penghematan biaya. Anda dapat menggunakan tombol Customize untuk memfilter berdasarkan Cluster, Node, Namespace\ Controller, dan lainnya.

Sebagai contoh, pertimbangkan bahwa Kubecost telah menghitung bahwa beberapa pod Anda dilebih-lebihkan dalam hal CPU dan RAM (memori). Kemudian, Kubecost merekomendasikan agar Anda menyesuaikan dengan nilai CPU dan RAM baru untuk mencapai perkiraan penghematan bulanannya. Untuk mengubah nilai CPU dan RAM, Anda harus memperbarui file manifes penyebaran Anda.

Kelola node yang kurang dimanfaatkan

Di Kubecost, pilih Savings dari navigation bar, lalu pilih Manage underutilized nodes.

Pertimbangkan contoh di mana halaman menunjukkan bahwa satu node dalam cluster kurang dimanfaatkan dalam hal CPU dan RAM (memori) dan oleh karena itu dapat dikeringkan dan dihentikan atau diubah ukurannya. Memilih node yang tidak lulus pemeriksaan node dan pod akan memberi Anda lebih banyak informasi tentang mengapa mereka tidak dapat dikeringkan.

Memperbaiki beban kerja yang ditinggalkan

Di Kubecost, pilih Savings dari navigation bar, lalu pilih halaman Abandoned Workloads. Dalam contoh ini, Anda memfilter berdasarkan Namespace yang disebut windows. Halaman ini menunjukkan pod yang belum memenuhi ambang lalu lintas dan dianggap ditinggalkan. Pod perlu mengirim atau menerima sejumlah lalu lintas jaringan selama periode yang ditentukan.

Setelah mempertimbangkan dengan cermat bahwa satu atau beberapa pod ditinggalkan, Anda dapat menghemat biaya dengan mengurangi jumlah replika, menghapus penerapan, mengubah ukurannya untuk mengkonsumsi lebih sedikit sumber daya, atau memberi tahu pemilik aplikasi bahwa Anda yakin penerapan tersebut ditinggalkan.

Bertindak berdasarkan rekomendasi

Di bagian Right-size your cluster nodes, Kubecost menganalisis penggunaan node worker di cluster, dan membuat rekomendasi tentang ukuran node yang tepat untuk mengurangi biaya. Ada dua jenis grup node yang dapat digunakan dengan Amazon EKS: dikelola sendiri dan dikelola.

Perbarui node yang dikelola sendiri

Untuk informasi tentang memperbarui node yang dikelola sendiri, lihat Pembaruan node yang dikelola sendiri dalam dokumentasi Amazon EKS. Ini menyatakan bahwa grup node yang dibuat dengan tidak `eksctl` dapat diperbarui dan harus dimigrasikan ke grup node baru dengan konfigurasi baru.

Sebagai contoh, asumsikan bahwa Anda memiliki grup node Windows yang disebut `ng-windows-m5-2xlarge` (yang menggunakan instance EC2 m5.2xlarge) dan Anda ingin memigrasikan pod ke grup node baru yang disebut `ng-windows-t3-large` (yang didukung oleh instance EC2 t3.large untuk menghemat biaya).

Untuk bermigrasi ke grup node baru saat Anda menggunakan grup node yang digunakan oleh`eksctl`, lakukan hal berikut:

1. Untuk menemukan node yang saat ini adalah pod, jalankan `kubectl describe pod <pod_name> -n <namespace>` perintah.

2. Jalankan perintah `kubectl describe node <node_name>`. Output menunjukkan bahwa node berjalan pada instance m5.2xlarge. Ini juga cocok dengan nama grup node (`ng-windows-m5-2xlarge`).
3. Untuk mengubah penyebaran untuk menggunakan grup node`eng-windows-t3-large`, hapus grup node `ng-windows-m5-2xlarge` dan jalankan`kubectl describe svc,deploy,pod -n windows`. Penerapan segera mulai diterapkan kembali sekarang setelah grup simpulnya telah dihapus.

> **Note**
>
> Akan ada downtime layanan saat Anda menghapus grup node.

4. Jalankan `kubectl describe svc,deploy,pod -n windows` perintah lagi setelah beberapa menit. Outputnya menunjukkan bahwa semua pod berada dalam keadaan Running lagi.
5. Untuk menunjukkan bahwa pod sekarang berjalan pada grup node`eng-windows-t3-large`, jalankan `kubectl describe node <node_name>` perintah `kubectl describe pod <pod_name> -n <namespace>` and lagi.

Metode pengubahan ukuran alternatif

Metode ini berlaku untuk kombinasi grup node yang dikelola sendiri atau dikelola. Beban kerja yang bermigrasi dengan mulus dari grup node yang dikelola sendiri EKS ke posting blog grup node yang dikelola EKS memberikan panduan tentang cara memigrasikan beban kerja Anda dari satu grup node dengan tipe instance besar ke grup node yang berukuran tepat tanpa waktu henti apa pun.

Langkah selanjutnya

Kubecost memudahkan untuk memvisualisasikan biaya lingkungan Amazon EKS Anda. Integrasi mendalam Kubecost dengan Kubernetes dan AWS APIs dapat membantu Anda menemukan potensi penghematan biaya. Anda dapat melihat ini sebagai rekomendasi di dasbor Tabungan Kubecost. Kubecost juga dapat mengimplementasikan beberapa rekomendasi ini untuk Anda melalui fitur cluster controller.

Kami menyarankan Anda meninjau step-by-step penerapan di AWS dan Kubecost berkolaborasi untuk memberikan pemantauan biaya untuk posting blog pelanggan EKS dari blog Containers. AWS

Sumber daya tambahan

- Lokakarya Amazon EKS (Lokakarya Amazon EKS)
- AWS dan Kubecost berkolaborasi untuk memberikan pemantauan biaya bagi pelanggan EKS (Blog)AWS
- Lokakarya Amazon EKS Finhack (Studio AWS Lokakarya)
- Wadah Windows aktif AWS (Studio AWS Bengkel)

Replatform aplikasi Windows dengan App2Container

Ikhtisar

AWS App2Containeradalah alat baris perintah untuk memigrasi dan memodernisasi aplikasi web Java dan .NET ke dalam wadah. App2Container menganalisis dan membuat inventaris semua aplikasi yang berjalan di bare metal, mesin virtual, instans Amazon Elastic Compute Cloud (Amazon EC2), atau di penyedia cloud lainnya. Anda memilih aplikasi yang ingin Anda kontainerisasi. App2Container mengemas artefak dan dependensi aplikasi ke dalam gambar kontainer, mengonfigurasi port jaringan, dan menghasilkan artefak penyebaran Amazon Elastic Container Service (Amazon ECS) dan Amazon Elastic Kubernetes Service (Amazon EKS) yang diperlukan, yang merupakan templat infrastruktur sebagai kode (IAc). App2Container menyediakan infrastruktur cloud dan pipeline CI\ CD yang diperlukan untuk menyebarkan aplikasi kontainer ke dalam lingkungan produksi. Untuk informasi selengkapnya, lihat Cara kerja App2Container dalam dokumentasi App2Container.

Dengan App2Container, Anda dapat bermigrasi ke AWS dan memodernisasi aplikasi Anda sebagai kontainer sambil juga menstandarisasi penerapan dan operasi untuk aplikasi Anda. Anda dapat menggunakan App2Container untuk membantu membangun bukti konsep (PoC) dengan cepat atau mempercepat penerapan beban kerja produksi dalam kontainer.

Ada beberapa hal yang perlu diingat ketika bekerja dengan aplikasi Windows. App2Container mendukung kontainerisasi aplikasi ASP.NET yang digunakan pada Microsoft Internet Information Services (IIS), termasuk aplikasi Windows Communication Foundation (WCF) yang dihosting IIS yang berjalan di Windows Server 2016, Windows Server 2019, atau Windows Server Core 2004. Untuk informasi selengkapnya, lihat Aplikasi yang didukung untuk Windows dalam dokumentasi App2Container. App2Container menggunakan Windows Server Core sebagai gambar dasar untuk artefak kontainernya, mencocokkan versi kontainer Windows Server Core dengan versi sistem operasi (OS) server tempat Anda menjalankan perintah containerization. Pendekatan ini memisahkan

aplikasi dari OS yang mendasarinya sehingga Anda dapat meningkatkan OS tanpa melakukan migrasi tradisional.

Jika Anda menggunakan mesin pekerja untuk mengkontainerisasi aplikasi Anda, gambar dasar kontainer, seperti saluran servis jangka panjang Windows Server 2019 (LTSC), cocok dengan OS mesin pekerja Anda, seperti Windows Server 2019. Jika Anda menjalankan kontainerisasi langsung di server aplikasi, versinya cocok dengan OS server aplikasi Anda. Jika aplikasi Anda berjalan pada Windows Server 2008 atau 2012 R2, Anda masih dapat menggunakan App2Container dengan menyiapkan mesin pekerja untuk langkah-langkah containerization dan deployment. App2Container tidak mendukung aplikasi yang berjalan pada sistem operasi klien Windows, seperti Windows 7 atau Windows 10. App2Container mendukung kerangka kerja Tomcat, ToMee, dan JBoss (mode mandiri) untuk proses Java. Untuk informasi selengkapnya, lihat kompatibilitas App2Container.

Manfaat biaya

Containerizing dan konsolidasi aplikasi Anda dapat menghasilkan penghematan komputasi hingga 60% jika dibandingkan dengan pola desain penerapan one-application-to-one -server. App2Container membantu mempercepat proses kontainerisasi aplikasi. Berikut ini adalah beberapa manfaat menggunakan App2Container untuk kebutuhan modernisasi Anda:

- App2Container ditawarkan tanpa biaya tambahan.
- App2Container mendukung beberapa aplikasi dalam gambar kontainer.
- Atasi sistem operasi yang mendekati akhir dukungan dengan menggunakan App2Container untuk memindahkan aplikasi.NET lama Anda ke kontainer. Anda dapat pindah ke sistem operasi yang lebih baru, menghindari membayar untuk dukungan yang diperpanjang, dan mengurangi risiko keamanan.
- Kontainer adalah metode yang efisien dan hemat biaya untuk mengemas aplikasi .NET Anda. Tinjau manfaat wadah dalam Rekomendasi MACO - Pindah ke wadah.
- Konsolidasi aplikasi dan kontainerisasi membantu mengurangi jejak komputasi, penyimpanan, dan lisensi Anda dengan menggunakan sumber daya komputasi Anda secara lebih efisien.
- Pindah ke kontainer dapat mengurangi biaya overhead dan infrastruktur operasional dan meningkatkan portabilitas pengembangan dan kelincahan penyebaran.

Rekomendasi optimisasi biaya

Untuk petunjuk tentang cara menggunakan App2Container, lihat Memulai. AWS App2Container Untuk informasi tentang perintah App2Container, lihat referensi perintah App2Container.

Langkah selanjutnya

App2Container dapat mempercepat proses aplikasi containerizing dan deploying ke Amazon EKS atau Amazon ECS. Penyebaran aplikasi ke kontainer mengurangi biaya komputasi, jaringan, dan penyimpanan serta mengurangi overhead operasional untuk operator aplikasi.

Untuk pengalaman langsung dengan App2Container, lihat Modernisasi dengan Workshop. AWS App2Container Jika Anda ingin memiliki pengalaman belajar mendalam, mintalah tim AWS akun Anda untuk menyiapkan hari imersi App2Container.

Sumber daya tambahan

- Containerizing Kompleks Aplikasi Windows Multi-tier menggunakan AWS App2Container(posting blog)AWS
- Containerizing aplikasi ASP.NET lama menggunakan AWS App2Container (posting blog)AWS
- Aplikasi yang didukung App2Container (dokumentasi)AWS
- Modernisasi dengan AWS App2Container Workshop (AWS Workshop Studio)
- AWS App2Container FAQs(AWS situs web)

Penyimpanan

Memilih penyimpanan yang tepat untuk beban kerja Microsoft Anda adalah keputusan arsitektur yang penting. Sebagai bagian dari proses pengambilan keputusan, kami menyarankan Anda mengembangkan rencana penyimpanan dan menentukan persyaratan fungsional untuk aplikasi dan layanan Anda. Bab ini memberikan gambaran umum tentang opsi penyimpanan berikut yang dapat menjadi faktor dalam perencanaan Anda.

Bagian:

- Amazon EBS
- Amazon FSx
- AWS Storage Gateway

Amazon EBS

Amazon Elastic Block Store (Amazon EBS) adalah layanan penyimpanan blok terkelola penuh yang memungkinkan Anda menyimpan volume penyimpanan tingkat blok persisten yang dapat Anda gunakan dengan instans Amazon Elastic Compute Cloud (Amazon EC2). Anda dapat memanfaatkan beberapa fitur di Amazon EBS untuk mengelola dan mengoptimalkan sumber daya penyimpanan Anda secara efektif untuk beban kerja Windows di cloud. Misalnya, Anda dapat menggunakan Amazon EBS untuk menyediakan jumlah IOPS dan throughput yang tepat yang Anda perlukan untuk beban kerja Anda, memilih dari berbagai jenis volume agar sesuai dengan persyaratan beban kerja Anda, dan menggunakan alat untuk mengidentifikasi dan menghilangkan sumber daya penyimpanan yang terbuang. Kontrol terperinci atas kinerja dan penggunaan penyimpanan ini membantu Anda mengoptimalkan sumber daya penyimpanan sekaligus menghindari biaya yang tidak perlu.

Bagian ini mencakup topik-topik berikut:

- Migrasikan volume Amazon EBS dari gp2 ke gp3
- Ubah snapshot Amazon EBS
- Hapus volume Amazon EBS yang tidak terpasang

Migrasikan volume Amazon EBS dari gp2 ke gp3

Ikhtisar

Solid-state drive (SSD) adalah opsi penyimpanan standar untuk produksi dan beban kerja berkinerja tinggi. Amazon EBS menawarkan volume SSD tujuan umum untuk beban kerja kinerja menengah hingga tinggi. Standar di banyak Layanan AWS (termasuk Amazon EC2) adalah gp2, generasi kedua dari volume SSD tujuan umum ini. Generasi ketiga tujuan umum SSDs, yang disebut gp3, dirilis pada Desember 2020.

Penawaran gp3 membuat peningkatan signifikan pada aspek kustomisasi kinerja dibandingkan generasi sebelumnya. Untuk volume Amazon EBS gp2, kinerjanya sangat erat dengan ukuran volume. Untuk setiap kapasitas 1 GB, volume gp2 mendapatkan 3 IOPS kinerja. Artinya, volume 2.000 GB gp2 mampu 6.000 IOPS. Untuk volume gp3, kinerja dapat disesuaikan secara independen dari kapasitas penyimpanan. Ini memungkinkan volume kapasitas kecil sekalipun untuk mencapai kemampuan kinerja hingga 80.000 IOPS dan throughput 2.000 MB/s.

Perubahan besar lainnya dengan volume gp3 adalah kinerja IOPS dasar. Volume gp3 mulai dari 3.000 IOPS. Sebagai perbandingan, volume gp2 harus mencapai ukuran 1 TiB sebelum mencapai kemampuan kinerja yang sama. Untuk Windows Server, yang biasanya memiliki drive C: jauh lebih kecil dari 1 TiB, upgrade dari gp2 ke gp3 adalah peningkatan kinerja yang signifikan.

Akhirnya, harga volume gp3 adalah salah satu peningkatan terbesar dibandingkan dengan volume gp2. Volume gp3 menawarkan kemampuan kinerja yang ditingkatkan dengan biaya 20 persen lebih rendah daripada volume gp2.

Dampak biaya

Dengan kemampuan untuk menskalakan kinerja secara independen dari kapasitas, penting untuk memahami aspek penetapan harga dari penambahan IOPS dan throughput tambahan. Untuk volume gp2, harga didasarkan pada kapasitas yang disediakan sebesar $0,10 per bulan GIB. Untuk volume gp3, harga mirip dengan volume IOPS SSD yang disediakan berkinerja tinggi, yang memiliki satu biaya untuk kapasitas dan biaya terpisah untuk IOPS dan throughput tambahan.

Seperti disebutkan dalam tabel berikut, volume gp3 memiliki harga kapasitas $0,08 per GIB-bulan (20 persen lebih murah daripada gp2) dan biaya terpisah untuk IOPS pada $0,005 per IOPS yang disediakan - bulan lebih dari 3.000 dan $0,04 per bulan yang disediakan lebih dari 125 untuk throughput. MiBs MiBs

	gp3	gp2
Ukuran volume	1 GiB - 64 TiB	1 GiB - 16 TiB
IOPS dasar	3.000	3 IOPS/GiB (minimal 100 IOPS) hingga maksimal 16.000 IOPS Volume yang lebih kecil dari 1 TiB dapat meledak hingga 3.000 IOPS
IOP/Volume Maks	80.000	16.000
Throughput dasar	125 MiBs	Batas throughput adalah antara 128 MiBs — 250 MiBs, tergantung pada ukuran volume
Throughput/volume maks	2.000 MiBs	250 MiBs
Harga	$0,08/GIB-bulan 3.000 IOPS gratis dan $0,005/bulan IOPS yang disediakan lebih dari 3.000 125 MiBs gratis dan $0,04/disediakan MiBs -bulan di atas 125 MiBs	$0.10/GIB-bulan

Important

Meskipun volume gp3 memiliki biaya terpisah untuk kapasitas dan kinerja, volume gp3 selalu lebih murah daripada volume gp2 jika dikonfigurasi pada tingkat kinerja yang sama.

Tabel berikut menunjukkan contoh penghematan biaya yang dapat dicapai dengan mengubah volume gp2 ke gp3 pada berbagai konfigurasi kapasitas dan kinerja.

Contoh untuk konfigurasi gp2

Ukuran volume (GiB)	Max IOPS	Throughput () MiBs	Biaya (USD/bulan)
30	3000	128	$3,00
100	3000	128	$10.00
500	3000	250	$50,00
1000	3000	250	$100.00
2000	6000	250	$200.00
6000	16000	250	$600.00

Contoh untuk konfigurasi gp3 (baseline)

Max IOPS	Throughput () MiBs	Biaya (USD/bulan)	Pengurangan biaya (dibandingkan dengan gp2)
3000	125	$2,40	20%
3000	125	$8.00	20%
3000	125	$40,00	20%
3000	125	$80,00	20%
3000	125	$160.00	20%
3000	125	$480.00	20%

Contoh untuk konfigurasi gp3 (gp2 matching)

Max IOPS	Throughput () MiBs	Biaya (USD/bulan)	Pengurangan biaya (dibandingkan dengan gp2)
3000	128	$2,52	16%
3000	128	$8.12	19%
3000	250	$45,00	10%
3000	250	$85.00	15%
6000	250	$180,00	10%
16000	250	$550,00	8%

Untuk analisis biaya, lihat bagian kalkulator penghematan biaya migrasi EBS gp2 ke gp3 di sumber daya Amazon EBS. Anda dapat mengunduh kalkulator dan menggunakannya untuk mengetahui berapa banyak yang dapat Anda hemat dengan memigrasikan volume gp2 Anda ke gp3.

Rekomendasi optimisasi biaya

Untuk petunjuk tentang cara menyelesaikan proses migrasi, lihat Memigrasikan volume Amazon EBS Anda dari gp2 ke gp3 dan hemat hingga 20% pada postingan biaya di Blog Penyimpanan. AWS

Sumber daya tambahan

- Migrasikan volume Amazon EBS Anda dari gp2 ke gp3 dan hemat biaya hingga 20% (Blog Penyimpanan)AWS
- Buat Aturan AWS Config Kustom untuk Mengoptimalkan Jenis Volume Amazon EBS (Blog Operasi & Migrasi AWS Cloud)
- Mengontrol AWS biaya Anda dengan menghapus volume Amazon EBS yang tidak digunakan (AWS Cloud Operations & Migrations Blog)
- Utilitas Migrasi Amazon EBS () GitHub
- Menemukan penghematan mulai 2020 re: Invent pengumuman (Cloud Financial Management)AWS
- Workshop Optimalisasi Biaya (AWS Well-Architected Labs)

- kalkulator penghematan biaya migrasi gp2 ke gp3 (unduh)

Ubah snapshot Amazon EBS

Ikhtisar

Menghapus volume EBS dan mengelola retensi dan pengarsipan snapshot merupakan aspek penting untuk mengontrol biaya sejak awal. Anda dapat mencadangkan data pada volume EBS Anda ke Amazon Simple Storage Service (Amazon S3) dengan mengambil snapshot. point-in-time Snapshot adalah cadangan tambahan, jadi mereka hanya menyimpan blok pada perangkat yang berubah setelah snapshot terbaru Anda. Hal ini meminimalkan waktu yang diperlukan untuk membuat snapshot dan menghemat biaya penyimpanan dengan tidak menduplikasi data. Setiap snapshot berisi semua informasi yang diperlukan untuk memulihkan data Anda (dari saat snapshot dibuat) ke volume EBS baru.

Biaya untuk snapshot EBS dihitung berdasarkan bulan gigabyte. Anda ditagih untuk seberapa besar snapshot dan berapa lama Anda menyimpan snapshot. Harga bervariasi tergantung pada tingkat penyimpanan. Untuk tingkat Standar, Anda hanya ditagih untuk blok yang diubah yang disimpan. Untuk tingkat Arsip, Anda ditagih untuk semua blok snapshot yang disimpan. Anda juga ditagih untuk mengambil snapshot dari tingkat Arsip. Berikut ini adalah contoh skenario untuk setiap tingkat penyimpanan:

- Tingkat standar — Anda memiliki volume yang menyimpan 100 GB data. Anda ditagih untuk 100 GB data penuh untuk snapshot pertama (snap A). Pada saat snapshot berikutnya (snap B), Anda memiliki 105 GB data. Anda kemudian ditagih hanya untuk penyimpanan tambahan 5 GB untuk snap B tambahan.
- Tingkat arsip — Anda mengarsipkan snap B. Snapshot kemudian dipindahkan ke tingkat Arsip, dan Anda ditagih untuk blok snapshot 105 GB penuh.

Anda dapat menggunakan Amazon Data Lifecycle Manager untuk membantu menyiapkan siklus hidup untuk mempertahankan dan mengelola snapshot sesuai jadwal.

Dampak biaya

Biaya untuk volume dan snapshot EBS dikelola secara terpisah. Snapshot EBS ditagih pada tingkat yang lebih rendah daripada volume EBS aktif. Ketika sebuah instance berakhir, nilai DeleteOnTermination atribut untuk setiap volume EBS terlampir menentukan apakah akan

mempertahankan atau menghapus volume. Secara default, `DeleteOnTermination` atribut diatur ke `True` untuk volume root. Ini diatur `False` untuk semua jenis volume lainnya. Ini menciptakan situasi di mana operator bermaksud untuk menghapus instans EC2, tetapi meninggalkan volume yang ditambahkan ke instance selain volume root. Untuk petunjuk tentang memeriksa volume (dan snapshot terkait) yang tidak lagi Anda perlukan, lihat Melihat informasi tentang volume Amazon EBS dalam dokumentasi Amazon EBS.

Secara default, saat Anda membuat snapshot, snapshot disimpan di tingkat Standar Amazon EBS Snapshot (tingkat standar). Snapshot yang disimpan di tingkat standar bersifat inkremental. Hal ini berarti bahwa hanya blok pada volume yang berubah setelah snapshot terbaru Anda disimpan. Amazon EBS Snapshots Archive adalah tingkat penyimpanan baru yang dapat Anda gunakan untuk penyimpanan jangka panjang berbiaya rendah dari snapshot yang jarang diakses yang tidak memerlukan pengambilan yang sering atau cepat. Perbedaan harga dari standar ke arsip sangat signifikan dan harus menjadi pertimbangan utama saat menyiapkan strategi snapshot Anda. Arsip Snapshot Amazon EBS menawarkan biaya penyimpanan snapshot hingga 75 persen lebih rendah untuk snapshot yang Anda rencanakan untuk disimpan selama 90 hari atau lebih lama dan yang jarang perlu Anda akses.

Penyimpanan Snapshot Amazon EBS	Biaya
Standar	$0,05/GB-bulan
Arsip	$0.0125/GB-bulan

Di lingkungan yang lebih kecil, penghematan biaya mungkin tidak signifikan. Penghematan lebih signifikan dalam skala besar di mana ada banyak akun dan ribuan instans EC2 dengan TBs snapshot EBS yang duduk bahkan ketika volume EBS telah dihapus.

Tabel berikut membandingkan tingkatan standar dan arsip per bulan dengan hanya 50 TB penggunaan. Bahkan pada skala yang lebih rendah ini masih ribuan dolar tabungan setiap tahunnya.

Penyimpanan Snapshot Amazon EBS	Biaya per bulan	Biaya per tahun
Standar 50 TB	$312,50	$3.750
Arsip 50 TB	$78,13	$937,60

Penyimpanan Snapshot Amazon EBS	Biaya per bulan	Biaya per tahun
	Tabungan tahunan	$2.812,40

Rekomendasi optimisasi biaya

Menghapus snapshot mungkin tidak akan mengurangi biaya penyimpanan data organisasi Anda. Snapshot lain mungkin merujuk pada data snapshot, dan data yang direferensikan selalu dipertahankan. Misalnya, ketika Anda mengambil snapshot pertama dari volume dengan 10 GiB data, ukuran snapshot juga 10 GiB. Karena snapshot bersifat penambahan, snapshot kedua yang Anda ambil dengan volume yang sama hanya berisi blok data yang berubah sejak snapshot pertama diambil. Snapshot kedua juga mereferensikan data di snapshot pertama. Jika Anda mengubah 4 GiB data dan mengambil snapshot kedua, ukuran snapshot kedua adalah 4 GiB. Selain itu, snapshot kedua mereferensikan pada 6 GiB yang tidak berubah di snapshot pertama. Untuk informasi selengkapnya, lihat Mengapa biaya penyimpanan saya tidak berkurang setelah saya menghapus snapshot volume EBS saya dan kemudian menghapus volume itu sendiri? di pusat AWS pengetahuan.

Pertimbangkan hal berikut:

- Anda tidak ditagih untuk snapshot yang Akun AWS dimiliki dan dibagikan orang lain dengan akun Anda. Anda ditagih hanya jika menyalin snapshot bersama ke akun Anda. Anda juga ditagih untuk volume EBS yang Anda buat dari snapshot bersama.
- Jika snapshot (snap A) direferensikan oleh snapshot lain (snap B), maka menghapus snap B mungkin tidak mengurangi biaya penyimpanan. Saat Anda menghapus snapshot, hanya data yang unik untuk snapshot tersebut yang akan dihapus. Data yang direferensikan oleh snapshot lain tetap ada, dan Anda ditagih untuk data yang direferensikan ini. Untuk menghapus snapshot tambahan, lihat Penghapusan snapshot tambahan di dokumentasi Amazon EBS.

Kebersihan snapshot adalah praktik pengoperasian standar saat menjalankan beban kerja Anda. AWS Seiring waktu, snapshot dapat menambah biaya mahal untuk data yang tidak Anda butuhkan.

Sumber daya tambahan

- Mengontrol AWS biaya Anda dengan menghapus volume Amazon EBS yang tidak digunakan (AWS Cloud Operations & Migrations Blog)

- Hapus snapshot Amazon EBS (dokumentasi Amazon EBS)
- Workshop Optimalisasi Biaya (AWS Well-Architected Labs)
- Secara otomatis mengarsipkan Snapshot Amazon EBS dengan Amazon DataAWS Lifecycle Manager (Blog Penyimpanan)

Hapus volume Amazon EBS yang tidak terpasang

Ikhtisar

Volume EBS yang tidak terikat (yatim piatu) dapat menyebabkan biaya penyimpanan yang tidak perlu di lingkungan Anda. AWS Sangat penting untuk memasukkan tinjauan rutin dan penghapusan volume EBS yang tidak terpakai dan tidak digunakan sebagai bagian dari kebersihan lingkungan Anda. AWS Merupakan praktik terbaik untuk memiliki proses untuk terus meninjau penggunaan volume EBS. Anda dapat menggunakan AWS Compute Optimizeruntuk meninjau contoh yang kurang dimanfaatkan. Bagian ini membantu Anda mengidentifikasi, mengelola, dan menghapus volume EBS yang tidak terhubung atau kurang dimanfaatkan.

Amazon EBS

Amazon Elastic Block Store (Amazon EBS) adalah perangkat tingkat blok yang menawarkan volume penyimpanan untuk instans Amazon Elastic Compute Cloud (Amazon EC2). EBS menyediakan penyimpanan persisten, dengan fleksibilitas untuk memasang dan melepaskan dari instans EC2. Ini berarti siklus hidup volume EBS tetap ada bahkan jika instans EC2 dihentikan. DeleteOnTerminationAtribut adalah fitur yang mengontrol apakah akan mempertahankan atau menghapus volume EBS terlampir pada saat penghentian instance. Secara default, atribut diatur ke `True` untuk volume root, menghasilkan penghapusan. Ini diatur `False` untuk volume lain, menghasilkan pelestarian.

Dampak biaya

Volume EBS yang tidak terpasang, juga disebut sebagai volume yang tidak digunakan atau yatim piatu, dikenakan biaya yang sama dengan volume terlampir berdasarkan ukuran penyimpanan dan jenis penyimpanan yang disediakan. Meskipun biaya rata-rata biaya Amazon EBS mungkin tampak minimal pada $0,10 per GB-bulan, penting untuk menyadari bahwa akumulasi volume EBS yang tidak digunakan dapat menghasilkan biaya yang signifikan dari waktu ke waktu.

Misalnya, pertimbangkan konsekuensi dari mempertahankan 50 volume EBS yang tidak terpakai, masing-masing disediakan dengan ukuran penyimpanan 100 GB, seperti yang ditunjukkan tabel berikut.

Jumlah volume penyimpanan	Tipe volume	Size	Total biaya bulanan
50 volume	gp2 ($0,10 USD)	100 GB	100 GB 50.00 volume EBS bulan $0.10 USD = $500.00 USD

Skenario dari tabel sebelumnya menghasilkan pengurangan biaya sekitar $500 per bulan atau $6.000 per tahun. Ini adalah langkah efektif menuju pengurangan biaya. Pastikan untuk memasukkan penghapusan volume EBS yang tidak terikat sebagai praktik rutin dalam kebersihan lingkungan Anda. AWS

Rekomendasi optimisasi biaya

Anda dapat menggunakannya AWS untuk dengan mudah mengotomatiskan penghapusan volume EBS yang tidak terpasang. Misalnya, Anda dapat menggunakan AWS Lambda, Amazon AWS Config CloudWatch, dan AWS Systems Manager untuk menentukan kriteria untuk menghapus volume yang tidak terikat berdasarkan usia, tag, dan spesifikasi lainnya. Anda juga dapat menggunakan ini Layanan AWS untuk mengotomatiskan proses pembersihan dalam skala besar.

Untuk menghindari konsekuensi yang tidak diinginkan, sebaiknya Anda melakukan uji tuntas sebelum menghapus volume EBS yang tidak terpasang.

Kelola volume EBS yang tidak terpasang

Kami menyarankan Anda mempertimbangkan praktik terbaik berikut:

- Memenuhi persyaratan kepatuhan — Verifikasi bahwa penghapusan volume EBS yang tidak terikat sesuai dengan persyaratan tata kelola dan kepatuhan organisasi Anda.
- Tetapkan kebijakan pencadangan dan penyimpanan data — Sebelum menghapus volume EBS yang tidak terpasang, buat cadangan data penting apa pun ke repositori penyimpanan lain (misalnya, Amazon S3). Untuk retensi data, snapshot Amazon EBS adalah cara yang lebih hemat biaya untuk menyimpan data daripada volume EBS, dan mereka dapat memulihkan volume jika

diperlukan di masa mendatang. Untuk informasi selengkapnya tentang mengelola snapshot secara efektif, lihat bagian Memodifikasi snapshot Amazon EBS dari panduan ini.

- Periksa dependensi — Periksa dependensi apa pun antara volume EBS yang tidak terpasang dan sumber daya lainnya. AWS Anda dapat menggunakan Konsol Manajemen AWS atau API untuk mengumpulkan informasi deskriptif tentang volume EBS Anda, seperti ukuran, status, dan sumber daya terkait. Ini adalah langkah penting untuk melindungi terhadap penghapusan sumber daya sementara yang tidak terikat.
- Buat kebijakan retensi — Tetapkan periode retensi untuk volume EBS yang tidak terikat. Ini dapat membantu Anda mengidentifikasi waktu yang tepat untuk menghapus volume yang tidak terikat, memastikan bahwa AWS lingkungan Anda tetap dioptimalkan. Misalnya, Anda dapat membuat EventBridge aturan Amazon untuk memulai fungsi Lambda secara terjadwal. Fungsi Lambda dapat menggunakan AWS SDK untuk secara aktif mengidentifikasi volume EBS yang tidak terpasang, menerapkan mekanisme penandaan untuk memudahkan pelacakan, dan mengirimkan notifikasi saat volume EBS yang tidak terpasang mencapai atau melebihi ambang batas yang ditentukan.
- Menandai volume EBS yang tidak terpasang — Menandai volume EBS adalah praktik yang berguna yang dapat membantu dalam mengatur dan mengidentifikasi volume berdasarkan atribut seperti lingkungan, aplikasi, atau pemilik. Ini bisa sangat membantu ketika memutuskan volume yang tidak terikat untuk dihapus, karena memungkinkan Anda untuk dengan cepat mengidentifikasi volume yang tidak lagi diperlukan berdasarkan tag mereka.
- Pastikan penghapusan aman — Meninjau kapan volume EBS terakhir dilampirkan dapat membantu Anda menentukan apakah aman untuk menghapus volume. Untuk informasi selengkapnya, lihat Bagaimana cara menggunakan AWS CLI perintah untuk mencantumkan lampiran atau riwayat detasemen volume Amazon EBS tertentu? di pusat AWS pengetahuan.
- Identifikasi volume EBS yang kurang dimanfaatkan — Mengidentifikasi dan menghapus volume EBS yang kurang dimanfaatkan adalah praktik yang sangat direkomendasikan untuk mengurangi biaya penyimpanan dan mempertahankan lingkungan yang dioptimalkan. AWS AWS Trusted Advisor dan AWS Compute Optimizerdapat membantu Anda mengidentifikasi volume EBS yang kurang dimanfaatkan dan memberikan rekomendasi untuk mengurangi biaya dan meningkatkan efisiensi. Misalnya, lihat Menyiapkan otomatisasi untuk mengoptimalkan volume EBS dengan AWS Trusted Advisor (GitHub), Membuat dasbor Trusted Advisor Organisasi (TAO) (Studio AWS Lokakarya), dan Mengoptimalkan biaya volume AWS Compute Optimizer Amazon EBS menggunakan (Blog Penyimpanan).AWS

Otomatiskan pembersihan volume EBS yang tidak terpasang

Kami menyarankan Anda mempertimbangkan alat-alat berikut untuk membantu Anda mengotomatiskan pembersihan volume EBS yang tidak terpasang:

- AWS APIs (DescribeVolumes) — Anda dapat memfilter dan menemukan volume EBS yang tidak terpasang dengan menggunakan AWS SDKs atau AWS Command Line Interface ()AWS CLI. Anda dapat menghemat waktu dan tenaga dengan mengotomatiskan proses ini dengan skrip atau fungsi Lambda yang berjalan sesuai jadwal. Contoh skrip dari GitHub menunjukkan cara kerjanya. Skrip menggunakan Lambda untuk menganalisis AWS CloudTrail log dan mengidentifikasi volume EBS yang tidak terikat.
- AWS Systems Manager Otomatisasi — Ini memungkinkan Anda untuk mengotomatiskan tugas pemeliharaan dan remediasi rutin di infrastruktur Anda. Untuk memulai, buat runbook otomatisasi, yang mendefinisikan serangkaian langkah yang akan dijalankan dalam urutan tertentu. Misalnya, Anda dapat membuat runbook yang pertama kali membuat snapshot dari volume EBS yang tidak terpasang dan kemudian menghapus volume itu sendiri. Ini dapat membantu Anda mengotomatiskan tugas yang seharusnya memakan waktu dan rawan kesalahan jika dilakukan secara manual.
- AWS Config— Ini memungkinkan Anda untuk menilai, mengaudit, dan melacak perubahan pada AWS sumber daya Anda dari waktu ke waktu. Dengan menangkap perubahan konfigurasi, Anda dapat menggunakannya AWS Config untuk mengevaluasi kepatuhan, tata kelola, dan pemanfaatan sumber daya di lingkungan Anda. Misalnya, AWS Config dapat mengidentifikasi volume EBS yang tidak digunakan. Selain itu, Anda dapat mengaitkan AWS Systems Manager Otomasi dengan AWS Config untuk secara otomatis memulihkan penghapusan volume EBS yang tidak digunakan.

Sumber daya tambahan

- Hapus volume Amazon Elastic Block Store (Amazon EBS) yang tidak terpakai dengan menggunakan AWS Config dan (Panduan Preskriptif) AWS Systems ManagerAWS
- Mengontrol AWS biaya Anda dengan menghapus volume Amazon EBS yang tidak digunakan (AWS Cloud Operations & Migrations Blog)
- AWSConfigRemediation-DeleteUnusedEBSVolume(Referensi buku runbook AWS Systems Manager otomatisasi)

Amazon FSx

Amazon FSx untuk Windows File Server adalah layanan penyimpanan file yang dikelola sepenuhnya yang dioptimalkan untuk beban kerja Windows. Ini memberi Anda solusi sederhana dan terukur untuk menjalankan aplikasi dan beban kerja berbasis Windows Anda, tanpa perlu manajemen infrastruktur penyimpanan yang kompleks. Anda dapat menggunakan Windows File Server FSx untuk dengan mudah menyediakan dan mengakses penyimpanan file bersama yang mendukung aplikasi Windows Anda secara native, termasuk Microsoft SQL Server, Microsoft SharePoint, dan aplikasi .NET kustom. Selain itu, FSx untuk Windows File Server membantu Anda mengelola biaya dengan menyediakan opsi harga yang fleksibel, seperti pay-as-you-go dan kuota penyimpanan, dan deduplikasi data otomatis untuk mengurangi jejak penyimpanan dan mengoptimalkan kinerja dan biaya.

Bagian ini mencakup topik-topik berikut:

- Pilih penyimpanan file SMB yang tepat
- Aktifkan deduplikasi data di Amazon FSx
- Memahami sharding data FSx untuk Windows File Server
- Memahami penggunaan volume HDD di Amazon FSx
- Gunakan satu Availability Zone

Pilih penyimpanan file SMB yang tepat

Ikhtisar

AWS menawarkan berbagai layanan penyimpanan yang dikelola sepenuhnya yang memberi Anda kemampuan yang kaya dari layanan file terkemuka di industri, sambil menggabungkan inovasi dan keamanan AWS infrastruktur terbaru. Anda dapat menggabungkan AWS layanan ke dalam alur kerja infrastruktur sebagai kode (IAc) dan mengintegrasikannya dengan layanan AWS komputasi, pemantauan, dan perlindungan data. Untuk beban kerja Windows, Anda dapat memilih dari dua layanan file yang dikelola sepenuhnya yang dapat digunakan untuk mencocokkan kebutuhan aplikasi Anda: FSx untuk Windows File Server dan Amazon FSx untuk NetApp ONTAP.

FSx untuk Windows File Server

Amazon FSx untuk Windows File Server menyediakan penyimpanan bersama yang dikelola sepenuhnya yang dibangun di Windows Server, dan memberikan berbagai akses data, manajemen data, dan kemampuan administratif. FSx untuk Windows File Server terintegrasi dengan mudah

dengan lingkungan Windows karena ini adalah layanan asli Windows. Sebaiknya gunakan FSx Windows File Server untuk berbagi pengguna dan grup, Instans Kluster Selalu Aktif untuk SQL Server, aplikasi Windows, dan infrastruktur desktop virtual (VDI). FSx untuk Windows File Server juga terintegrasi dengan baik dengan Amazon FSx File Gateway, Amazon Kendra, log audit untuk Amazon S3, dan Amazon Data Firehose.

FSx untuk ONTAP

FSx untuk ONTAP didasarkan pada sistem NetApp file ONTAP milik. Dibutuhkan beberapa tingkat peningkatan keterampilan dan direkomendasikan sebagian besar untuk pengguna lokal NetApp yang ada. Kasus penggunaan umum termasuk berbagi pengguna dan grup, Instans Kluster Selalu Aktif untuk SQL Server, dan aplikasi Windows. FSx untuk ONTAP mendukung beberapa protokol, lebih besar dari sistem file 64 TB (skala PB tanpa server namespace DFS), kloning, replikasi, snapshot, kompresi (efisiensi penyimpanan), dan tingkatan data yang cerdas.

Dampak biaya

FSx untuk Windows File Server

FSx untuk Windows File Server adalah solusi penyimpanan bersama pertama AWS untuk menerapkan Instans Cluster Failover untuk SQL Server. Dengan FSx Windows File Server, Anda dapat meluncurkan Instans Failover Cluster menggunakan lisensi edisi SQL Standard. Namun, ini mencegah Anda mengandalkan grup ketersediaan Always On, yang memerlukan lisensi edisi SQL Server Enterprise. Dengan beralih dari edisi SQL Server Enterprise Standard ke edisi SQL Server Standard, Anda dapat menghemat 65-75 persen pada lisensi SQL Server Anda.

Anda dapat menggunakan FSx for Windows File Server untuk Instans Failover Cluster untuk membongkar I/O penyimpanan dari penyimpanan EBS biasa. Dengan membongkar I/O ke FSx for Windows File Server, Anda dapat mengurangi instans EC2, yang mengandalkan throughput Amazon EBS dan IOPS yang tinggi, tanpa memengaruhi throughput penyimpanan.

FSx untuk ONTAP

Anda dapat menggunakan FSx untuk ONTAP untuk menjalankan cluster failover Microsoft Anda pada protokol blok iSCSI dan mendapatkan manfaat dari inisialisasi file instan SQL Server, penggunaan replikasi lintas wilayah, dukungan antivirus, dan kloning. SnapMirror Jika Anda membuat beberapa salinan database untuk pengujian, kloning dapat membuat perbedaan yang signifikan dalam konsumsi ruang dan seberapa cepat salinan database tersebut dapat dibuat. Selain itu, Anda dapat menggunakan NetApp SnapCenter untuk mengelola fungsionalitas pencadangan, pemulihan, dan kloning dengan instans EC2 Anda untuk SQL Server dengan menggunakan untuk ONTAP. FSx

FSx untuk ONTAP juga menyediakan tiering otomatis dari SSD ke penyimpanan kolam berkapasitas rendah untuk campuran kinerja dan efisiensi biaya.

FSx untuk ONTAP mendukung NetApp sistem file (ONTAP), tidak seperti FSx untuk Windows File Server, yang mendukung sistem file NTFS asli Windows. Ukuran minimum FSx untuk ONTAP adalah 1024 GB, sedangkan FSx untuk Windows File Server dapat mulai serendah 32 GB.

Integrasi dengan Microsoft Distributed File System

FSx untuk Windows File Server dan FSx untuk ONTAP terintegrasi dengan Microsoft Distributed File System (DFS) untuk integrasi tanpa batas ke dalam penerapan yang ada. Ingatlah hal-hal berikut saat merencanakan arsitektur Anda:

- FSx untuk Windows File Server dan FSx untuk ONTAP mendukung DFS Namespaces (DFSN) pada kedua jenis penerapan (beberapa Availability Zone dan Availability Zone tunggal).
- Hanya FSx untuk Windows File Server mendukung Replikasi DFS (DFSR), dan hanya bila menggunakan Availability Zone tunggal.

Rekomendasi optimisasi biaya

Kinerja FSx untuk Windows File Server dan FSx untuk ONTAP sangat bergantung pada konfigurasi, seperti harganya. FSx untuk Windows File Server harga terutama tergantung pada kapasitas penyimpanan dan jenis penyimpanan, kapasitas throughput, cadangan, dan data yang ditransfer. Dengan FSx ONTAP, Anda membayar untuk penyimpanan SSD, IOPS SSD, penggunaan kolam kapasitas, kapasitas throughput, dan cadangan.

Layanan berkas	Biaya untuk penyimpanan 5 TB	Konfigurasi	Region
FSx untuk Windows File Server	$982,78	Zona Ketersediaan Tunggal SSD (15.000 IOPS) 32 MBps Pencadangan 5 TB (tidak ada penghematan deduplikasi)	AS Timur (Virginia Utara)

Layanan berkas	Biaya untuk penyimpanan 5 TB	Konfigurasi	Region
FSx untuk ONTAP	$979,28	Zona Ketersediaan Tunggal 100% SSD 15.000 tingkat kapasitas baca-tulis 15.000 SSD IOPS 128 MBps Pencadangan 5 TB (tidak ada penghematan deduplikasi)	AS Timur (Virginia Utara)

Ingatlah hal berikut:

- Deduplikasi dan kompresi memungkinkan Anda untuk menyimpan lebih banyak data pada perangkat fisik dengan mengecilkan ukuran data, tetapi Anda membayar untuk penyimpanan solid state drive (SSD) atau hard disk drive (HDD) yang disediakan.
- Anda dapat menggunakan FSx ONTAP untuk meningkatkan data Anda. Sangat jarang 100 persen data Anda diakses secara teratur dan membutuhkan penyimpanan SSD. Anda dapat memindahkan data dingin dan jarang diakses ke tingkat kapasitas untuk penghematan biaya.
- Harga yang disebutkan di sini dihitung dengan data 100 persen pada tingkat SSD dan 15.000 IOPS pada tingkat SSD.

Pencadangan

Secara default, baik FSx untuk ONTAP dan FSx untuk Windows File Server menyimpan cadangan yang dikelola sepenuhnya di Amazon S3. Namun, dengan FSx untuk ONTAP ada opsi tambahan untuk penggunaan cadangan SnapVault, yang dapat mengonfigurasi cadangan untuk berada di tingkat kapasitas. Pencadangan SnapVault adalah mekanisme yang dikelola sendiri yang lebih hemat biaya daripada opsi pencadangan default yang dikelola sepenuhnya. Opsi pencadangan yang

dikelola sepenuhnya adalah $0,05 per GB-bulan. SnapVault Cadangan FSx untuk ONTAP (SSD 10:1 ke penyimpanan kolam kapasitas) adalah $0.03221 (0.9x0.0219+0.1x0.125).

Ingatlah hal berikut:

- AWS backup terkelola menawarkan perincian satu jam. SnapVaultmemungkinkan Anda untuk pergi serendah lima menit.
- Anda dapat menggunakan NetApp alat (seperti CLI dan API) untuk mengonfigurasi SnapVault hubungan dan replikasi snapshot.
- Aktifkan kebijakan `all` tiering pada SnapVault volume untuk menggunakan tingkat kapasitas sebagai penyimpanan untuk data cadangan.
- SnapVault tujuan dapat berada di tempat yang sama Wilayah AWS, Lintas wilayah, atau lokal. Ini biasanya untuk satu Availability Zone atau beberapa tujuan cadangan sistem file Availability Zone. Sebagai perbandingan, AWS Backup didukung oleh ketahanan regional Amazon S3.

Ukuran yang tepat

Anda juga dapat menghemat biaya dan mendapatkan hasil maksimal dari sistem file Anda dengan ukuran yang tepat dan mencegah penyediaan berlebih.

Untuk ukuran yang tepat, lakukan hal berikut:

1. Identifikasi kebutuhan Anda saat ini berdasarkan data. Untuk beban kerja Windows yang khas, Anda dapat menggunakan alat sistem operasi bawaan seperti Performance Monitor.
2. Di Performance Monitor, gunakan penghitung berikut untuk mengukur kebutuhan kinerja Anda saat ini. Interval pengambilan diatur ke satu detik, dengan ukuran log maksimum 1.000 MB dan timpa diaktifkan.

```
Logman.exe create counter PerfLog-Short -o "c:\perflogs\PerfLog-Long.blg" -f bincirc
 -v mmddhhmm -max 1024 -c "\LogicalDisk(*)\*" "\Memory\*" "\.NET CLR Memory(*)\*"
 "\Cache\*" "\Network Interface(*)\*" "\Paging File(*)\*" "\PhysicalDisk(*)\*"
 "\Processor(*)\*" "\Processor Information(*)\*" "\Process(*)\*" "\Thread(*)\*"
 "\Redirector\*" "\Server\*" "\System\*" "\Server Work Queues(*)\*" "\Terminal
 Services\*" -si 00:00:01
```

3. Untuk memulai pengambilan log, jalankan `logman start PerfLog-Short` perintah. Untuk menghentikan pengambilan log, jalankan `logman stop PerfLog-Short` perintah.

> **Note**
>
> Anda dapat menemukan file log kinerja di c:\perflogs di server yang menjalankan tangkapan. Untuk informasi selengkapnya, lihat Ikhtisar Monitor Kinerja Windows di dokumentasi Microsoft.

4. Setelah Anda mengidentifikasi konfigurasi yang benar, uji apakah perkiraan Anda benar pada sistem FSx file Amazon dengan menggunakan alat stres disk seperti Microsoft DISKSPD.
5. Jika Anda puas dengan kinerjanya, potong ke berbagi file.

Kami merekomendasikan pendekatan konservatif untuk kapasitas penyimpanan karena hanya dapat ditingkatkan. Kapasitas throughput dapat ditingkatkan dan diturunkan sesuai kebutuhan.

Sumber daya tambahan

- Amazon FSx untuk NetApp ONTAP FAQs (AWS situs web)
- Mengoptimalkan kinerja Amazon FSx untuk Windows File Server dengan metrik baru (BlogAWS Penyimpanan)

Aktifkan deduplikasi data di Amazon FSx

Ikhtisar

Deduplikasi data adalah fitur yang memungkinkan Anda menyimpan data Anda lebih efisien dan dengan persyaratan kapasitas yang lebih sedikit. Ini melibatkan menemukan dan menghapus duplikasi dalam data tanpa mengorbankan kesetiaan atau integritasnya. Deduplikasi data menggunakan chunking dan kompresi ukuran variabel subfile, yang memberikan rasio optimasi 2:1 untuk server file umum dan hingga 20:1 untuk data virtualisasi. Deduplikasi data jauh lebih efektif daripada kompresi NTFS. Inheren dalam arsitektur deduplikasi adalah ketahanan selama kegagalan perangkat keras—dengan validasi checksum penuh pada data dan metadata, termasuk redundansi untuk metadata dan potongan data yang paling banyak diakses.

FSx untuk Windows File Server sepenuhnya mendukung deduplikasi data. Menggunakannya dapat menghasilkan penghematan rata-rata 50-60% untuk berbagi file tujuan umum. Dalam saham, penghematan berkisar antara 30-50% untuk dokumen pengguna dan hingga 70-80% untuk kumpulan data pengembangan perangkat lunak. Penting untuk dipahami bahwa penghematan penyimpanan

yang dapat Anda capai dengan deduplikasi data bergantung pada sifat kumpulan data Anda, termasuk berapa banyak duplikasi yang ada di seluruh file. Deduplikasi bukanlah pilihan yang baik jika data yang disimpan bersifat dinamis.

Dampak biaya

Untuk mengatasi pertumbuhan penyimpanan data di perusahaan, administrator mengkonsolidasikan server dan menjadikan penskalaan kapasitas dan pengoptimalan data tujuan utama. Pengaturan default deduplikasi data dapat memberikan penghematan segera, atau administrator dapat menyempurnakan pengaturan untuk melihat keuntungan tambahan. Misalnya, Anda dapat mengonfigurasi deduplikasi untuk dijalankan hanya pada jenis file tertentu, atau Anda dapat membuat jadwal pekerjaan khusus.

Pada tingkat tinggi, deduplikasi memiliki tiga jenis pekerjaan: optimasi, pengumpulan sampah, dan scrubbing. Ketahuilah bahwa ruang tidak akan dibebaskan sampai Anda menjalankan pekerjaan pengumpulan sampah setelah pengoptimalan. Anda dapat menjadwalkan pekerjaan atau Anda dapat menjalankannya secara manual. Semua pengaturan yang tersedia saat Anda menjadwalkan pekerjaan deduplikasi data juga tersedia saat Anda memulai pekerjaan secara manual (kecuali yang khusus penjadwalan).

Bahkan dengan penghematan efektif 25 persen dari deduplikasi, ada penghematan biaya yang signifikan FSx untuk Windows File Server. Penghematan yang diproyeksikan ini didasarkan pada perkiraan dalam. AWS Kalkulator Harga

Rekomendasi optimisasi biaya

Deduplikasi pada FSx untuk sistem file Windows File Server tidak diaktifkan secara default. Untuk mengaktifkan deduplikasi dengan menggunakan manajemen jarak jauh PowerShell, Anda harus menjalankan `Enable-FSxDedup` perintah dan kemudian menggunakan `Set-FSxDedupConfiguration` perintah untuk mengatur konfigurasi. Untuk informasi selengkapnya, lihat Mengelola sistem file dalam dokumentasi FSx untuk Windows File Server.

Untuk mengaktifkan deduplikasi, jalankan perintah berikut:

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

Untuk memverifikasi konfigurasi deduplikasi Anda, jalankan perintah berikut:

```
Invoke-Command -ComputerName amznfsxzzzzzzzz.corp.example.com -ConfigurationName
 FSxRemoteAdmin -ScriptBlock {
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days
 Mon,Tues,Wed,Sat -Start 09:00 -DurationHours 7
}
```

Dengan menjalankan PowerShell `Measure-DedupFileMetadata` cmdlet, Anda dapat menentukan berapa banyak ruang disk potensial yang dapat direklamasi pada volume jika Anda menghapus sekelompok folder, satu folder, atau satu file, dan kemudian menjalankan pekerjaan pengumpulan sampah. Secara khusus, `DedupDistinctSize` nilainya memberi tahu Anda berapa banyak ruang yang Anda dapatkan kembali jika Anda menghapus file-file itu. File sering memiliki potongan yang dibagikan di folder lain, sehingga mesin deduplikasi menghitung potongan mana yang unik dan akan dihapus setelah pekerjaan pengumpulan sampah.

Jadwal pekerjaan deduplikasi data default dirancang untuk bekerja dengan baik untuk beban kerja yang direkomendasikan dan tidak mengganggu mungkin (tidak termasuk pekerjaan pengoptimalan prioritas yang diaktifkan untuk jenis penggunaan cadangan). Jika beban kerja memiliki persyaratan sumber daya yang besar, sebaiknya Anda menjadwalkan pekerjaan yang dijalankan hanya selama jam idle, atau untuk mengurangi atau menambah jumlah sumber daya sistem yang diizinkan untuk dikonsumsi oleh pekerjaan deduplikasi data.

Secara default, deduplikasi data menggunakan 25 persen dari memori yang tersedia. Namun, ini dapat ditingkatkan dengan menggunakan`-memory switch`. Untuk pekerjaan pengoptimalan, kami menyarankan Anda menetapkan rentang dari 15 hingga 50. Untuk pekerjaan terjadwal, Anda dapat menggunakan konsumsi memori yang lebih tinggi. Misalnya, dengan pekerjaan pengumpulan sampah dan penggosokan (yang biasanya Anda jadwalkan untuk dijalankan di luar jam kerja), Anda dapat mengatur konsumsi memori yang lebih tinggi (seperti 50).

Untuk informasi tambahan mengenai pengaturan deduplikasi data, lihat Mengurangi biaya penyimpanan dengan Deduplikasi Data dalam dokumentasi FSx untuk Windows File Server.

Sumber daya tambahan

- Memahami Deduplikasi Data (dokumentasi Microsoft)
- Mengurangi biaya penyimpanan dengan Data Deduplication (FSx untuk dokumentasi Windows File Server)

Memahami sharding data FSx untuk Windows File Server

Ikhtisar

FSx untuk Windows File Server kinerja tergantung konfigurasi. Ini terutama didasarkan pada jenis penyimpanan, kapasitas penyimpanan, dan konfigurasi throughput. Kapasitas throughput yang Anda pilih menentukan sumber daya kinerja yang tersedia untuk server file — termasuk I/O batas jaringan, CPU dan memori, dan I/O batas disk yang diberlakukan oleh server file. Kapasitas penyimpanan dan jenis penyimpanan yang Anda pilih menentukan sumber daya kinerja yang tersedia untuk volume I/O penyimpanan—batas disk yang diberlakukan oleh disk penyimpanan. Selain kinerja, pilihan konfigurasi juga mempengaruhi biaya. FSx untuk Windows File Server harga terutama tergantung pada kapasitas penyimpanan dan jenis penyimpanan, kapasitas throughput, cadangan, dan data yang ditransfer.

Jika Anda memiliki penyimpanan file dan persyaratan kinerja yang relatif besar, Anda bisa mendapatkan keuntungan dari sharding data. Data sharding melibatkan membagi data file Anda menjadi kumpulan data yang lebih kecil (pecahan) dan menyimpannya di berbagai sistem file. Aplikasi yang mengakses data Anda dari beberapa instans dapat mencapai tingkat performa yang tinggi dengan membaca dan menulis ke serpihan ini secara paralel. Pada saat yang sama, Anda masih dapat menyajikan tampilan terpadu di bawah namespace umum untuk aplikasi Anda. Selain itu, ini juga dapat membantu untuk menskalakan penyimpanan data file di luar apa yang didukung oleh setiap sistem file (64 TB) untuk kumpulan data file besar — hingga ratusan petabyte.

Dampak biaya

Untuk kumpulan data besar, biasanya lebih efektif untuk menyebarkan beberapa kecil FSx untuk sistem file Windows File Server, daripada satu pangsa SSD besar untuk mencapai tingkat kinerja yang sama. Menggunakan kombinasi FSx untuk Windows File Server HDD dan jenis penyimpanan SSD memungkinkan penghematan biaya yang lebih baik, dan memungkinkan Anda untuk mencocokkan beban kerja dengan subsistem disk dasar terbaik. Dalam tabel berikut, Anda dapat melihat perbedaan antara satu sistem file 17 TB dan membandingkannya dengan beberapa sistem file yang lebih kecil yang menambah kapasitas yang sama.

Sistem file SSD besar dengan banyak beban kerja

Nama Server	Biaya	Konfigurasi	Region
Amazon FSx untuk Server File Windows	$5,716 USD	17 TB SSD	AS Timur (Virginia Utara)

Nama Server	Biaya	Konfigurasi	Region
		30 persen deduplikasi 256 Mbps Cadangan 17 TB	

Beban kerja yang dipartisi menggunakan DFSN

Nama Server	Biaya	Konfigurasi	Region	Bagikan
Amazon FSx untuk Server File Windows	$1,024 USD	2 TB SSD 20% deduplikasi 128 Mbps Cadangan 2 TB Multi-AZ	AS Timur (Virginia Utara)	Bagikan 1
Amazon FSx untuk Server File Windows	$2,132 USD	5 TB SSD 30% deduplikasi 256 Mbps Cadangan 5 TB Multi-AZ	AS Timur (Virginia Utara)	Bagikan 2
Amazon FSx untuk Server File Windows	$1,036 USD	10 TB HDD 40% deduplikasi 128 Mbps Cadangan 10 TB Multi-AZ	AS Timur (Virginia Utara)	Bagikan 3

Nama Server	Biaya	Konfigurasi	Region	Bagikan
Instans DFSN Windows EC2	$27 USD	t3a.medium 2 v CPUs Memori 4 GiB	AS Timur (Virginia Utara)	Instans DFSN

Biaya tahunan untuk sistem file SSD besar adalah $68.592. Biaya tahunan beban kerja yang dipartisi adalah $50.640. Dalam contoh ini, penghematan 26 persen dapat dicapai sambil mencocokkan beban kerja dengan penyimpanan backend yang sesuai. Untuk informasi selengkapnya tentang estimasi harga, lihat AWS Kalkulator Hargaestimasi.

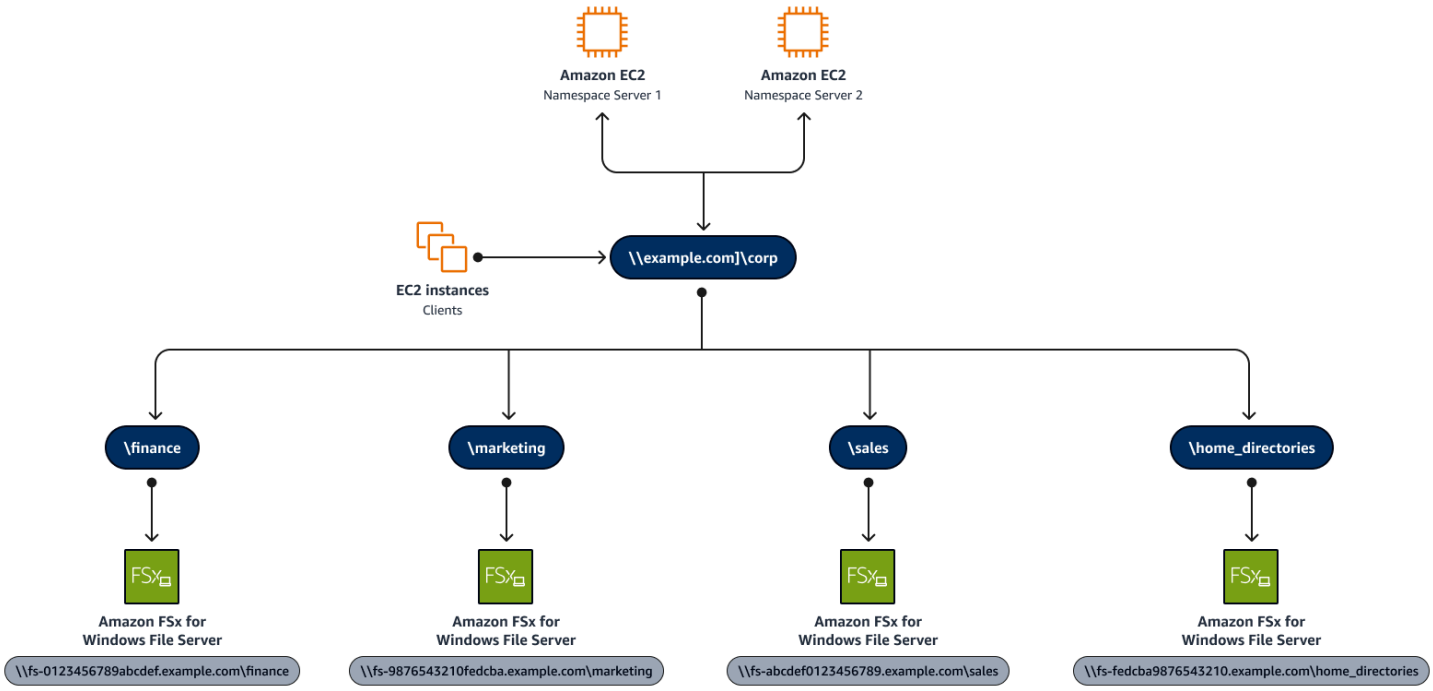
Rekomendasi optimisasi biaya

Untuk menerapkan solusi deduplikasi data, Anda harus menyiapkan Microsoft DFS Namespace berdasarkan jenis data, ukuran, I/O dan pola akses. I/O Setiap namespace mendukung hingga 50.000 berbagi file dan ratusan petabyte kapasitas penyimpanan secara agregat.

Ini bekerja paling efisien untuk memilih konvensi sharding yang mendistribusikan I/O secara merata di semua sistem file yang Anda rencanakan untuk digunakan. Memantau beban kerja Anda akan membantu dengan pengoptimalan tambahan atau pengurangan biaya. Jika Anda memerlukan bantuan untuk mengukur informasi kinerja untuk sistem FSx file Amazon, lihat FSx performa Windows File Server dalam dokumentasi FSx untuk Windows File Server.

Setelah Anda memilih strategi sharding, Anda dapat mengelompokkan sistem file untuk memudahkan akses ke saham Anda dengan menggunakan DFS Namespaces. Ini memungkinkan pengguna untuk melihat satu sistem file homogen, padahal pada kenyataannya mereka mengakses berbagai sistem file yang berbeda dengan kasus penggunaan yang dibuat khusus. Penting untuk membuat saham dengan konvensi penamaan yang tepat sehingga pengguna akhir Anda dapat dengan mudah menguraikan beban kerja apa yang dirancang untuk saham tersebut. Penting juga untuk memberi label produksi dan saham non-produksi, sehingga pengguna akhir tidak menempatkan file di sistem file yang salah karena kesalahan.

Diagram berikut menunjukkan bagaimana satu DFS Namespace dapat digunakan sebagai titik akses untuk beberapa sistem file Amazon FSx .



Ingatlah hal berikut:

- Anda dapat menambahkan yang ada FSx untuk berbagi Windows File Server ke pohon DFS.
- Amazon tidak FSx dapat ditambahkan ke root jalur berbagi DFS. Anda hanya memiliki satu subfolder.
- Anda harus menerapkan instans EC2 untuk melayani konfigurasi namespace DFS.

Untuk informasi selengkapnya tentang konfigurasi DFS-N, lihat ikhtisar Ruang Nama DFS di dokumentasi Microsoft. Untuk informasi selengkapnya tentang penggunaan ruang nama DFS, lihat video Menggunakan Ruang Nama DFS dengan Amazon FSx untuk Windows File Server aktif. YouTube

Sumber daya tambahan

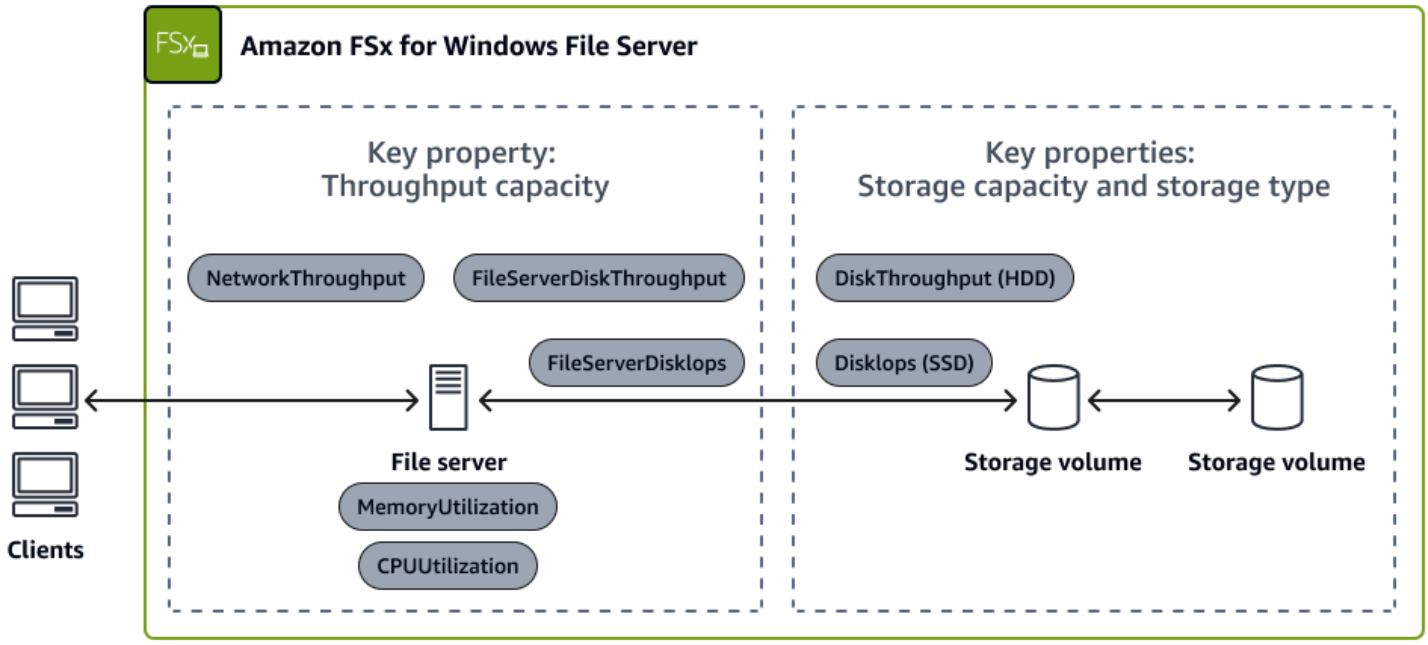
- Mengelompokkan beberapa sistem file dengan Ruang Nama DFS (dokumentasi Amazon) FSx
- Walkthrough 6: Menskalakan kinerja dengan pecahan (dokumentasi Amazon) FSx
- Menggunakan Ruang Nama DFS dengan Amazon FSx untuk Windows File Server (Labs)AWS

Memahami penggunaan volume HDD di Amazon FSx

Ikhtisar

Amazon FSx untuk Windows File Server menawarkan fleksibilitas untuk memilih throughput secara independen dari kapasitas sistem file. Tersedia dua pengaturan kapasitas: hard disk drive (HDD) dan solid state drive (SSD).

Diagram berikut menunjukkan hubungan antara throughput dan pengaturan penyimpanan.



Dengan penyimpanan berbasis HDD, Anda menerima baseline 12 IOPS dengan 80 burst disk IOPS (per IOPs TiB penyimpanan) dan throughput 12 baseline Megabytes/second dengan 80 burst (per TiB penyimpanan). Megabytes/second Misalnya, jika bagian Anda berukuran 50 TB, Anda mendapatkan 50 * 12 = 600 sebagai dasar untuk throughput dan IOPS.

Amazon FSx untuk Windows File Server menyediakan 80 burst IOPS. Kredit burst diisi ulang secara otomatis ketika pemanfaatan Anda di bawah tingkat dasar Anda dan secara otomatis dikonsumsi ketika pemanfaatan Anda di atas tingkat dasar Anda. Misalnya, jika beban kerja Anda hanya menggunakan 10 IOPS/TB selama satu jam (2 IOPS/TB below your baseline rate), you can then utilize 14 IOPS/TB (2 IOPS/TB di atas garis dasar Anda) selama satu jam berikutnya sebelum kehabisan kredit burst lagi.

Untuk operasi file, Amazon FSx untuk Windows File Server menyediakan latensi sub-milidetik yang konsisten dengan penyimpanan SSD dan latensi milidetik satu digit dengan penyimpanan

HDD. Untuk semua sistem file, termasuk yang memiliki penyimpanan HDD, Amazon FSx untuk Windows File Server menyediakan cache cepat (dalam memori) pada server file, sehingga Anda bisa mendapatkan kinerja tinggi dan latensi sub-milidetik untuk data yang diakses secara aktif, terlepas dari jenis penyimpanan.

Jika perlu, penggunaan penyimpanan HDD dapat membantu mengurangi biaya kapasitas penyimpanan Anda secara keseluruhan dan menyediakan platform penyimpanan yang andal untuk kebutuhan Anda.

Dampak biaya

Amazon FSx untuk Windows File Server kinerja tergantung pada tiga faktor: kapasitas penyimpanan, jenis penyimpanan, dan throughput. I/O Kinerja jaringan dan ukuran cache dalam memori hanya ditentukan oleh kapasitas throughput, sedangkan I/O kinerja disk ditentukan oleh kombinasi kapasitas throughput, jenis penyimpanan, dan kapasitas penyimpanan.

Meskipun SSD direkomendasikan untuk beban kerja I/O intensif, ada berbagai beban kerja yang kebutuhannya dapat dipenuhi dengan spesifikasi kinerja HDD. Penyimpanan HDD dirancang untuk spektrum beban kerja yang luas, termasuk direktori beranda, pembagian file pengguna dan departemen, serta sistem pengelolaan konten. Misalnya, jika pengguna Anda hanya memerlukan akses latensi rendah ke data yang mendukung proyek saat ini, maka sebagian besar data yang Anda simpan jarang diakses.

Anda dapat menggunakan AWS Kalkulator Hargauntuk memberikan perbandingan SSD 20 TB ke sistem file HDD di`us-east-1`. Seperti yang ditunjukkan tabel berikut, bahkan tanpa penghematan deduplikasi, perbedaan biaya signifikan ketika membandingkan sistem file HDD dengan sistem file SSD.

Konfigurasi sistem FSx file Amazon	Biaya bulanan
SSD Multi-AZ 20 TB () `us-east-1`	$4.699,30
HDD Multi-AZ 20 TB () `us-east-1`	$542,88
Perkiraan penghematan bulanan	$4,156.42

> **Note**
>
> Untuk tambahan FSx untuk penghematan Windows File Server, lihat FSx bagian Aktifkan deduplikasi data di Amazon pada panduan ini.

Dengan mengidentifikasi kebutuhan kinerja Anda dengan benar, Anda dapat memilih penyimpanan yang tepat untuk beban kerja Anda dan mengurangi biaya Anda.

Rekomendasi optimisasi biaya

Jika Anda memutuskan untuk menggunakan penyimpanan HDD, uji sistem file Anda untuk memastikannya dapat memenuhi persyaratan kinerja Anda. Penyimpanan HDD datang dengan biaya yang lebih rendah dibandingkan dengan penyimpanan SSD, tetapi dengan tingkat throughput disk dan IOPS disk yang lebih rendah per unit penyimpanan. Ini mungkin cocok untuk berbagi pengguna tujuan umum dan direktori rumah dengan I/O persyaratan rendah, sistem manajemen konten besar di mana data jarang diambil, atau kumpulan data dengan sejumlah kecil file besar.

Jenis penyimpanan untuk sistem file yang ada tidak dapat diubah. Untuk mengonversi jenis penyimpanan untuk sistem file Amazon FSx untuk Windows File Server, Anda harus mencadangkan sistem file yang ada dan mengembalikannya ke sistem file baru dengan jenis penyimpanan yang diinginkan. Jika Anda ingin mengonversi sistem file SSD yang ada ke sistem file HDD, ketahuilah bahwa HDD memiliki kapasitas minimum 2 TB yang jauh lebih tinggi.

Untuk memulihkan cadangan dengan jenis penyimpanan yang berbeda, lakukan hal berikut:

1. Cadangkan sistem file Anda yang ada.
2. Buat sistem FSx file Amazon baru dengan jenis penyimpanan HDD.
3. Kembalikan cadangan ke sistem file baru dengan jenis penyimpanan yang diinginkan.
4. Verifikasi sistem file baru memiliki jenis penyimpanan yang benar dan data Anda utuh.

Sebelum memindahkan perubahan Anda ke produksi, kami sarankan Anda menganalisis kinerja sistem FSx file Amazon Anda dan memverifikasi perubahan tersebut dapat diterima. Untuk panduan selengkapnya, lihat Mengoptimalkan kinerja Amazon FSx untuk Windows File Server dengan metrik baru di Blog AWS Penyimpanan.

Sumber daya tambahan

- Mengoptimalkan biaya dengan Amazon FSx (FSx dokumentasi Amazon)

Gunakan satu Availability Zone

Ikhtisar

Bagian ini menjelaskan kapan lebih bermanfaat untuk menggunakan implementasi Availability Zone tunggal Amazon FSx untuk Windows File Server. Ini mencakup skenario di mana pindah ke Availability Zone tunggal mengurangi biaya, sambil tetap memungkinkan Anda untuk menggunakan Amazon FSx untuk Windows File Server sebagai layanan penyimpanan file terkelola Anda. Kami menyarankan Anda menerapkan Availability Zone tunggal untuk Amazon FSx untuk beban kerja produksi. Ini dapat membantu memastikan bahwa Anda memiliki redundansi beberapa Availability Zone.

Dampak biaya

Sistem file Availability Zone tunggal menawarkan pengurangan biaya sekitar 40 persen dibandingkan dengan beberapa implementasi Availability Zone. Dengan beberapa sistem file Availability Zone, Anda membayar $0.230 per GB-bulan dalam SSD dan $0.025 per GB-bulan dalam HDD dibandingkan dengan $0.130 per GB-bulan untuk SSD dan $0.013 per GB-bulan untuk HDD pada sistem file Availability Zone tunggal. Anda dapat melihat perbandingan biaya dan membuat perkiraan Anda sendiri dengan menggunakan AWS Kalkulator Harga.

Untuk sistem file 10 TB ini bisa menjadi perbedaan membayar sekitar $1.200 per bulan untuk beberapa Availability Zone atau $680 per bulan untuk Availability Zone tunggal. Contoh ini menggunakan 10 TB FSx untuk sistem file Windows File Server dengan SSD. Perkiraan penghematan untuk deduplikasi adalah 50 persen. Secara keseluruhan, satu Availability Zone memiliki biaya masuk yang lebih rendah tetapi dilengkapi dengan beberapa peringatan yang tercakup dalam bagian berikutnya.

Rekomendasi optimisasi biaya

Penerapan Zona Ketersediaan Tunggal

Untuk memastikan bahwa satu Availability Zone cocok, pertimbangkan internal Anda sendiri SLAs untuk data yang disimpan FSx untuk Windows File Server. Ini memerlukan pemahaman jika Anda

harus memberikan SLAs kepada pelanggan Anda (internal dan eksternal) dan jika tiga sembilan ketersediaan untuk Zona Ketersediaan FSx tunggal Amazon masih memungkinkan Anda untuk bertemu dengan mereka. SLAs FSx untuk Windows File Server dengan Availability Zone tunggal masih memiliki uptime 99,9 persen. SLA untuk Amazon FSx untuk beberapa Availability Zone lebih besar dari 99,99 persen. Untuk beban kerja yang sangat penting, sebaiknya gunakan beberapa Availability Zone dalam satu Availability Zone, bahkan dengan biaya tambahan.

Penerapan Zona Ketersediaan Tunggal ideal untuk beban kerja seperti cadangan untuk database SQL Server. Mereka dapat menyediakan penyimpanan berbiaya rendah dengan tingkat HDD, sambil tetap memberi Anda waktu aktif yang konsisten. Jika Anda memerlukan tingkat ketersediaan yang lebih tinggi untuk beban kerja produksi, seperti server SQL yang sangat tersedia atau akses aplikasi produksi, maka satu Availability Zone tidak cocok untuk beban kerja Anda. Untuk lingkungan pencadangan, pengujian non-produksi, dan pengembangan, implementasi Zona Ketersediaan FSx tunggal Amazon dapat mengurangi biaya operasional Anda.

Salah satu kasus penggunaan di mana sistem file Zona Ketersediaan FSx tunggal Amazon berfungsi dengan baik adalah dalam situasi produksi di mana beberapa sistem file Zona Ketersediaan FSx tunggal Amazon digunakan, sebagai penyimpanan per server dalam kluster SQL Server yang sangat tersedia menggunakan grup ketersediaan Selalu Aktif. Untuk informasi selengkapnya, lihat Mengoptimalkan biaya untuk penerapan SQL Server ketersediaan tinggi Anda pada AWS posting di Blog Penyimpanan. AWS

Replikasi multi-Region

Opsi potensial untuk mengurangi biaya dengan satu sistem file Availability Zone (satu di mana hanya satu sistem file Availability Zone berfungsi) adalah jika Anda ingin memanfaatkan replikasi Multi-wilayah dengan Amazon. FSx Anda dapat menggunakan sistem file Single-AZ yang mendukung penggunaan dengan Microsoft DFS-R asli. DFS-R memiliki kemampuan untuk secara otomatis mereplikasi data di seluruh Wilayah dan beberapa situs. Untuk informasi selengkapnya tentang mengonfigurasi DFS-R menggunakan FSx Amazon, lihat Menggunakan Replikasi Sistem File Terdistribusi Microsoft di dokumentasi Amazon. FSx

Alternatif lain untuk penghematan biaya Multi-region adalah menggunakan AWS Storage Gateway. Ini memungkinkan Anda untuk menerapkan Amazon FSx File Gateway di Wilayah lain untuk akses Multi-wilayah Amazon FSx. Untuk informasi lebih lanjut, lihat AWS Storage Gatewaybagian panduan ini.

Jika Anda bekerja di seluruh Wilayah, Anda harus mempertimbangkan biaya transfer data untuk lalu lintas data lintas wilayah. Lalu lintas yang bergerak melintasi Wilayah dikenakan biaya $0,02/Gb.

Jadi, jika Anda memiliki perubahan data yang konsisten pada volume tinggi, ini akan menambah biaya keseluruhan Anda. Misalnya, 1 TB transfer data sama dengan sekitar $20,48.

Periode pemeliharaan

Jendela pemeliharaan adalah pertimbangan utama jika Anda menggunakan Zona Ketersediaan Tunggal dengan Amazon FSx. Selama jendela pemeliharaan, sistem FSx file Amazon tidak tersedia selama sekitar 20 menit, karena patching perangkat lunak rutin untuk Windows Server yang mendasarinya. Jika Anda menggunakan sistem file untuk pencadangan semalam, sesuaikan jendela FSx pemeliharaan Amazon yang sesuai untuk menghindari gangguan selama pencadangan Anda. Anda dapat menyesuaikan jendela pemeliharaan setelah membuat sistem FSx file Amazon Anda.

Sumber daya tambahan

- Ketersediaan dan daya tahan: Sistem file Single-AZ dan Multi-AZ (dokumentasi Amazon FSx)
- Amazon FSx untuk Windows File Server Harga (AWS situs web)

AWS Storage Gateway

AWS Storage Gateway adalah layanan penyimpanan cloud hybrid yang menghubungkan lingkungan lokal dengan penyimpanan AWS cloud. Ini memungkinkan Anda mengintegrasikan infrastruktur lokal yang ada dengan mulus AWS, memungkinkan Anda menyimpan dan mengambil data dari cloud dan menjalankan aplikasi di lingkungan hybrid. Untuk beban kerja Windows, Anda dapat menggunakan Storage Gateway untuk menyimpan dan mengakses data menggunakan protokol Windows asli seperti SMB dan NFS. Anda dapat menggunakan Storage Gateway untuk mengurangi biaya yang terkait dengan menjalankan beban kerja Windows AWS dengan menggunakan perangkat keras dan perangkat lunak lokal sebagai jembatan ke cloud. Hal ini memungkinkan Anda untuk mengambil keuntungan dari skalabilitas dan efisiensi biaya AWS tanpa harus membuat perubahan signifikan pada infrastruktur yang ada.

Di bawah payung Storage Gateway, Anda mendapatkan Amazon S3 File Gateway, Amazon FSx File Gateway, Tape Gateway, dan Volume Gateway. S3 File Gateway dan FSx File Gateway paling sering digunakan dengan beban kerja Microsoft.

Gateway File Amazon S3

Amazon S3 File Gateway memungkinkan Anda untuk menyimpan file Anda di Amazon S3 sambil memberikan akses ke pengguna Anda dengan menggunakan berbagi SMB tradisional.

Ini menyediakan antarmuka pengguna yang akrab dan membantu mengurangi biaya dengan menyimpan data Anda di Amazon S3 dan memanfaatkan berbagai tingkatan penyimpanan Amazon S3. Anda dapat menerapkan Storage Gateway dengan S3 Intelligent Tiering untuk membantu Anda memindahkan file siklus hidup secara otomatis ke tingkat penyimpanan dengan biaya terendah untuk menurunkan biaya lebih jauh. Kami merekomendasikan S3 File Gateway untuk scale-out, akses hanya-baca, pembacaan berulang yang cepat (dari cache), dan dump database. Umumnya tidak direkomendasikan untuk penulisan berkinerja tinggi atau ketersediaan tinggi, mengedit file, atau berbagi departemen.

Gerbang FSx File Amazon

Amazon FSx File Gateway juga dapat menawarkan penghematan biaya saat bekerja dengan sistem file Amazon FSx Windows. Anda dapat menjalankan FSx File Gateway untuk menyediakan akses lokal ke sistem FSx file Amazon di Wilayah lain untuk menghindari biaya memiliki dua sistem file independen. Ini juga dapat membantu jika Anda memiliki beberapa server file lokal dan ingin mengkonsolidasikannya untuk menghindari pembayaran untuk beberapa perangkat keras.

Dampak biaya

Gateway File Amazon S3

Menyiapkan S3 File Gateway mudah karena Anda dapat menggunakan wizard peluncuran untuk Storage Gateway. Anda dapat menerapkan gateway dalam hitungan menit dengan menggunakan instans EC2 di lingkungan Anda AWS . Setelah gateway diatur, Anda dapat mengonfigurasi share Storage Gateway agar dapat diakses melalui protokol SMB dan NFS. Untuk beban kerja Windows yang khas, Anda juga dapat menggunakan pengaturan ini untuk memanfaatkan lingkungan Direktori Aktif dan menetapkan izin pada pembagian file Anda. Anda dapat secara efektif mengintegrasikan Storage Gateway ke dalam penggunaan normal Anda, karena akan berfungsi sebagai berbagi file Windows biasa. File dan folder disimpan sebagai objek dan daftar kontrol akses NTFS (ACLs) sebagai metadata.

Tabel berikut membandingkan biaya penyimpanan 10 TB dengan tiga opsi penyimpanan yang tersedia:

- FSx untuk Windows File Server
- Gateway File Amazon S3
- Amazon Elastic Block Store (Amazon EBS)

Harga untuk menyimpan penyimpanan 10 TB jauh lebih murah jika Anda menggunakan Amazon S3, karena Anda dapat mempartisi data Anda ke berbagai tingkatan penggunaan. Dalam perkiraan harga, S3 Intelligent Tiering digunakan untuk fleksibilitas harga. Ini termasuk 80 persen dalam Standar S3, 10 persen di Akses Jarang, dan 10 persen di Amazon Glacier. Meskipun Anda dapat menggunakan Amazon Glacier, penting untuk menetapkan aturan siklus hidup yang tepat untuk memastikan bahwa file apa pun yang dipindahkan ke Amazon Glacier tidak perlu segera diakses. Amazon Glacier murni untuk penggunaan arsip, bukan penggunaan akses reguler.

Sistem penyimpanan	Biaya penyimpanan 10 TB	Region
FSx untuk Windows File Server (dengan asumsi penghematan 50% untuk deduplikasi)	$683.20 USD SSD	AS Timur (Virginia Utara)
Gateway File Amazon S3	$449.51 USD Tingkat Cerdas	AS Timur (Virginia Utara)
Amazon EBS	$1,335.69 USD GP3	AS Timur (Virginia Utara)

Pertimbangkan hal berikut:

- Di Amazon Glacier, Anda menerima kesalahan I/O umum kecuali Anda menggunakan RestoreObjectAPI untuk memulihkan objek kembali ke Amazon S3. Kami menyarankan Anda menggunakan pemberitahuan untuk I/O kesalahan ini dengan menggunakan Amazon CloudWatch Events. Dengan begitu, tim operasi Anda dapat bereaksi terhadap pengguna yang mendapatkan kesalahan ini pada file yang mungkin perlu mereka akses. Untuk informasi selengkapnya tentang kesalahan ini, lihat Kesalahan: InaccessibleStorageClass dalam dokumentasi Amazon S3 File Gateway.
- Selain pembatasan Amazon Glacier pada akses, hanya ada ACLs 10 yang diizinkan per objek/folder di Storage Gateway. Sebelum Anda memutuskan untuk menggunakan Storage Gateway, pastikan Anda tidak memerlukan lebih dari 10 entri ACL.

Gerbang FSx File Amazon

Mirip dengan Amazon S3 File Gateway, FSx File Gateway menyediakan akses ke sistem file yang menyimpan data jangka panjang. Di Amazon S3 File Gateway, data berada di Amazon S3. Untuk FSx File Gateway, data Anda berada di FSx untuk Windows File Server. Meskipun opsi Multi-AZ

tersedia FSx untuk Windows File Server, tidak ada opsi Multi-wilayah. Jika Anda memiliki perusahaan global atau kantor jarak jauh, Anda mungkin perlu menyediakan platform penyimpanan bersama yang secara geografis lebih dekat dengan pengguna akhir untuk menghindari latensi. Jika Anda menggunakan sistem FSx file Amazon lain, ini akan menambah biaya sistem file Amazon FSx untuk Windows File Server yang sama sekali baru dan penyimpanan yang diperlukan. Untuk menghindari pembuatan sistem file yang sama sekali baru dan menduplikasi biaya, Anda dapat menerapkan FSx File Gateway di Wilayah sekunder. Ini memberikan akses lokal ke file untuk pengguna, sambil membantu mengurangi biaya keseluruhan Anda.

Sistem penyimpanan	Biaya penyimpanan 10 TB	Region
Amazon FSx untuk Server File Windows	$683.20 USD SSD	AS Timur (Virginia Utara)
Gerbang FSx File Amazon	$503,70/Gerbang Tunggal	AS Timur (Virginia Utara)

> **Note**
>
> Harga di tabel sebelumnya didasarkan pada harga Storage Gateway.

Ingatlah hal berikut:

- FSx File Gateway dapat membantu Anda menghemat sekitar $180 per bulan (atau $2100 per tahun) untuk beban kerja Multi-wilayah.
- Biaya transfer data jauh lebih rendah dengan FSx File Gateway, karena hanya perlu menyimpan file yang diakses secara teratur dan bukan salinan sekunder lengkap.
- Meskipun Anda dapat memiliki dua penerapan FSx untuk Windows File Server di Wilayah yang berbeda dan terus memperbaruinya dengan AWS Backup atau AWS DataSync, tidak ada opsi yang mendekati waktu nyata.

Rekomendasi optimisasi biaya

Gateway File Amazon S3

S3 File Gateway menyediakan opsi berbiaya rendah untuk menyimpan file, tetapi ada beberapa masalah yang perlu dipertimbangkan mengenai bagaimana mengimplementasikan dan menggunakan sistem file. Misalnya, S3 File Gateway membutuhkan penggunaan mesin virtual untuk menjalankan perangkat lunak Storage Gateway. Di AWS, Storage Gateway di-deploy di Amazon EC2 dengan menggunakan instance m5.xlarge, secara default. Jika ingin mengurangi biaya penyimpanan lokal, Anda dapat menerapkan Storage Gateway sebagai alat virtual pada platform virtualisasi seperti VMware dan Hyper-V.

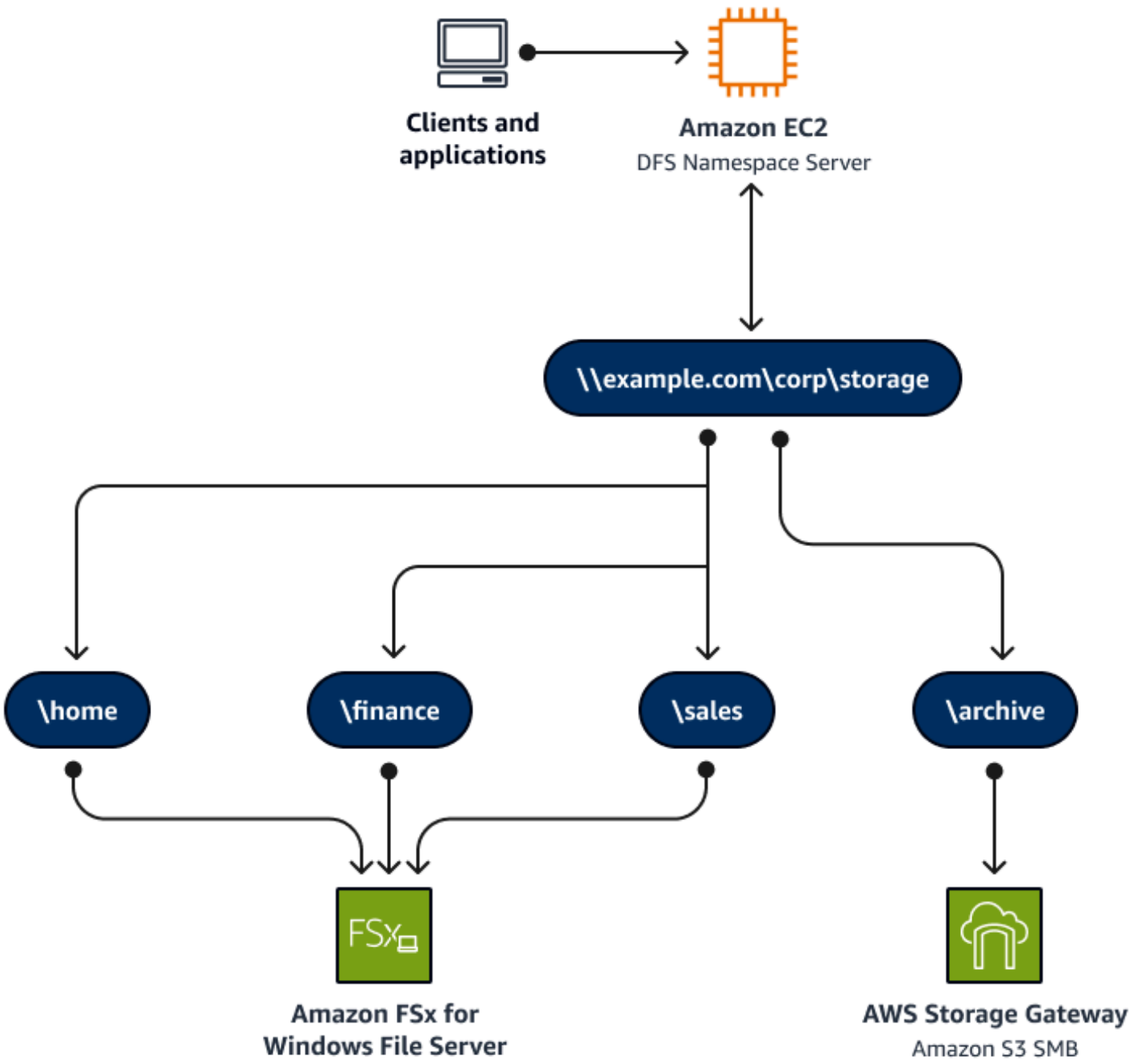
Pertimbangan ketersediaan tinggi

Running Storage Gateway adalah titik kegagalan tunggal untuk akses ke file. Untuk mencegah downtime yang tidak perlu, sebaiknya Anda menerapkan kontrol akses yang ketat di mana pengguna dapat membuat perubahan atau menghentikan dan memulai instance Storage Gateway. Selain itu, untuk penerapan AWS, ada baiknya menggunakan Amazon Data Lifecycle Manager untuk membuat snapshot perutean guna memulihkan implementasi Storage Gateway Anda dengan cepat. Jika Anda menjalankan Storage Gateway lokal menggunakan VMware, Anda dapat mengonfigurasinya untuk ketersediaan tinggi.

Menjalankan beberapa sistem file

Memisahkan beban kerja file yang digunakan sehari-hari dari beban kerja arsip dapat membantu Anda menghindari biaya penyimpanan yang tidak perlu. Storage Gateway memiliki kemampuan untuk digunakan bersama sistem file Windows File Server FSx untuk Windows. Dengan menggunakan Ruang Nama DFS, Anda dapat menampilkan penyimpanan penggunaan harian utama yang berjalan FSx untuk Windows File Server dan penyimpanan Anda berjalan di Amazon S3 (yang diakses melalui Storage Gateway).

Diagram berikut menunjukkan bagaimana satu DFS Namespace dapat digunakan sebagai titik akses frontend untuk opsi penyimpanan backend yang berbeda.



Klien diarahkan ke struktur folder, seperti\\ example.com\ storage. Direktori utama ini berisi sub-direktori. Sistem file FSx untuk Windows File Server berisi berbagi file yang diakses secara normal. Anda dapat menggunakan berbagi file yang dibuat di Storage Gateway untuk data arsip. Pengguna dapat mengarsipkan item secara manual ke folder arsip atau Anda dapat membuat proses untuk mengotomatiskan pemindahan beberapa file dari berbagi file normal ke folder arsip.

Pertimbangkan hal berikut:

- Tinjau persyaratan penyimpanan Anda dan berikan penyimpanan yang memadai untuk cache.
- Tambahkan gateway Anda ke konfigurasi Active Directory Anda dan gunakan Windows standar ACLs untuk akses ke file.

FSx Gerbang File

Penyebaran FSx File Gateway mirip dengan penyebaran S3 File Gateway, tetapi bahkan lebih mudah jika Anda menggunakan wizard peluncuran. Untuk petunjuk terperinci, lihat Langkah 3: Membuat dan mengaktifkan Amazon FSx File Gateway dalam dokumentasi Amazon FSx File Gateway. Setelah menerapkan FSx File Gateway di lingkungan Anda, Anda dapat mengaitkannya ke sistem FSx file Amazon yang ada dan mendapatkan akses ke file Anda.

Penyimpanan adalah pertimbangan utama saat menerapkan FSx File Gateway. Penyimpanan default menyediakan 150 GB, yang merupakan jumlah ruang yang layak untuk file caching. Membuat peringatan pemantauan untuk ruang kosong rendah dapat membantu penyimpanan dengan ukuran yang tepat tanpa alokasi berlebihan.

Sumber daya tambahan

- AWS Storage Gateway sumber daya (AWS dokumentasi)

Active Directory

Amazon Elastic Compute Cloud (Amazon EC2) yang menjalankan Windows Server adalah lingkungan yang aman, andal, dan berkinerja tinggi untuk menerapkan aplikasi dan beban kerja berbasis Windows. Anda dapat menyediakan instance dengan cepat dan meningkatkan atau menurunkan skala sesuai kebutuhan, sementara hanya membayar untuk apa yang Anda gunakan. Layanan Active Directory digunakan sebagai sumber utama manajemen identitas di lingkungan Windows Server.

Bagian ini mencakup topik-topik berikut:

- Direktori Aktif yang dikelola sendiri di Amazon EC2
- AWS Managed Microsoft AD
- AD Connector

Direktori Aktif yang dikelola sendiri di Amazon EC2

Ikhtisar

Bagian ini memberikan rekomendasi untuk mengurangi biaya menjalankan Active Directory di Amazon Elastic Compute Cloud (Amazon EC2). Fokus utamanya adalah memastikan bahwa Anda dapat mengukur pengontrol domain Active Directory dengan tepat dan menggunakan fleksibilitas AWS Cloud untuk menyesuaikan sesuai kebutuhan untuk lingkungan Anda. AWS dapat membantu Anda menghentikan instans dengan mudah dan mengubah ukurannya untuk memenuhi kebutuhan Anda yang berubah, atau mengurangi ukuran instance jika Anda meningkatkan skala terlalu cepat. Memilih ukuran dan jenis instans yang tepat dapat menghasilkan penghematan yang signifikan.

Dampak biaya

Tabel berikut menunjukkan perbedaan antara memilih instance keluarga instance burstable atas instance tujuan umum. Pilihan ini dapat menghemat banyak uang setiap bulan. Perencanaan dan ukuran instans yang tepat dapat membantu Anda mengelola biaya.

Tipe instans	Jumlah instans	vCPU	Memori	Biaya
t3a.medium	2	2	8	$81.76/bulan

Tipe instans	Jumlah instans	vCPU	Memori	Biaya
m5a.large	2	2	8	$259.88/bulan

Untuk informasi lebih lanjut tentang biaya, lihat AWS Kalkulator Harga perkiraan.

Penghematan $178,12 per bulan berakhir menjadi lebih dari $2.000 dalam penghematan per tahun untuk pengontrol domain Anda. Perlu diingat bahwa untuk jejak kecil hanya dua pengontrol domain dalam satu akun. Pada skala dengan beberapa akun dan pengontrol domain tambahan, penghematan tersebut dapat menambah hingga pengurangan biaya yang signifikan.

Rekomendasi optimisasi biaya

Microsoft menyediakan rekomendasi perencanaan kapasitas saat Anda menerapkan lingkungan Active Directory. Kami menyarankan Anda mempertimbangkan komponen utama berikut saat merencanakan atau menskalakan lingkungan Active Directory Anda:

- Memori
- Jaringan
- Penyimpanan
- Prosesor

Sambil mengingat komponen-komponen utama ini, Anda dapat bekerja dengan memilih jenis instans yang masuk akal untuk lingkungan Active Directory Anda AWS. Bagian ini mencakup beberapa contoh Active Directory untuk skenario AWS penerapan. Skenario ini memperjelas bahwa tidak perlu mereplikasi lingkungan lokal Anda AWS, jika Anda tidak berencana untuk menangani jumlah pengguna dan komputer yang sama seperti yang Anda lakukan di lingkungan lokal.

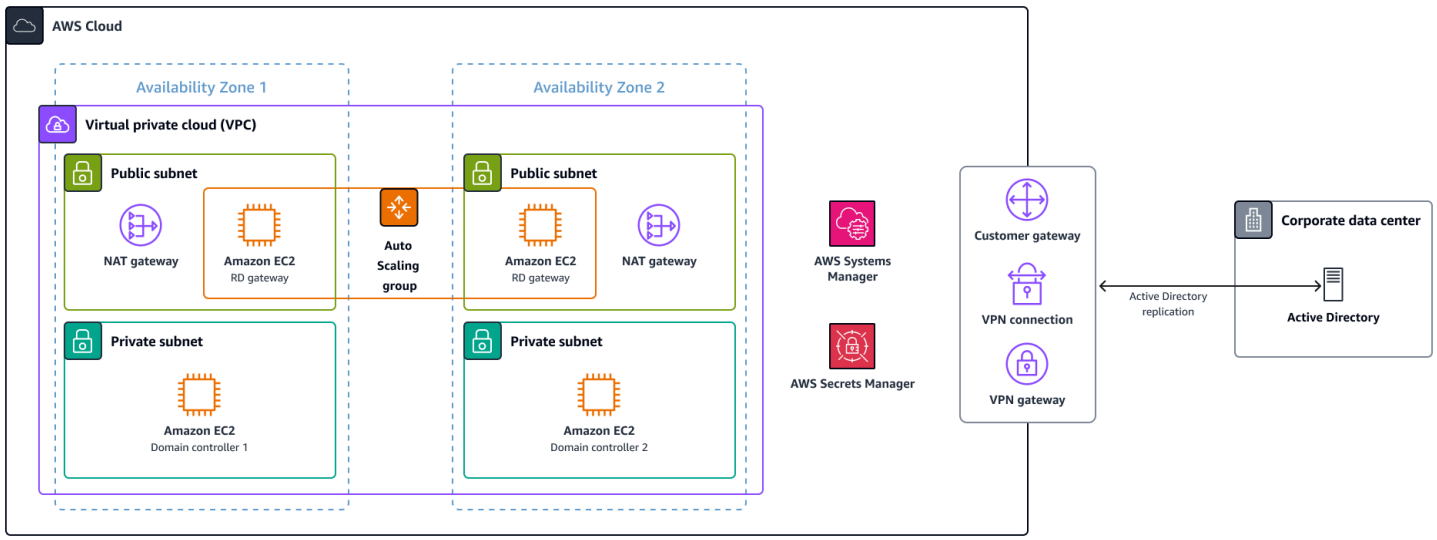
Tabel berikut menyoroti komponen penting mengenai vCPU, memori, dan disk untuk footprint Anda AWS .

Komponen	Estimasi
Ukuran penyimpanan/database	40—60 KB untuk setiap pengguna
RAM	Ukuran database

Komponen	Estimasi
	Rekomendasi sistem operasi dasar Aplikasi pihak ketiga
Jaringan	1 GB
CPU	1.000 pengguna bersamaan untuk setiap inti

Skenario penyebaran hibrida

Diagram berikut menunjukkan contoh arsitektur untuk penyebaran hybrid Active Directory.



Seperti yang ditunjukkan diagram, Anda biasanya memiliki jejak lokal dan kemudian memperluas ini ke dalam. AWS Cloud Pada fase awal migrasi, Anda biasanya tidak akan memiliki semua pengguna dan server Anda disebarkan. AWS Itulah mengapa penting untuk awalnya menggunakan jejak berukuran lebih kecil untuk menghemat uang pada upaya migrasi.

Jika Anda akan mempertahankan footprint lokal dengan server dan pengguna yang mengautentikasi di lokasi, maka Anda tidak memerlukan footprint yang sama untuk pengontrol domain. AWS Dengan mengikuti praktik terbaik Active Directory, Anda dapat menerapkan situs dan layanan Active Directory yang tepat untuk mengautentikasi pengguna dan komputer ke footprint lokal Anda, sementara hanya mengautentikasi AWS footprint Anda ke pengontrol domain di. AWS Hal ini memungkinkan Anda untuk menghindari melebih-lebihkan jejak Active Directory Anda AWS dengan membatasi penggunaan hanya AWS sumber daya dan tidak semua infrastruktur lokal Anda. Untuk panduan

merancang penyiapan hibrida, lihat Penempatan pengontrol domain dan pertimbangan situs yang tepat dalam dokumentasi Microsoft.

Optimalkan AWS migrasi dengan ukuran yang tepat

Jika Anda menerapkan instance baru Active Directory untuk pengguna Anda atau berencana untuk sepenuhnya bermigrasi ke AWS infrastruktur Direktori Aktif Anda, sebaiknya Anda merencanakan ukuran berdasarkan rekomendasi Microsoft untuk vCPU, memori, dan ruang disk untuk pilihan instans di tabel sebelumnya.

Jika ini adalah jejak baru, Anda dapat memulai dari yang kecil dan memanfaatkan kemampuan untuk dengan mudah mengubah jenis instance untuk mengubah ukuran lingkungan Anda saat ia tumbuh. AWS Bagian Windows di Amazon EC2 dari panduan ini menunjukkan kepada Anda cara memantau dan meninjau pemanfaatan CPU dan memori Anda. AWS Dengan begitu, Anda tahu kapan harus meningkatkan ukuran instans EC2 Anda.

Jika Anda sepenuhnya memigrasikan lingkungan Active Directory lokal AWS, Anda dapat menerapkan rencana ukuran yang sama untuk memastikan kinerja yang tepat. Sebelum menduplikasi apa yang Anda miliki di lokasi AWS, kami sarankan Anda menyelesaikan tinjauan menyeluruh terhadap lingkungan Direktori Aktif Anda. Ini dapat membantu Anda mencegah penyediaan berlebihan. Pastikan untuk menggunakan Performance Monitor untuk mengumpulkan informasi tentang jumlah lalu lintas dan pemanfaatan untuk pengontrol domain yang ada. Ini dapat memberi Anda pemahaman tentang penggunaan keseluruhan sehingga Anda dapat mengukur ukuran yang tepat dan pada akhirnya mengurangi biaya Anda.

Optimalkan Active Directory pada AWS

Jika Anda menjalankan Active Directory AWS, penting juga untuk terus memantau pemanfaatan dan mengubah ukuran instance sesuai kebutuhan untuk mengurangi pengeluaran Anda. Anda dapat menggunakan AWS Compute Optimizer untuk mendapatkan informasi tentang sumber daya yang Anda jalankan AWS. Untuk informasi tentang menggunakan Compute Optimizer untuk mengukur beban kerja Windows Anda dengan benar, lihat bagian Windows di Amazon EC2 dari panduan ini. Untuk penyelaman mendalam yang lebih komprehensif, Anda dapat menggunakan Performance Monitor untuk memantau pemanfaatan pengontrol domain Active Directory, menilai kinerja, dan kemudian mengubah ukuran yang sesuai.

Anda juga dapat menggunakan CloudWatch untuk memantau kinerja pengontrol domain. Untuk mengoptimalkan pengontrol domain Anda (meningkatkan atau menurunkan skala), Anda dapat menggunakan metrik yang tersedia CloudWatch untuk membantu Anda membuat keputusan

yang tepat. Anda dapat menggunakan CloudWatch agen untuk mengonfigurasi metrik Monitor Kinerja kustom yang akan dikirim untuk pengumpulan data. Untuk petunjuk, lihat Bagaimana cara menggunakan CloudWatch agen untuk melihat metrik Monitor Kinerja di server Windows? di pusat AWS pengetahuan.

Setelah menerapkan CloudWatch agen, Anda dapat mengonfigurasi metrik berikut dalam file konfigurasi agen di bawah: `metrics_collected`

Kategori metrik	Nama metrik
Database ke instance (NTDSA)	Database cache% tekan
Database I/O membaca latensi rata-rata	
I/O database reads/sec	
Log I/O menulis latensi rata-rata	
DirectoryServices (NTDS)	Waktu mengikat LDAP
DRA menunggu operasi replikasi	
DRA menunggu sinkronisasi replikasi	
DNS	Kueri rekursif/detik
Kegagalan kueri rekursif/detik	
Kueri TCP diterima/detik	
Total kueri yang diterima/detik	
Total respons yang dikirim/detik	
Kueri UDP diterima/detik	
LogicalDisk	Rata-rata panjang antrian disk
% ruang kosong	
Memori	% byte berkomitmen digunakan

Kategori metrik	Nama metrik
Masa pakai cache siaga rata-rata jangka panjang	
Antarmuka jaringan	Byte dikirim/detik
Byte Diterima/detik	
Bandwidth saat ini	
NTDS	ATQ memperkirakan penundaan antrian
Latensi permintaan ATQ	
Direktori DS membaca/detik	
Pencarian direktori DS/detik	
Direktori DS menulis/detik	
Sesi klien LDAP	
Pencarian LDAP/detik	
Ikatan sukses LDAP/detik	
Prosesor	% waktu prosesor
Statistik seluruh sistem keamanan	Otentikasi Kerberos
Otentikasi NTLM	

Sumber daya tambahan

- Layanan Domain Direktori Aktif di AWS: Panduan Penyebaran Solusi Mitra (AWS dokumentasi)
- Perencanaan kapasitas untuk Layanan Domain Direktori Aktif (dokumentasi Microsoft)
- Pertimbangan desain untuk menjalankan Active Directory pada instans EC2 (Whitepaper)AWS

AWS Managed Microsoft AD

Ikhtisar

AWS Directory Service for Microsoft Active Directory, juga dikenal sebagai AWS Managed Microsoft AD, didukung oleh Windows Server Active Directory dan dikelola oleh AWS. Anda dapat menggunakan AWS Managed Microsoft AD untuk memigrasikan berbagai aplikasi Active Directory —aware ke. AWS Cloud AWS Managed Microsoft AD bekerja dengan berbagai aplikasi dan layanan Active Directory asli. Ini juga mendukung aplikasi dan layanan yang AWS dikelola. Meskipun tidak banyak tuas pengoptimalan biaya AWS Managed Microsoft AD karena layanan dan mekanisme penagihannya, ada beberapa prinsip desain yang dapat membantu Anda meminimalkan biaya.

Dampak biaya

Karena AWS Managed Microsoft AD merupakan layanan terkelola berdasarkan saat ini SKUs, ukuran adalah proses yang relatif mudah. Saat ini ada dua ukuran yang SKUs tersedia: edisi Standar dan Perusahaan. Lainnya SKUs termasuk berbagi direktori, menambahkan pengontrol domain tambahan (termasuk Wilayah tambahan), dan transfer data lintas wilayah.

Rekomendasi optimisasi biaya

Ada perbedaan antara AWS Managed Microsoft AD Standard Edition dan AWS Managed Microsoft AD Enterprise Edition. Enterprise Edition mendukung hingga 500.000 objek Direktori Aktif, 500 pembagian akun (batas lunak), dan memiliki dukungan Multi-wilayah. Edisi Standar mendukung hingga 30.000 objek Direktori Aktif, lima pembagian akun (batas lunak hingga sekitar 25 maksimum), dan tidak memiliki dukungan Multi-wilayah.

> **Note**
>
> Batas atas objek Active Directory adalah perkiraan. Direktori Anda mungkin mendukung lebih banyak atau lebih sedikit objek tergantung pada ukuran dan perilaku serta kebutuhan kinerja aplikasi Anda.

Pertanyaan yang perlu dipertimbangkan sebelum memilih jenis direktori Anda adalah:

- Apakah dukungan Multi-wilayah diperlukan?
- Apakah direktori akan dibagikan dengan lebih dari 25 akun?

- Apakah jumlah objek Active Directory akan lebih dari 30.000?

Jika jawabannya ya untuk salah satu pertanyaan di atas, maka Enterprise Edition diperlukan. Jika jawaban untuk semua pertanyaan adalah tidak, kami sarankan Anda mulai dengan Edisi Standar.

> **Note**
>
> Anda dapat meng-upgrade direktori dari Standard Edition ke Enterprise Edition tetapi direktori tidak dapat diturunkan. Menerapkan Edisi Standar tidak melalui pintu satu arah. Jika Anda ingin meng-upgrade direktori Anda ke Enterprise Edition, hubungi AWS.

Ada biaya untuk setiap saham saat Anda berbagi direktori di AWS Managed Microsoft AD Enterprise Edition. Ini kurang dari biaya penyebaran direktori di setiap akun, tetapi perlu diingat bahwa biaya berbagi dapat merayap naik jika dibiarkan tidak dicentang. Kami menyarankan Anda hanya berbagi direktori dengan akun yang berisi Amazon Relational Database Service (Amazon RDS) dan FSx Amazon untuk Windows File Server, karena hanya layanan tersebut yang mendukung fitur ini. Perlu diingat bahwa Anda memiliki opsi FSx untuk mengintegrasikan Windows File Server dengan Active Directory yang dikelola sendiri, termasuk file AWS Managed Microsoft AD. Jika hanya Amazon yang FSx diperlukan di akun lain, maka Anda dapat melakukan FSx penyebaran Amazon yang dikelola sendiri terhadap AWS Managed Microsoft AD tanpa perlu membagikan direktori.

Saat memutuskan kapan akan menggunakan pengontrol domain tambahan, perlu diingat bahwa hanya AWS Managed Microsoft AD mendukung dua subnet di Availability Zone terpisah di VPC yang sama. Menambahkan pengontrol domain tambahan tidak memungkinkan Anda menambahkan subnet tambahan. Untuk menentukan apakah Anda harus menambahkan pengontrol domain tambahan karena masalah kinerja, tinjau metrik kinerja pengontrol domain di. CloudWatch Ini memberi tahu Anda jika satu atau semua pengontrol domain kewalahan. Jika Anda menentukan bahwa hanya satu pengontrol domain yang kewalahan, menambahkan pengontrol domain tambahan tidak akan mengurangi beban dan Anda harus menggali lebih dalam aplikasi yang tidak menyeimbangkan beban di seluruh pengontrol domain yang tersedia saat ini. Jika semua pengontrol domain sedang banyak digunakan, menambahkan pengontrol domain tambahan dapat mengurangi beban pada pengontrol domain yang ada. Untuk petunjuk tentang cara mengotomatiskan penskalaan, lihat Cara mengotomatiskan AWS Managed Microsoft AD penskalaan berdasarkan metrik pemanfaatan di Blog Keamanan. AWS

Jika Anda memperluas direktori Anda ke beberapa Wilayah, sebaiknya Anda tidak menggunakan direktori saham NETLOGON atau SYSVOL untuk penyimpanan file. Semua pengontrol domain

mereplikasi konten saham tersebut. Tidak menggunakan saham untuk penyimpanan file membuat biaya transfer data seminimal mungkin.

Anda juga memiliki opsi untuk mendaftar dalam Perjanjian Perusahaan dengan AWS. Perjanjian Perusahaan memberi Anda pilihan untuk menyesuaikan perjanjian yang paling sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat Pelanggan Perusahaan.

Sumber daya tambahan

- AWS Managed Microsoft AD kuota (AWS Directory Service dokumentasi)
- AWS Directory Service Harga (AWS situs web)
- Layanan Domain Direktori Aktif di AWS (AWS Whitepaper)

AD Connector

Ikhtisar

AD Connector adalah layanan proxy yang menyediakan cara mudah untuk menghubungkan Microsoft Active Directory lokal yang ada ke AWS aplikasi yang kompatibel, seperti Amazon, Amazon Quick WorkSpaces, dan domain join tanpa batas untuk instans Amazon Elastic Compute Cloud (Amazon EC2), tanpa menyimpan informasi apa pun di cloud. Anda dapat menggunakan AD Connector untuk menambahkan satu akun layanan ke Active Directory Anda. AD Connector menghilangkan kebutuhan untuk sinkronisasi direktori atau biaya dan kompleksitas hosting infrastruktur federasi. Meskipun tidak banyak tuas pengoptimalan biaya untuk AD Connector karena sifat layanan dan mekanisme penagihannya, Anda dapat mengikuti rekomendasi desain di bagian ini untuk meminimalkan biaya.

Dampak biaya

AD Connector adalah layanan terkelola berdasarkan preset SKUs. Ini membuat ukuran menjadi proses yang mudah. Ada dua ukuran yang SKUs tersedia: ukuran kecil dan besar. Anda dapat menggunakan AWS Kalkulator Hargauntuk estimasi biaya yang melibatkan AD Connector.

Rekomendasi optimisasi biaya

Selain sumber daya komputasi backend tidak ada perbedaan antara ukuran konektor kecil dan besar.

Pertanyaan yang perlu dipertimbangkan sebelum memilih jenis direktori Anda adalah:

- Apakah ada sejumlah besar (10.000+) pengguna aktif yang menggunakan AWS aplikasi yang terintegrasi dengan AD Connector?
- Apakah pengguna merupakan anggota dari banyak grup bersarang, dalam, atau melingkar?

Jika jawaban untuk kedua pertanyaan tidak, kami sarankan Anda mulai dengan ukuran kecil. Jika Anda menjawab ya untuk salah satu pertanyaan di atas, maka ukuran besar mungkin layak dipertimbangkan. Anda dapat memulai dengan AD Connector ukuran kecil dan, jika direktori menjadi terganggu karena kinerja, Anda dapat meminta direktori ditingkatkan ke ukuran besar.

> **Note**
>
> Anda dapat memutakhirkan Konektor AD dari kecil ke besar, tetapi AD Connector tidak dapat diturunkan peringkatnya.

Sebagian besar masalah kinerja tidak terkait dengan AD Connector, tetapi pengontrol domain Active Directory lokal menjadi kewalahan karena banyak pengguna menjadi anggota dari banyak grup bersarang, dalam, atau melingkar.

Anda juga memiliki opsi untuk mendaftar dalam Perjanjian Perusahaan dengan AWS. Perjanjian Perusahaan memberi Anda pilihan untuk menyesuaikan perjanjian yang paling sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya, lihat Pelanggan Perusahaan.

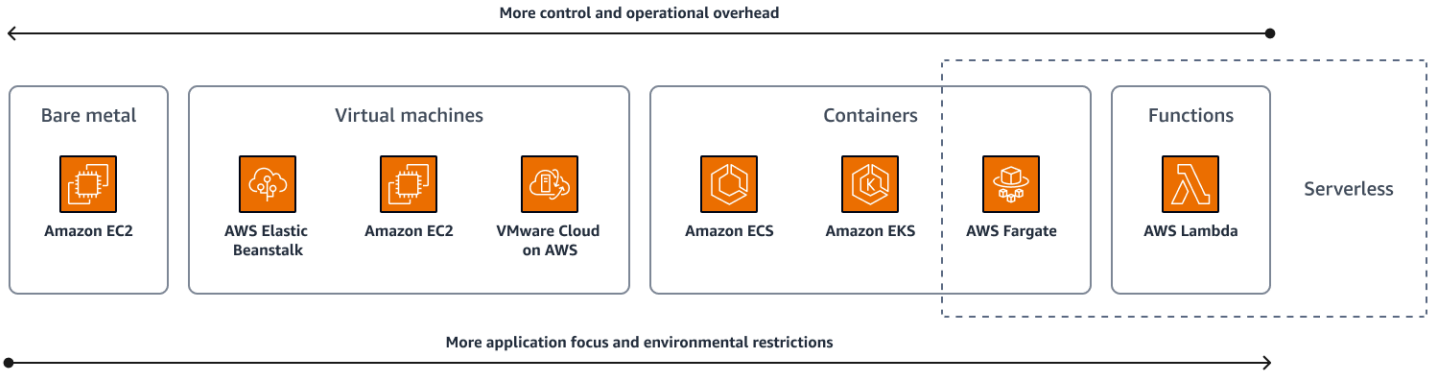
Sumber daya tambahan

- Kuota AD Connector (AWS Directory Service dokumentasi)
- Harga jenis direktori lainnya (AWS situs web)
- Layanan Domain Direktori Aktif di AWS (AWS Whitepaper)

.NET

Mengembangkan dan menerapkan aplikasi.NET adalah kunci penting dalam membantu Anda mencapai skala dan kelincahan yang ditawarkan oleh komputasi awan. Untuk banyak aplikasi .NET lama, pilihan komputasi yang paling cocok untuk menjalankan aplikasi AWS adalah menggunakan mesin virtual, baik melalui atau Amazon Elastic Compute Cloud (AWS Elastic Beanstalk Amazon EC2). Dimungkinkan juga untuk menjalankan aplikasi.NET di wadah Windows dan Linux.

Pengenalan inti .NET memungkinkan Anda untuk merancang aplikasi.NET modern yang memanfaatkan semua manfaat cloud. Aplikasi modern dapat menggunakan serangkaian pilihan komputasi tradisional dan juga menargetkan berbagai jenis lingkungan tanpa server, termasuk atau. AWS Fargate AWS Lambda.NET 6+ sekarang menawarkan hosting beban kerja yang berkinerja baik pada instans ARM64 EC2 seperti keluarga Graviton2 EC2. Ini memungkinkan akses ke prosesor generasi terbaru yang tersedia di Amazon EC2. Ini berarti bahwa aplikasi Anda dapat di-host pada komputasi khusus untuk jenis beban kerja Anda, seperti pengkodean video, server web, dan komputasi kinerja tinggi (HPC).



Bagian ini memberikan rekomendasi untuk membantu Anda menyesuaikan aplikasi.NET Anda untuk memanfaatkan manfaat cloud dengan fokus pada efisiensi biaya.

Bagian ini mencakup topik-topik berikut:

- Refactor ke .NET modern dan pindah ke Linux
- Kontainerisasi aplikasi.NET
- Gunakan instance dan wadah Graviton
- Mendukung penskalaan dinamis untuk aplikasi.NET Framework statis
- Gunakan caching untuk mengurangi permintaan database

- Pertimbangkan .NET tanpa server
- Pertimbangkan database yang dibuat khusus

Refactor ke .NET modern dan pindah ke Linux

Ikhtisar

Memodernisasi aplikasi .NET Framework lama dapat membantu Anda meningkatkan keamanan, kinerja, dan skalabilitas. Cara efektif untuk memodernisasi aplikasi.NET Framework adalah dengan memigrasikannya ke versi.NET modern (6+). Berikut adalah beberapa manfaat utama untuk memindahkan aplikasi ini ke Open-Source .NET:

- Untuk mengurangi biaya lisensi Windows dengan menjalankannya pada sistem operasi Linux
- Manfaatkan ketersediaan bahasa modern
- Dapatkan kinerja yang dioptimalkan untuk berjalan di Linux

Banyak organisasi masih menjalankan versi lama dari .NET Framework. Ini dapat menimbulkan risiko keamanan, karena kerentanan di versi yang lebih lama tidak lagi ditangani oleh Microsoft. Microsoft telah mengakhiri dukungan untuk versi terbaru dari .NET Framework 4.5.2, 4.6, dan 4.6.1. Sangat penting untuk mengevaluasi risiko dan manfaat untuk terus menjalankan versi kerangka kerja yang lebih lama. Untuk mengurangi risiko dan mengurangi biaya, ada baiknya menginvestasikan waktu dan upaya ke refactoring ke versi modern.NET.

Dampak biaya

Pertimbangkan tipe instans EC2 tujuan umum (m5), yang menawarkan keseimbangan sumber daya komputasi, memori, dan jaringan. Contoh ini cocok untuk berbagai aplikasi seperti server web, database berukuran sedang, dan repositori kode sumber.

Misalnya, instance m5.xlarge on-demand dengan memori 4 v CPUs dan 16 GB pada Windows Server (termasuk lisensi) di US East (Virginia N.) berharga $274,48 per bulan. Sumber daya yang sama pada server Linux berharga $140,16 per bulan. Dalam contoh ini, ada pengurangan biaya sebesar 49 persen saat Anda memigrasikan aplikasi dari .NET Framework ke versi modern .NET dan menjalankan aplikasi Anda di server Linux. Biaya Anda dapat bervariasi tergantung pada opsi (misalnya, jenis instans, sistem operasi, penyimpanan) yang Anda pilih saat memilih instans EC2. Anda dapat lebih mengoptimalkan biaya dengan menggunakan Savings Plans atau Instans

Cadangan. Untuk lebih jelasnya, gunakan perkiraan biaya AWS Kalkulator Hargauntuk menjalankan. Untuk instance yang disertakan Windows, biaya lisensi adalah $0,046 per vCPU per jam, terlepas dari model harga.

Porting aplikasi.NET Framework ini ke .NET modern membutuhkan upaya pengembang. Anda harus menilai aplikasi dan dependensinya untuk melihat apakah aplikasi tersebut kompatibel dengan versi platform target. AWS Porting Assistant for .NET adalah alat bantu yang memindai aplikasi.NET Framework dan menghasilkan penilaian kompatibilitas.NET, membantu Anda mem-port aplikasi agar kompatibel dengan Linux lebih cepat. Porting Assistant untuk.NET mengidentifikasi ketidakcocokan dengan.NET, menemukan pengganti yang diketahui, dan menghasilkan penilaian kompatibilitas terperinci. Setelah mem-porting solusi Anda, Anda harus membuat perubahan kode manual agar proyek Anda berhasil dikompilasi dengan dependensi. Ini mengurangi upaya manual yang terlibat dalam memodernisasi aplikasi Anda ke Linux. Jika aplikasi Anda mendukung prosesor ARM, pindah ke Linux membuka kemampuan untuk menggunakan instance Graviton. Ini dapat membantu Anda mencapai tambahan 20 persen dalam pengurangan biaya lebih lanjut. Untuk informasi lebih lanjut, lihat Powering .NET 5 dengan AWS Graviton2: Benchmarks in the Compute Blog. AWS

Ada alat lain, seperti AWS Toolkit untuk .NET Refactoring dan .NET Upgrade Assistant, yang dapat membantu Anda dengan porting aplikasi framework .NET lama ke .NET modern.

Rekomendasi optimisasi biaya

Untuk memigrasikan aplikasi.NET Framework, lakukan hal berikut:

1. Prasyarat — Untuk menggunakan Porting Assistant untuk.NET, Anda harus menginstal .NET 5+ pada mesin tempat Anda berencana untuk menganalisis kode sumber aplikasi. Sumber daya pada mesin harus memiliki kecepatan GHz pemrosesan minimal 1,8, memori 4 GB, dan ruang penyimpanan 5 Gb. Untuk informasi selengkapnya, lihat Prasyarat dalam dokumentasi Porting Assistant for .NET.
2. Penilaian - Unduh Porting Assistant untuk.NET sebagai file yang dapat dieksekusi (unduh). Anda dapat mengunduh dan menginstal alat di mesin Anda untuk memulai penilaian aplikasi Anda. Halaman penilaian berisi proyek porting, paket, dan APIs yang tidak kompatibel dengan .NET modern. Untuk alasan ini, Anda mendapatkan kesalahan build dalam solusi setelah penilaian. Anda dapat melihat atau mengunduh temuan penilaian ke file CSV. Untuk informasi selengkapnya, lihat Mem-port solusi dalam dokumentasi Porting Assistant for .NET.
3. Refactoring — Setelah menilai aplikasi, Anda dapat mem-port proyek Anda ke versi kerangka kerja target. Saat Anda mem-port solusi, file proyek Anda dan beberapa kode akan dimodifikasi oleh Porting Assistant. Anda dapat memeriksa log untuk meninjau perubahan pada kode sumber

Anda. Dalam kebanyakan kasus, kode akan membutuhkan upaya tambahan untuk menyelesaikan migrasi dan pengujian untuk membuatnya siap produksi. Tergantung pada aplikasi, beberapa perubahan mungkin termasuk kerangka kerja entitas, identitas, dan otentikasi. Untuk informasi selengkapnya, lihat Mem-port solusi dalam dokumentasi Porting Assistant for .NET.

Ini adalah langkah pertama untuk memodernisasi aplikasi Anda ke wadah. Mungkin ada sejumlah driver bisnis dan teknis untuk memodernisasi aplikasi.NET Framework Anda ke wadah Linux. Salah satu driver yang signifikan adalah mengurangi total biaya kepemilikan dengan beralih dari sistem operasi Windows ke Linux. Ini mengurangi biaya lisensi saat memigrasikan aplikasi Anda ke versi lintas platform .NET dan ke kontainer untuk mengoptimalkan pemanfaatan sumber daya.

Setelah aplikasi Anda di-porting ke Linux, Anda dapat menggunakan AWS App2Containeruntuk containerize aplikasi Anda. App2Container menggunakan Amazon ECS atau Amazon EKS sebagai layanan endpoint yang dapat Anda gunakan secara langsung. App2Container menyediakan semua infrastruktur yang diperlukan sebagai artefak penyebaran kode (IAc) untuk mengkontainerisasi aplikasi Anda berulang kali.

Pertimbangan dan sumber daya tambahan

- Jika Anda memiliki aplikasi yang dibangun di atas VB.NET (kerangka kerja lama dari tahun 2002) dan ingin mem-portingnya ke.NET 6, lihat aplikasi VB.NET lama Port ke .NET 6.0 dengan Porting Assistant for .NET posting di Microsoft Workloads di blog. AWS
- Jika Anda memiliki aplikasi lama di Windows Communication Foundation (WCF) dan ingin menjalankannya di .NET modern, Anda dapat mengadopsi CoreWCF. Untuk informasi selengkapnya, lihat memodernisasi aplikasi WCF lama ke CoreWCF menggunakan Porting Assistant for .NET posting di Microsoft Workloads di blog. AWS
- Anda dapat menambahkan asisten porting sebagai ekstensi ke Visual Studio IDE Anda. Ini memungkinkan Anda untuk melakukan semua tugas yang diperlukan untuk mengonversi kode Anda tanpa perlu beralih antara IDE Anda dan Porting Assistant for .NET tool. Untuk informasi selengkapnya, lihat Percepat modernisasi aplikasi.NET dengan Porting Assistant for .NET Visual Studio IDE posting ekstensi di Microsoft Workloads di blog. AWS
- AWS Porting Assistant untuk.NET sekarang alat open source dengan kode sumber dan komponen analisis kompatibilitas penilaian. Ini dapat mendorong pengembang Anda untuk menggunakan dan berbagi pengetahuan porting.NET dan praktik terbaik.
- Anda dapat mem-port aplikasi.NET framework ke .NET modern di Linux dengan menggunakan AWS Toolkit untuk.NET Refactoring. Untuk informasi selengkapnya, lihat Akselerasi

modernisasi .NET dengan AWS Toolkit for .NET Refactoring posting di Microsoft Workloads di blog. AWS

- Anda dapat mempercepat kontainerisasi dan migrasi aplikasi ASP.NET Core untuk digunakan. AWS AWS App2Container

Kontainerisasi aplikasi.NET

Ikhtisar

Kontainer adalah cara yang ringan dan efisien untuk mengemas dan menyebarkan aplikasi secara konsisten dan dapat direproduksi. Bagian ini menjelaskan bagaimana Anda dapat menggunakan AWS Fargate, layanan kontainer tanpa server, untuk mengurangi biaya aplikasi.NET Anda sambil juga menyediakan infrastruktur yang dapat diskalakan dan andal.

Dampak biaya

Beberapa faktor yang mempengaruhi efektivitas penggunaan kontainer untuk penghematan biaya termasuk ukuran dan kompleksitas aplikasi, jumlah aplikasi yang perlu digunakan, dan tingkat lalu lintas dan permintaan pada aplikasi. Untuk aplikasi kecil atau sederhana, kontainer mungkin tidak memberikan penghematan biaya yang signifikan dibandingkan dengan pendekatan infrastruktur tradisional karena overhead mengelola kontainer dan layanan terkait sebenarnya dapat meningkatkan biaya. Namun, untuk aplikasi yang lebih besar atau lebih kompleks, menggunakan kontainer dapat memberikan penghematan biaya dengan meningkatkan pemanfaatan sumber daya dan mengurangi jumlah instance yang diperlukan.

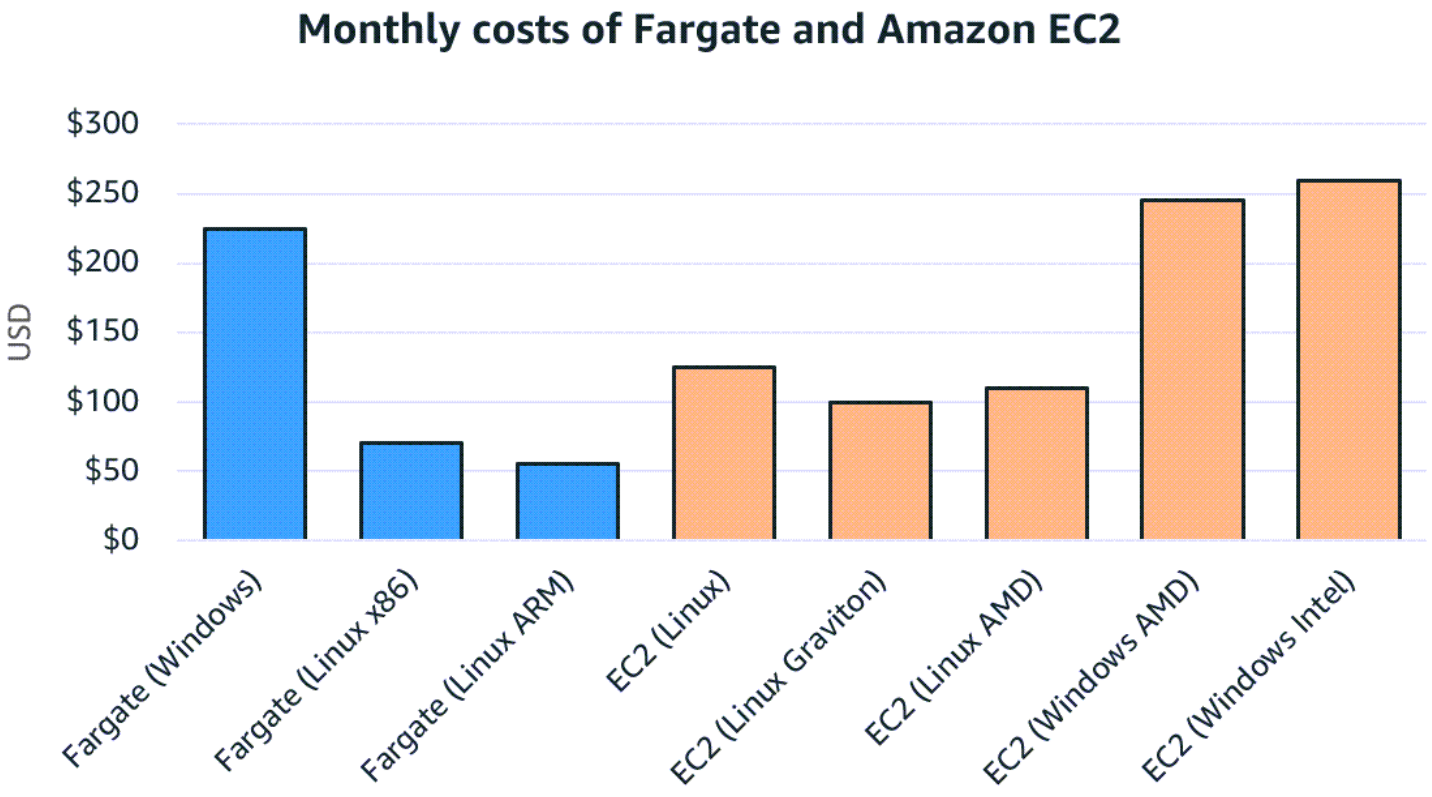
Kami menyarankan Anda untuk mengingat hal-hal berikut saat menggunakan wadah untuk penghematan biaya:

- Ukuran dan kompleksitas aplikasi - Aplikasi yang lebih besar dan lebih kompleks lebih cocok untuk kontainerisasi karena cenderung membutuhkan lebih banyak sumber daya dan dapat memperoleh manfaat lebih banyak dari peningkatan pemanfaatan sumber daya.
- Jumlah aplikasi — Semakin banyak aplikasi yang harus diterapkan organisasi Anda, semakin banyak penghematan biaya yang dapat dicapai melalui kontainerisasi.
- Lalu lintas dan permintaan — Aplikasi yang mengalami lalu lintas dan permintaan tinggi dapat memperoleh manfaat dari skalabilitas dan elastisitas yang disediakan kontainer. Ini dapat menyebabkan penghematan biaya.

Arsitektur dan sistem operasi yang berbeda mempengaruhi biaya kontainer. Jika Anda menggunakan wadah Windows, biaya mungkin tidak berkurang karena pertimbangan lisensi. Biaya lisensi lebih rendah atau tidak ada dengan wadah Linux. Bagan berikut menggunakan konfigurasi dasar AWS Fargate di Wilayah AS Timur (Ohio) dengan pengaturan berikut: 30 tugas per bulan masing-masing berjalan selama 12 jam dengan 4 v CPUs dan 8 GB memori dialokasikan.

Anda dapat memilih dari dua platform komputasi utama untuk menjalankan kontainer Anda di AWS: host kontainer berbasis EC2 dan tanpa server atau. AWS Fargate Jika Anda menggunakan Amazon Elastic Container Service (Amazon ECS) alih-alih Fargate, maka Anda harus tetap menjalankan komputasi (instance) untuk memungkinkan mesin penempatan membuat instance container saat diperlukan. Jika Anda menggunakan Fargate sebagai gantinya, hanya kapasitas komputasi yang diperlukan yang disediakan.

Bagan berikut menunjukkan perbedaan untuk kontainer setara menggunakan Fargate versus Amazon EC2. Karena fleksibilitas Fargate, tugas untuk aplikasi dapat berjalan 12 jam per hari, dengan nol pemanfaatan selama jam libur. Namun, untuk Amazon ECS, Anda harus mengontrol kapasitas komputasi dengan menggunakan grup Auto Scaling dari instans EC2. Hal ini dapat menyebabkan kapasitas berjalan 24 jam sehari, yang pada akhirnya dapat meningkatkan biaya.



Rekomendasi optimisasi biaya

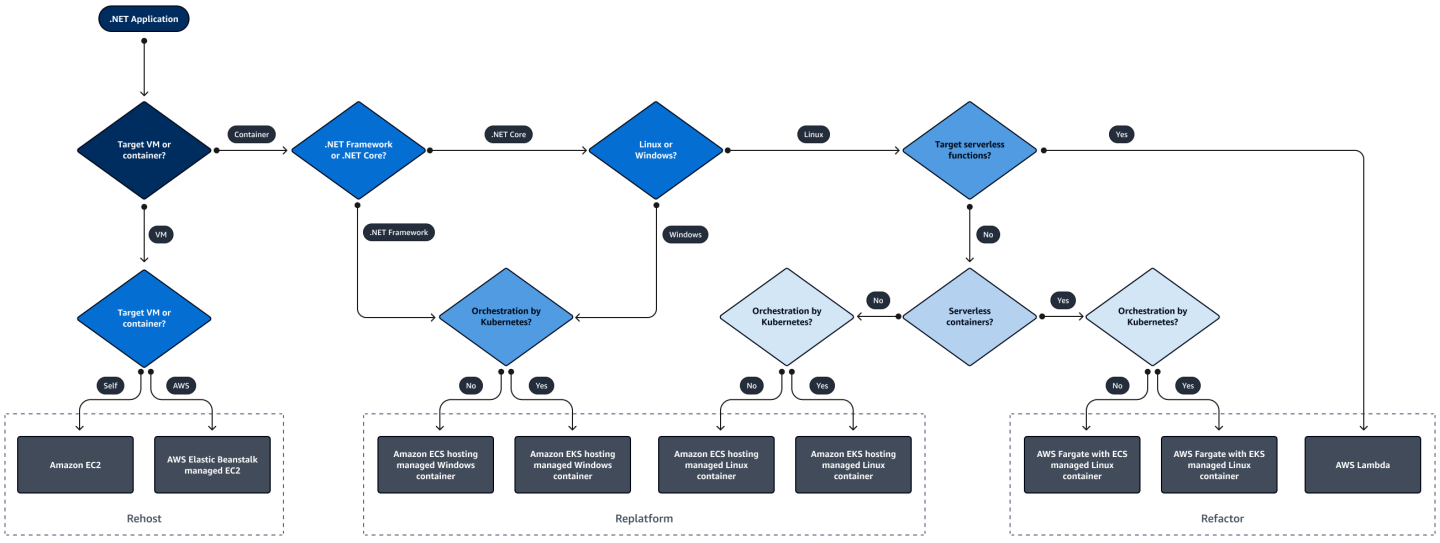
Gunakan wadah Linux daripada Windows

Anda dapat mencapai penghematan yang signifikan jika Anda menggunakan wadah Linux alih-alih wadah Windows. Misalnya, Anda dapat mencapai penghematan sekitar 45 persen pada biaya komputasi jika Anda menjalankan .NET Core di EC2 Linux alih-alih menjalankan .NET Framework pada Windows EC2. Anda bisa mendapatkan penghematan tambahan 40 persen jika Anda menggunakan arsitektur ARM (AWS Graviton) alih-alih x86.

Jika Anda berencana untuk menjalankan kontainer berbasis Linux untuk aplikasi.NET Framework yang ada, Anda harus mem-port aplikasi ini ke versi lintas-platform modern .NET (seperti.NET 6.0) untuk menggunakan wadah Linux. Pertimbangan utama adalah menimbang biaya refactoring dibandingkan dengan penghematan biaya yang diperoleh melalui pengurangan biaya kontainer Linux. Untuk informasi selengkapnya tentang porting aplikasi ke .NET modern, lihat Porting Assistant untuk.NET dalam dokumentasi. AWS

Manfaat lain dari pindah ke .NET modern (yaitu, jauh dari .NET Framework) adalah bahwa peluang modernisasi tambahan tersedia. Misalnya, Anda dapat mempertimbangkan untuk merancang ulang aplikasi Anda ke arsitektur berbasis layanan mikro yang lebih skalabel, gesit, dan hemat biaya.

Diagram berikut menggambarkan proses pengambilan keputusan untuk mengeksplorasi peluang modernisasi.



Manfaatkan Savings Plans

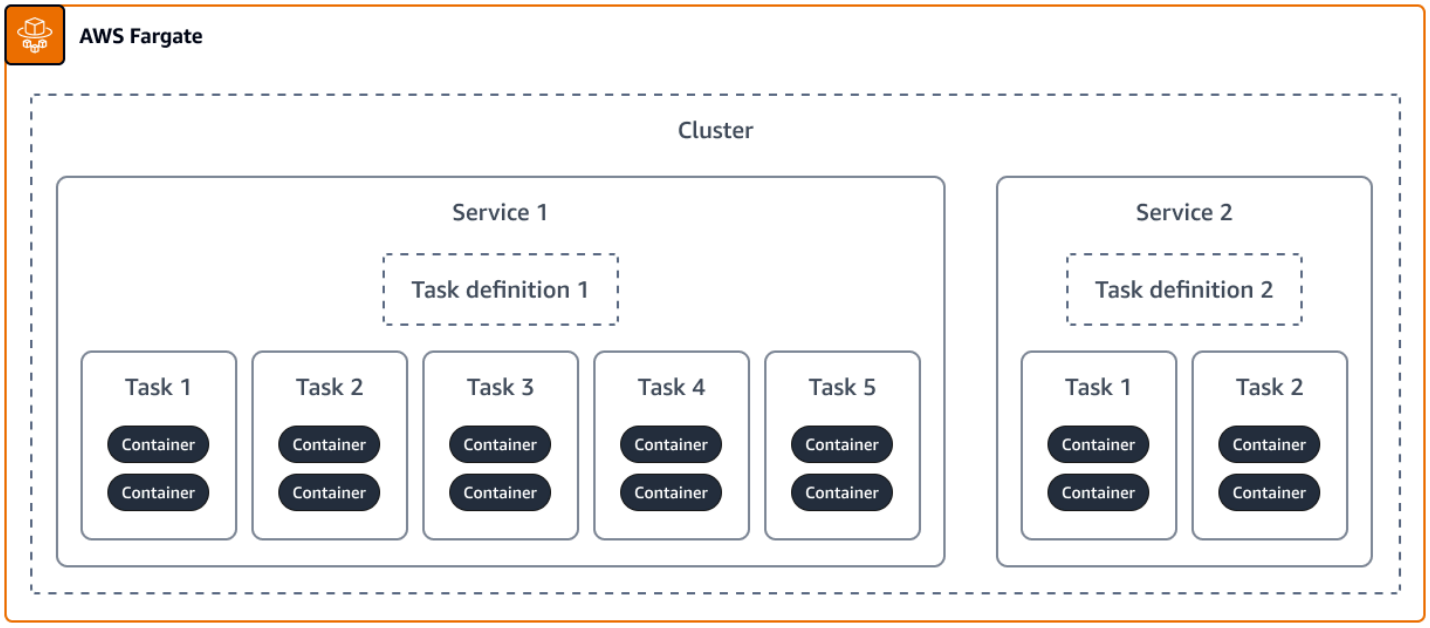
Kontainer dapat membantu Anda memanfaatkan Compute Savings Plans untuk mengurangi biaya Fargate Anda. Model diskon fleksibel menawarkan diskon yang sama dengan Instans Cadangan Konvertibel. Harga Fargate didasarkan pada vCPU dan sumber daya memori yang digunakan sejak Anda mulai mengunduh gambar kontainer hingga tugas Amazon ECS berakhir (dibulatkan ke detik terdekat). Savings Plans untuk Fargate menawarkan penghematan hingga 50 persen pada penggunaan Fargate dengan imbalan komitmen untuk menggunakan jumlah tertentu dari penggunaan komputasi (diukur dalam dolar per jam) untuk jangka waktu satu tahun atau tiga tahun. Anda dapat menggunakan AWS Cost Exploreruntuk membantu Anda memilih Savings Plans.

Penting untuk dipahami bahwa Compute Savings Plans diterapkan pada penggunaan yang memberi Anda penghematan terbesar terlebih dahulu. Misalnya, jika Anda menjalankan instans Linux t3.medium `us-east-2` dan instance Windows t3.medium yang identik, instance Linux menerima manfaat Savings Plans terlebih dahulu. Ini karena instance Linux memiliki potensi penghematan 50 persen sedangkan instance Windows yang sama memiliki potensi penghematan 35 persen. Jika Anda memiliki sumber daya lain yang memenuhi syarat Savings Plans berjalan di Anda Akun AWS, seperti Amazon EC2 atau Lambda, maka Savings Plans Anda tidak perlu diterapkan terlebih dahulu ke Fargate. Untuk informasi selengkapnya, lihat Memahami bagaimana Savings Plans berlaku untuk AWS penggunaan Anda di dokumentasi Savings Plans dan bagian Optimalkan pengeluaran untuk Windows di Amazon EC2 pada panduan ini.

Tugas Fargate ukuran yang tepat

Penting untuk memastikan bahwa tugas Fargate berukuran benar untuk mencapai tingkat optimasi biaya maksimum. Seringkali, pengembang tidak memiliki semua informasi penggunaan yang diperlukan ketika awalnya menentukan konfigurasi untuk tugas Fargate yang digunakan dalam aplikasi mereka. Hal ini dapat menyebabkan penyediaan tugas yang berlebihan dan kemudian menghasilkan pengeluaran yang tidak perlu. Untuk menghindari hal ini, kami menyarankan Anda memuat aplikasi pengujian yang berjalan di Fargate untuk memahami kinerja konfigurasi tugas tertentu di bawah skenario penggunaan yang berbeda. Anda dapat menggunakan hasil pengujian beban, vCPU, alokasi memori tugas, dan kebijakan penskalaan otomatis untuk menemukan keseimbangan yang tepat antara kinerja dan biaya.

Diagram berikut menunjukkan bagaimana Compute Optimizer menghasilkan rekomendasi untuk tugas dan ukuran wadah yang optimal.



Salah satu pendekatannya adalah dengan menggunakan alat pengujian beban, seperti yang dijelaskan dalam Pengujian Beban Terdistribusi pada AWS, untuk menetapkan dasar untuk vCPU dan pemanfaatan memori. Setelah Anda menjalankan uji beban untuk mensimulasikan beban aplikasi biasa, maka Anda dapat menyempurnakan konfigurasi vCPU dan memori untuk tugas tersebut hingga pemanfaatan dasar tercapai.

Sumber daya tambahan

- Daftar Periksa Pengoptimalan Biaya untuk Amazon ECS dan AWS Fargate (Posting blog AWS Container)
- Optimalisasi biaya teoritis oleh jenis peluncuran Amazon ECS: Fargate vs EC2 AWS (Posting blog kontainer)
- Porting Assistant untuk .NET (AWS dokumentasi)
- Pengujian Beban Terdistribusi pada AWS (Perpustakaan AWS Solusi)
- AWS Compute Optimizer meluncurkan dukungan untuk layanan Amazon ECS di AWS Fargate (posting blog AWS Cloud Financial Management)

Gunakan instance dan wadah Graviton

Ikhtisar

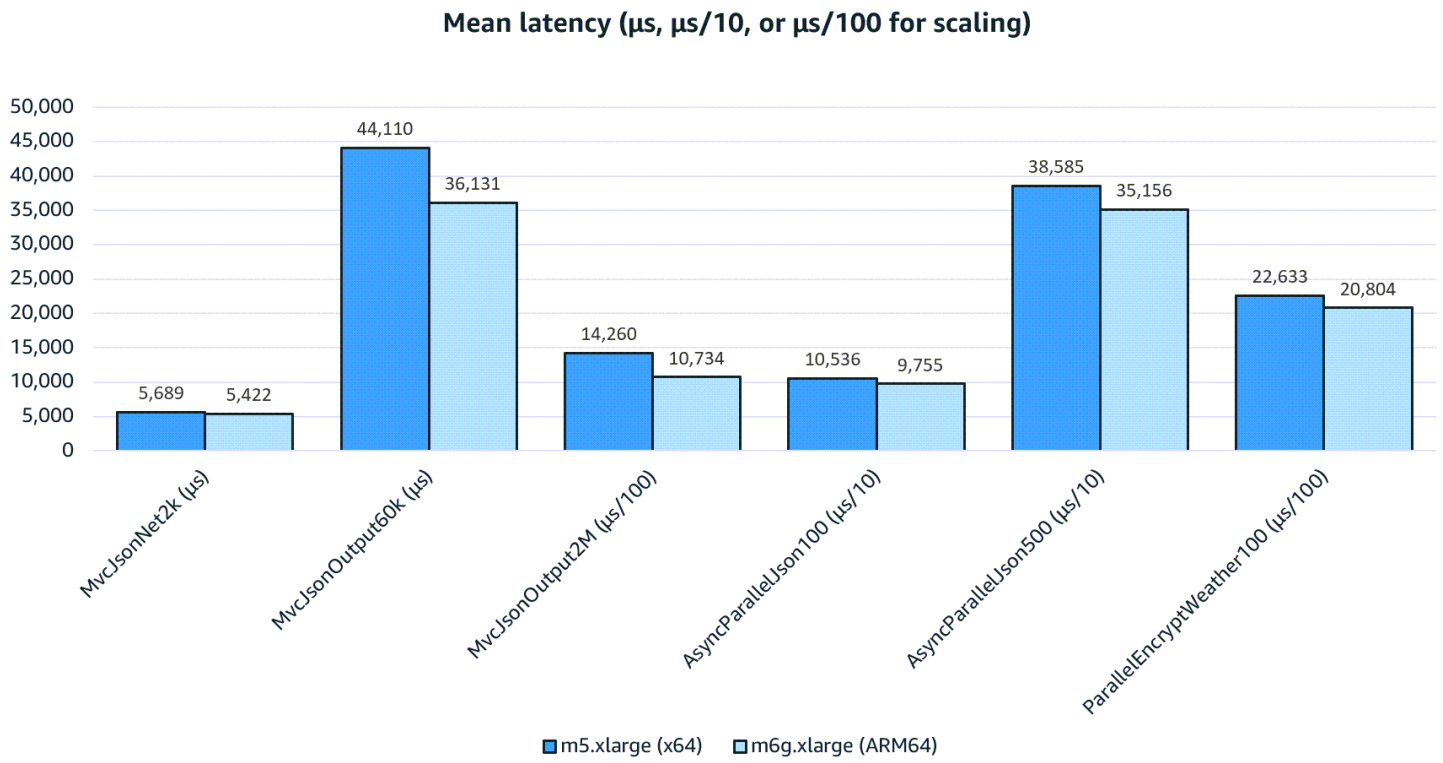
AWS Instans Graviton didukung oleh prosesor ARM yang dirancang AWS untuk memberikan kinerja harga terbaik untuk beban kerja cloud Anda yang berjalan di Amazon Elastic Compute Cloud (Amazon EC2), termasuk container yang berjalan di dalamnya. AWS Saat ini ada tiga generasi Graviton yang tersedia untuk digunakan di Amazon EC2. Panduan ini berfokus pada penggunaan Graviton 2 dan 3 dengan aplikasi.NET karena ada penghematan biaya yang signifikan ketika Anda menggunakan Graviton versi terbaru. Perlu diingat bahwa instance Graviton hanya menjalankan sistem operasi Linux. Akibatnya, instance Graviton adalah penawaran yang kuat untuk .NET yang berjalan di Linux, tetapi bukan pilihan untuk sistem operasi Windows atau aplikasi .NET Framework lama.

Graviton 3 60 persen lebih efisien dibandingkan instans EC2 yang sebanding dengan kinerja hingga 40 persen lebih baik. Panduan ini berfokus pada manfaat biaya penggunaan Graviton, tetapi penting untuk dicatat bahwa Graviton menawarkan manfaat tambahan dari peningkatan kinerja dan peningkatan kelestarian lingkungan.

Dampak biaya

Anda dapat mencapai penghematan hingga 45 persen saat beralih ke Graviton. Setelah Anda memfaktorkan ulang aplikasi.NET Framework lama ke versi.NET modern, Anda membuka kunci kemampuan menggunakan instance Graviton. Pindah ke Graviton adalah teknik optimasi biaya yang efektif untuk pengembang.NET.

Contoh dalam tabel berikut menunjukkan potensi peningkatan kinerja yang dapat Anda capai dengan bermigrasi ke instance Graviton.



Untuk rincian lengkap dan penjelasan tentang pendekatan benchmarking yang digunakan untuk membuat hasil dalam diagram sebelumnya, lihat Powering .NET 5 dengan AWS Graviton2: Benchmarks in the Compute Blog. AWS

Salah satu alasan peningkatan efisiensi adalah perbedaan arti vCPU antara x86 dan Graviton. Dalam arsitektur x86, vCPU adalah inti logis yang dicapai oleh hyperthreading. Di Graviton, vCPU setara dengan inti fisik yang memungkinkan vCPU berkomitmen penuh terhadap beban kerja.

Hasilnya dengan Graviton2 adalah kinerja harga 40 persen lebih baik dibandingkan instans x86/x64 yang sebanding. Graviton3 menawarkan yang berikut ini di atas Graviton2:

- Profil kinerja yang meningkat dengan kinerja hingga 25 persen lebih baik
- Performa floating-point hingga dua kali lebih tinggi
- Kinerja beban kerja kriptografi hingga dua kali lebih cepat
- Performa machine learning hingga tiga kali lebih baik

Selain itu, Graviton3 adalah contoh pertama di cloud yang menampilkan memori. DDR5

Tabel berikut menunjukkan perbedaan penghematan biaya antara instance berbasis Graviton dan instance berbasis x86 yang setara.

Tabel ini menunjukkan penghematan Graviton sebesar 19,20 persen.

Tipe instans	Arsitektur	vCPU	Memori (GB)	Biaya per jam (sesuai permintaan)
t4g.xlarge	LENGAN	4	16	$0,1344
t3.xlarge	x86	4	16	$0,1664

Tabel ini menunjukkan penghematan Graviton sebesar 14,99 persen.

Tipe instans	Arsitektur	vCPU	Memori (GB)	Biaya per jam (sesuai permintaan)
c7g.4xlarge	LENGAN	16	32	$0,5781
c6i.4xlarge	x86	16	32	$0.6800

Penting untuk menguji profil kinerja aplikasi Anda saat mempertimbangkan Graviton. Graviton bukanlah pengganti praktik pengembangan perangkat lunak yang solid. Anda dapat menggunakan pengujian untuk memverifikasi apakah Anda mendapatkan hasil maksimal dari sumber daya komputasi yang mendasarinya.

Rekomendasi optimisasi biaya

Ada beberapa cara untuk memanfaatkan prosesor/instance Graviton. Bagian ini memandu Anda melalui perubahan yang diperlukan untuk beralih dari menggunakan mesin arsitektur x86 ke instance Graviton (ARM).

Ubah pengaturan runtime di Lambda

Kami menyarankan Anda untuk mengganti pengaturan runtime. AWS Lambda Untuk informasi selengkapnya, lihat Memodifikasi lingkungan runtime dalam dokumentasi Lambda. Karena .NET adalah bahasa yang dikompilasi, Anda harus mengikuti proses pembuatan agar ini berfungsi. Untuk contoh bagaimana melakukan ini, lihat. NET di Graviton di. GitHub

Wadah

Untuk beban kerja kontainer, buat gambar kontainer multi-arsitektur. Anda dapat melakukan ini dengan menentukan beberapa arsitektur dalam perintah build Docker. Contoh:

```
docker buildx build -t "myImageName:latest" --platform linux/amd64,linux/arm64 --push .
```

Anda juga dapat menggunakan alat seperti AWS Cloud Development Kit (AWS CDK) untuk membantu mengatur build. Untuk contoh dari Docker, lihat Membangun Gambar Multi-Arch untuk Arm dan x86 dengan Desktop Docker dalam dokumentasi Docker.

Amazon EC2

Untuk bermigrasi ke ARM dari x86/x64, targetkan arsitektur ARM di langkah kompilasi. Di Visual Studio, Anda dapat membuat ARM64 CPU. Untuk petunjuknya, lihat Untuk mengonfigurasi proyek untuk menargetkan Arm64 dan platform lain dalam dokumentasi Microsoft.

Jika Anda menggunakan .NET CLI, menjalankan build pada mesin ARM menghasilkan build yang kompatibel dengan Graviton. Untuk melihat demo, tonton kinerja Accelerate .NET 6 dengan Arm64 di AWS Graviton2 aktif. YouTube Masalah ketergantungan akan mengakibatkan kesalahan waktu kompilasi yang kemudian dapat diatasi secara individual. Selama ada perpustakaan ARM untuk ketergantungan apa pun, transisi harus relatif mudah.

Sumber daya tambahan

- Cara membuat wadah Anda untuk ARM dan menyimpan dengan instans Graviton dan Spot di Amazon ECS (blog)AWS
- AWS Lambda Fungsi Didukung oleh Prosesor AWS Graviton2 - Jalankan Fungsi Anda di Lengan dan Dapatkan Kinerja Harga Hingga 34% Lebih Baik (blog)AWS
- Migrasi AWS Lambda fungsi ke prosesor AWS Graviton2 berbasis ARM (blog)AWS
- Bangun dan terapkan aplikasi web.NET ke AWS Graviton 2 Amazon ECS Cluster yang didukung ARM menggunakan (blog) AWS CDKAWS
- Graviton Fast Start - Program Baru untuk Membantu Memindahkan Beban Kerja Anda ke AWS Graviton (blog)AWS
- Mendukung .NET 5 dengan AWS Graviton2: Tolok Ukur (blog)AWS

Mendukung penskalaan dinamis untuk aplikasi.NET Framework statis

Ikhtisar

Salah satu manfaat utama menggunakan cloud untuk aplikasi adalah elastisitas, atau kemampuan untuk menskalakan komputasi masuk atau keluar berdasarkan permintaan. Ini memungkinkan Anda untuk hanya membayar kapasitas komputasi yang Anda butuhkan, daripada menyediakan untuk penggunaan puncak. Cyber Monday, di mana pengecer online dapat dengan cepat mendapatkan lalu lintas berkali-kali lebih banyak dari biasanya (misalnya, ribuan persen dalam beberapa menit), adalah contoh elastisitas yang baik.

Jika Anda membawa aplikasi web.NET lama ke cloud (misalnya, aplikasi ASP.NET Framework yang berjalan pada IIS), kemampuan untuk dengan cepat menskalakan beban peternakan server yang seimbang mungkin sulit atau tidak mungkin karena sifat aplikasi yang stateful. Data sesi pengguna disimpan dalam memori aplikasi, biasanya dengan status sesi ASP.NET atau variabel statis yang menyimpan data permintaan silang yang harus dipertahankan. Afinitas sesi pengguna biasanya dipertahankan melalui sesi lengket penyeimbang beban.

Ini terbukti menantang secara operasional. Ketika peningkatan kapasitas diperlukan, Anda harus dengan sengaja menyediakan dan menambahkan server. Ini bisa menjadi proses yang lambat. Mengambil node keluar dari layanan jika terjadi penambalan atau kegagalan yang tidak terduga dapat menjadi masalah bagi pengalaman pengguna akhir, kehilangan status untuk semua pengguna yang terkait dengan node yang terpengaruh. Paling-paling, ini akan mengharuskan pengguna untuk masuk lagi.

Dengan memusatkan status sesi untuk aplikasi ASP.NET dan menerapkan aturan penskalaan otomatis ke aplikasi ASP.NET lama, Anda dapat memanfaatkan elastisitas cloud dan berpotensi memanfaatkan penghematan biaya saat menjalankan aplikasi. Misalnya, Anda mendapatkan pengurangan biaya melalui skalabilitas komputasi, tetapi Anda juga dapat memilih dari berbagai model harga yang tersedia, seperti mengurangi penggunaan instans cadangan dan menggunakan harga Instans Spot Amazon.

Dua teknik umum termasuk menggunakan Amazon DynamoDB sebagai penyedia status sesi dan menggunakan ElastiCache Amazon (Redis OSS) sebagai penyimpanan sesi ASP.NET.

Diagram berikut menunjukkan arsitektur yang menggunakan DynamoDB sebagai penyedia status sesi.

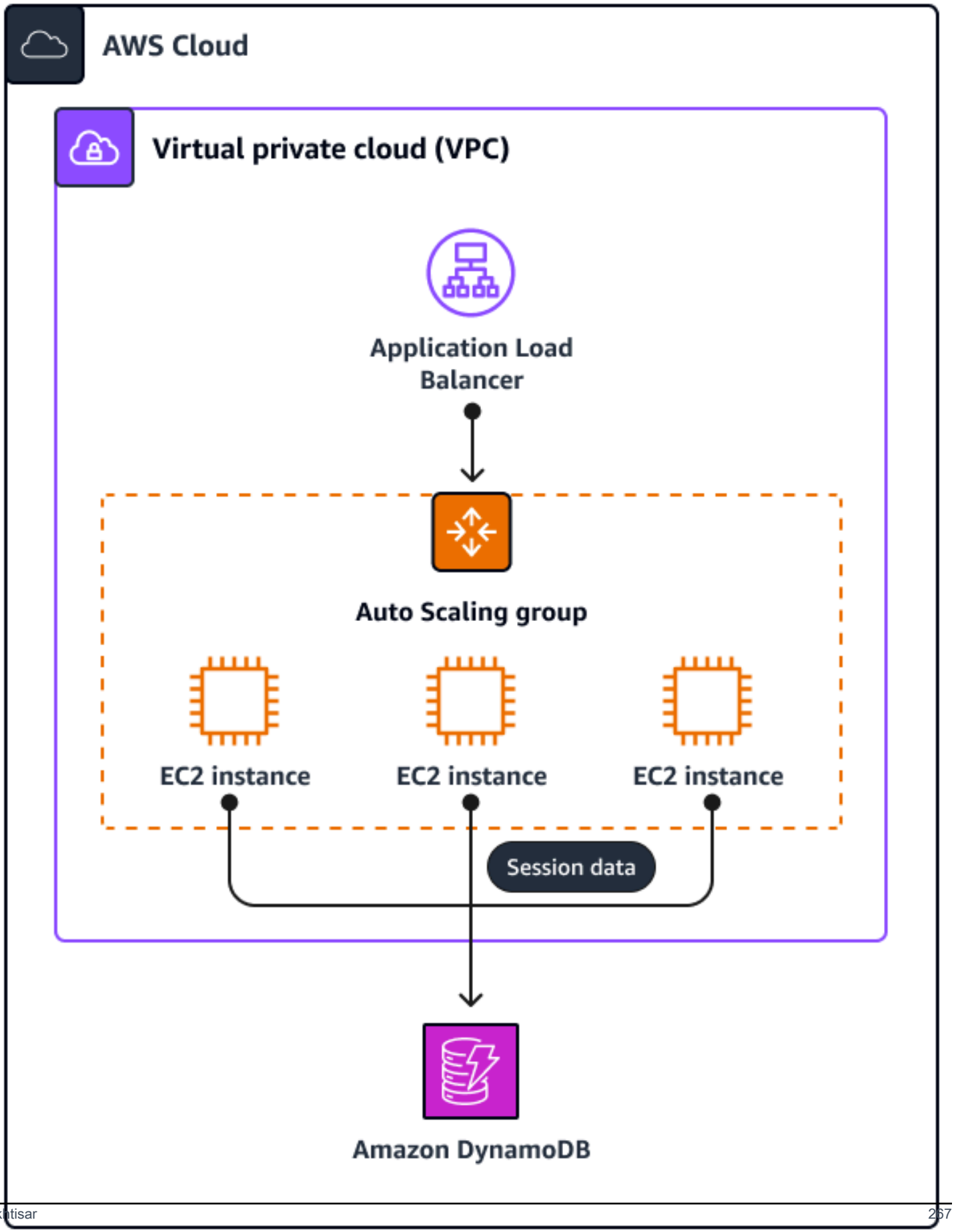
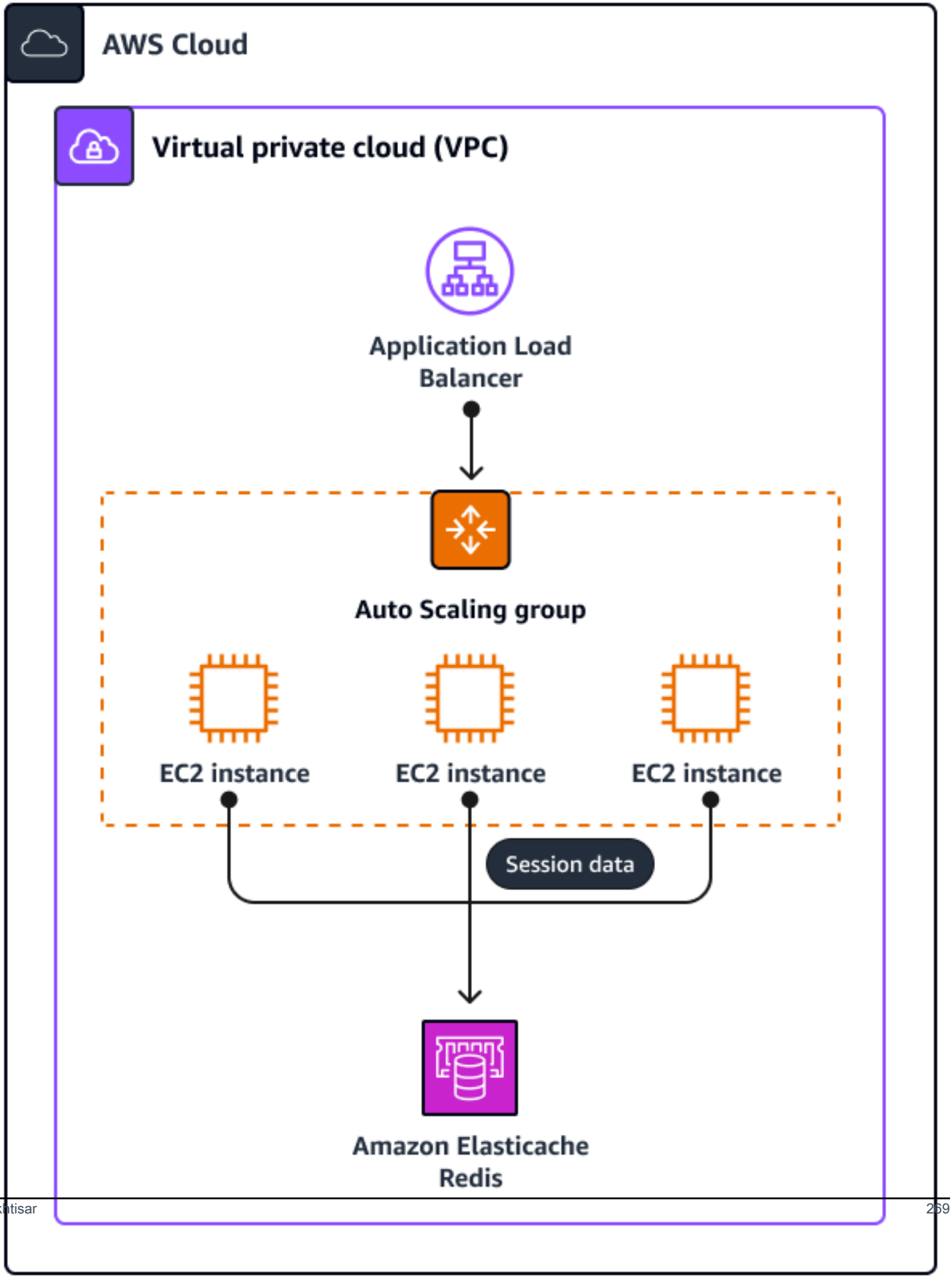

AWS Cloud
Virtual private cloud (VPC)
Application Load Balancer
Auto Scaling group
EC2 instance
EC2 instance
EC2 instance
Session data
Amazon DynamoDB

Diagram berikut menunjukkan arsitektur yang menggunakan ElastiCache (Redis OSS) sebagai penyedia status sesi.

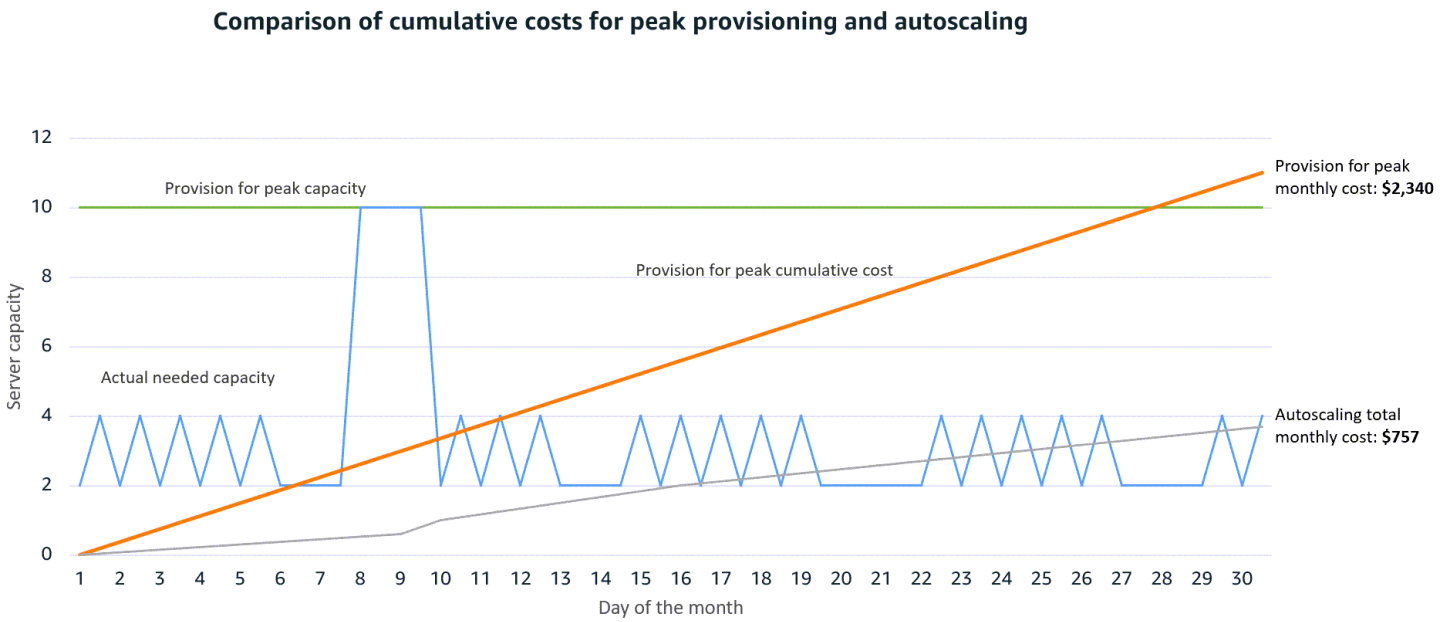

AWS Cloud
Virtual private cloud (VPC)
Application Load Balancer
Auto Scaling group
EC2 instance
EC2 instance
EC2 instance
Session data
Amazon Elasticache Redis

Dampak biaya

Untuk menentukan manfaat penskalaan untuk aplikasi produksi, kami menyarankan Anda memodelkan permintaan Anda yang sebenarnya. Bagian ini membuat asumsi berikut untuk memodelkan aplikasi sampel:

- Contoh yang ditambahkan dan dihapus dari rotasi identik dan tidak ada variasi ukuran instance yang diperkenalkan.
- Pemanfaatan server tidak pernah turun di bawah dua server aktif untuk menjaga ketersediaan aplikasi yang tinggi.
- Jumlah server berskala linier dengan lalu lintas (yaitu, lalu lintas dua kali lebih banyak akan membutuhkan komputasi dua kali lebih banyak).
- Lalu lintas dimodelkan selama sebulan dalam peningkatan enam jam, dengan variasi intra-hari dan satu puncak lalu lintas abnormal (misalnya, penjualan promosi) untuk satu hari lalu lintas 10x. Lalu lintas akhir pekan dimodelkan pada pemanfaatan dasar.
- Lalu lintas malam hari dimodelkan pada pemanfaatan dasar, sementara lalu lintas hari kerja dimodelkan pada pemanfaatan 4x.
- Harga Instans Cadangan menggunakan harga satu tahun tanpa dimuka. Harga siang hari normal menggunakan harga sesuai permintaan sementara permintaan burst menggunakan harga Instans Spot.

Diagram berikut menggambarkan bagaimana model ini memanfaatkan elastisitas dalam aplikasi.NET daripada penyediaan untuk penggunaan puncak. Ini menghasilkan penghematan sekitar 68 persen.



Jika Anda menggunakan DynamoDB sebagai mekanisme penyimpanan status sesi, maka gunakan parameter berikut:

```
Storage: 20GB
Session Reads: 40 million
Session Writes: 20 million
Pricing Model: On demand
```

Perkiraan biaya bulanan untuk layanan ini adalah sekitar $35,00 per bulan.

Jika Anda menggunakan ElastiCache (Redis OSS) sebagai mekanisme penyimpanan status sesi, maka gunakan parameter berikut:

```
Number of Nodes: 3
Node size: cache.t4g.medium
Pricing Model: 1y reserved
```

Perkiraan biaya bulanan untuk layanan ini adalah sekitar $91,00 per bulan.

Rekomendasi optimisasi biaya

Langkah pertama adalah mengimplementasikan status sesi dalam aplikasi.NET lama. Jika Anda menggunakan ElastiCache sebagai mekanisme penyimpanan status Anda, ikuti panduan dari ElastiCache sebagai ASP.NET Session Store di Blog Alat AWS Pengembang. Jika Anda

menggunakan DynamoDB, ikuti panduan dari Apa yang ada AWS SDK untuk .NET di dokumentasi. SDK untuk .NET

Jika aplikasi menggunakan InProcsesi untuk memulai, pastikan bahwa semua objek yang Anda rencanakan untuk disimpan dalam sesi dapat diserialkan. Untuk melakukan ini, gunakan `SerializableAttribute` atribut untuk menghias kelas yang instance akan disimpan dalam sesi. Contoh:

```
[Serializable()]
public class TestSimpleObject {
public string SessionProperty {get;set;}
}
```

Selain itu, .NET `MachineKey` harus sama antara semua server yang digunakan. Ini biasanya terjadi ketika instance dibuat dari Amazon Machine Image (AMI) yang umum. Contoh:

```
<machineKey
validationKey="some long hashed value"
decryptionKey="another long hashed value"
validation="SHA1"/>
```

Namun, penting untuk memastikan bahwa jika gambar dasar diubah, itu dikonfigurasi dengan image mesin.NET yang sama (dapat dikonfigurasi pada tingkat IIS atau server). Untuk informasi lebih lanjut, lihat SystemWebSectionGroup. MachineKey Properti dalam dokumentasi Microsoft.

Terakhir, Anda harus menentukan mekanisme untuk menambahkan server ke grup Auto Scaling sebagai respons terhadap peristiwa penskalaan. Ada beberapa cara untuk mencapai ini. Kami merekomendasikan metode berikut untuk menerapkan aplikasi.NET Framework dengan mulus ke instans EC2 dalam grup Auto Scaling:

- Gunakan EC2 Image Builder untuk mengonfigurasi AMI yang berisi server dan aplikasi yang sepenuhnya dikonfigurasi. Anda kemudian dapat menggunakan AMI ini untuk mengonfigurasi template peluncuran grup Auto Scaling Anda.
- Gunakan AWS CodeDeployuntuk menyebarkan aplikasi Anda. CodeDeploy memungkinkan integrasi langsung dengan Amazon EC2 Auto Scaling. Ini memberikan alternatif untuk membuat AMI baru untuk setiap rilis aplikasi.

Sumber daya tambahan

- Membuat gambar dengan EC2 Image Builder (dokumentasi EC2 Image Builder)
- Menyebarkan Aplikasi Web .NET Menggunakan AWS CodeDeploy Layanan Tim Visual Studio (Blog Alat AWS Pengembang)

Gunakan caching untuk mengurangi permintaan database

Ikhtisar

Anda dapat menggunakan caching sebagai strategi yang efektif untuk membantu mengurangi biaya untuk aplikasi.NET Anda. Banyak aplikasi menggunakan basis data backend, seperti SQL Server, ketika aplikasi memerlukan akses yang sering ke data. Biaya pemeliharaan layanan backend ini untuk mengatasi permintaan bisa tinggi, tetapi Anda dapat menggunakan strategi caching yang efektif untuk mengurangi beban pada basis data backend dengan mengurangi persyaratan ukuran dan penskalaan. Ini dapat membantu Anda mengurangi biaya dan meningkatkan kinerja aplikasi Anda.

Caching adalah teknik yang berguna untuk menghemat biaya yang terkait dengan membaca beban kerja berat yang menggunakan sumber daya yang lebih mahal seperti SQL Server. Sangat penting untuk menggunakan teknik yang tepat untuk beban kerja Anda. Misalnya, caching lokal tidak dapat diskalakan dan mengharuskan Anda untuk mempertahankan cache lokal untuk setiap instance aplikasi. Anda harus mempertimbangkan dampak kinerja dibandingkan dengan biaya potensial, sehingga biaya yang lebih rendah dari sumber data yang mendasarinya mengimbangi biaya tambahan yang terkait dengan mekanisme caching.

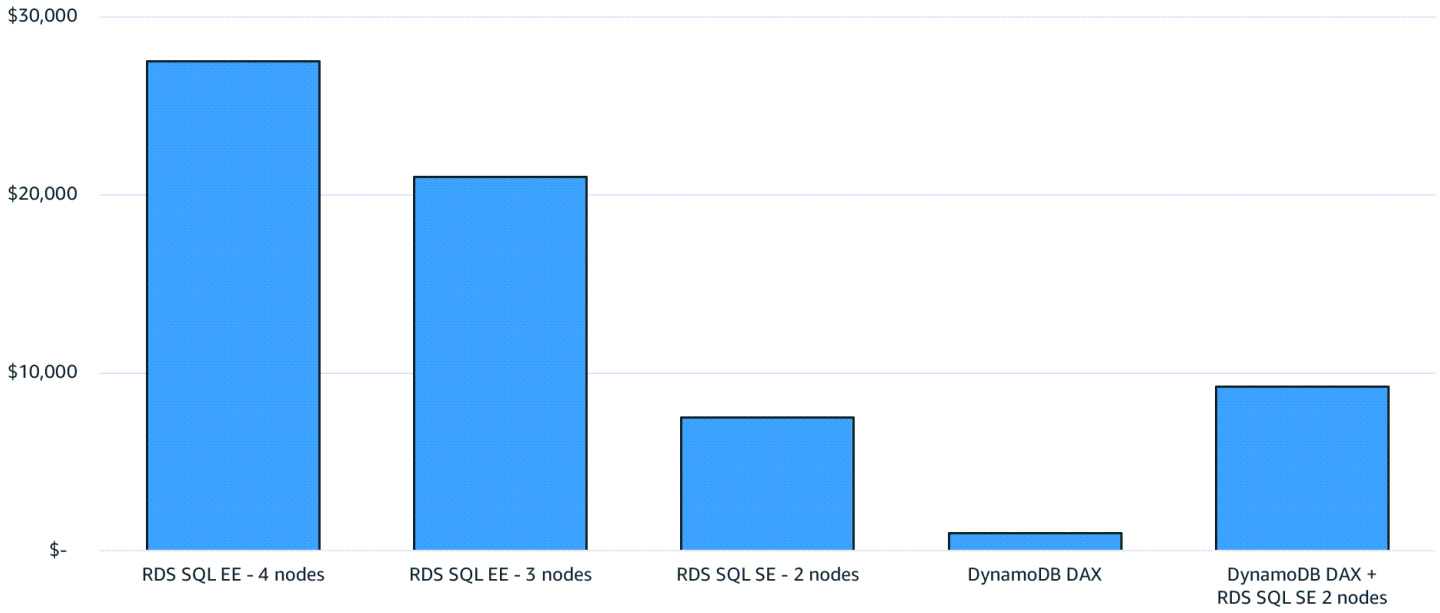
Dampak biaya

SQL Server mengharuskan Anda untuk memperhitungkan permintaan baca saat mengukur database Anda. Ini dapat memengaruhi biaya, karena Anda mungkin perlu memperkenalkan replika baca untuk mengakomodasi beban. Jika Anda menggunakan replika baca, penting untuk dipahami bahwa replika tersebut hanya tersedia di edisi SQL Server Enterprise. Edisi ini membutuhkan lisensi yang lebih mahal daripada edisi Standar SQL Server.

Diagram berikut dirancang untuk membantu Anda memahami efektivitas caching. Ini menunjukkan Amazon RDS for SQL Server dengan empat node db.m4.2xlarge yang menjalankan edisi SQL Server Enterprise. Ini digunakan dalam konfigurasi multi-AZ dengan satu replika baca. Lalu lintas baca

eksklusif (misalnya, kueri SELECT) diarahkan ke replika baca. Sebagai perbandingan, Amazon DynamoDB menggunakan klaster DynamoDB Accelerator (DAX) dua node r4.2xlarge.

Bagan berikut menunjukkan hasil penghapusan kebutuhan untuk replika baca khusus yang menangani lalu lintas baca tinggi.



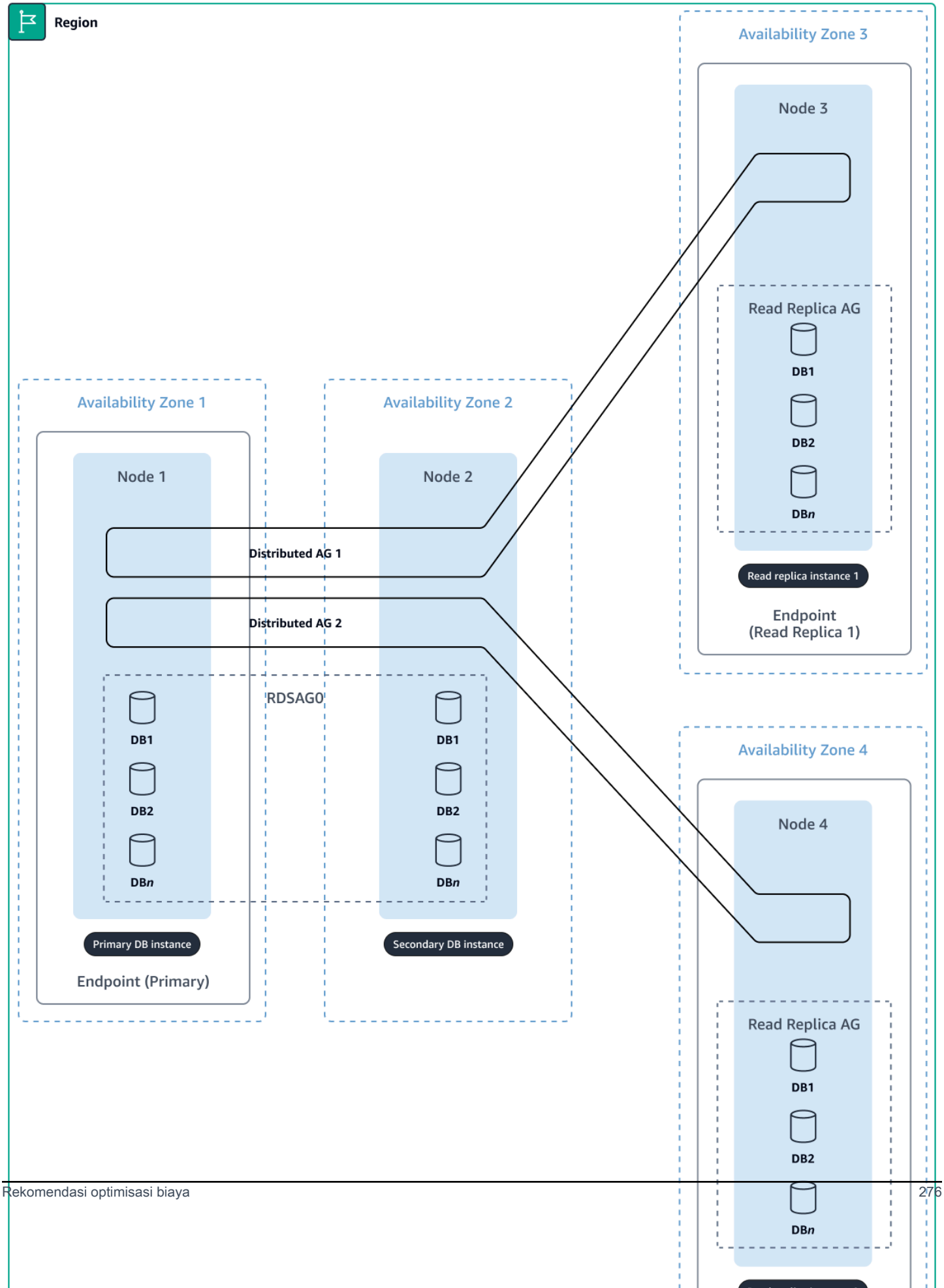
Anda dapat mencapai penghematan biaya yang signifikan dengan menggunakan caching lokal tanpa replika baca atau dengan memperkenalkan DAX berdampingan dengan SQL Server di Amazon RDS sebagai lapisan caching. Lapisan ini diturunkan dari SQL Server dan mengurangi ukuran SQL Server yang diperlukan untuk menjalankan database.

Rekomendasi optimisasi biaya

Caching lokal

Caching lokal adalah salah satu cara yang paling umum digunakan untuk menyimpan konten cache untuk aplikasi yang dihosting baik di lingkungan lokal atau di cloud. Ini karena relatif mudah dan intuitif untuk diterapkan. Caching lokal melibatkan pengambilan konten dari database atau sumber lain dan baik caching lokal di memori atau pada disk untuk akses yang lebih cepat. Pendekatan ini, meskipun mudah diterapkan, tidak ideal untuk beberapa kasus penggunaan. Misalnya, ini termasuk kasus penggunaan ketika konten caching perlu bertahan dari waktu ke waktu, seperti mempertahankan status aplikasi atau status pengguna. Kasus penggunaan lainnya adalah ketika konten yang di-cache diperlukan untuk diakses dari instance aplikasi lain.

Diagram di bawah ini menggambarkan cluster SQL Server yang sangat tersedia dengan empat node dan dua replika baca.


Region
Availability Zone 3
Node 3
Read Replica AG
DB1
DB2
DBn
Read replica instance 1
Endpoint
(Read Replica 1)
Availability Zone 1
Availability Zone 2
Node 1
Node 2
Distributed AG 1
Distributed AG 2
RDSAG0
DB1
DB2
DBn
DB1
DB2
DBn
Primary DB instance
Secondary DB instance
Endpoint (Primary)
Availability Zone 4
Node 4
Read Replica AG
DB1
DB2
DBn

Dengan caching lokal, Anda mungkin perlu memuat lalu lintas keseimbangan di beberapa instans EC2. Setiap instance harus mempertahankan cache lokalnya sendiri. Jika cache menyimpan informasi stateful, perlu ada komit reguler ke database, dan pengguna mungkin perlu diteruskan ke instance yang sama untuk setiap permintaan berikutnya (sesi lengket). Ini menghadirkan tantangan ketika mencoba menskalakan aplikasi karena beberapa contoh dapat digunakan secara berlebihan, sementara beberapa kurang dimanfaatkan karena distribusi lalu lintas yang tidak merata.

Anda dapat menggunakan caching lokal, baik dalam memori atau menggunakan penyimpanan lokal, untuk aplikasi.NET. Untuk melakukannya, Anda dapat menambahkan fungsionalitas untuk menyimpan objek pada disk dan mengambilnya saat diperlukan, atau kueri data dari database dan menyimpannya di memori. Untuk melakukan caching lokal dalam memori dan penyimpanan data lokal dari SQL Server di C #, misalnya, Anda dapat menggunakan kombinasi dan pustaka. `MemoryCache` `LiteDB` `MemoryCache`menyediakan caching dalam memori, sementara `LiteDB` merupakan database berbasis disk NoSQL tertanam yang cepat dan ringan.

Untuk melakukan caching dalam memori, gunakan Library .NET. `System.Runtime.MemoryCache` Contoh kode berikut menunjukkan bagaimana menggunakan `System.Runtime.Caching.MemoryCache` kelas untuk cache data dalam memori. Kelas ini menyediakan cara untuk sementara menyimpan data dalam memori aplikasi. Ini dapat membantu meningkatkan kinerja aplikasi dengan mengurangi kebutuhan untuk mengambil data dari sumber daya yang lebih mahal, seperti database atau API.

Begini cara kerja kode:

1. Sebuah instance statis pribadi `MemoryCache` bernama `_memoryCache` dibuat. Cache diberi nama (`dataCache`) untuk mengidentifikasinya. Kemudian, cache menyimpan dan mengambil data.
2. `GetData`Metode ini adalah metode generik yang mengambil dua argumen: `string` kunci dan `Func<T>` delegasi dipanggil. `getData` Kunci digunakan untuk mengidentifikasi data cache, sedangkan `getData` delegasi mewakili logika pengambilan data yang dieksekusi ketika data tidak ada dalam cache.
3. Metode pertama memeriksa apakah data ada dalam cache menggunakan `_memoryCache.Contains(key)` metode ini. Jika data dalam cache, metode mengambil data dengan menggunakan `_memoryCache.Get(key)` dan mengirimkannya ke tipe yang diharapkan T.
4. Jika data tidak ada dalam cache, metode memanggil `getData` delegasi untuk mengambil data. Kemudian, ia menambahkan data ke cache dengan menggunakan`_memoryCache.Add(key,`

`data, DateTimeOffset.Now.AddMinutes(10))`. Panggilan ini menentukan bahwa entri cache harus kedaluwarsa setelah 10 menit, di mana data dihapus dari cache secara otomatis.

5. `ClearCache`Metode ini mengambil `string` kunci sebagai argumen dan menghapus data yang terkait dengan kunci itu dari cache dengan menggunakan`_memoryCache.Remove(key)`.

```
using System;
using System.Runtime.Caching;

public class InMemoryCache
{
    private static MemoryCache _memoryCache = new MemoryCache("dataCache");

    public static T GetData<T>(string key, Func<T> getData)
    {
        if (_memoryCache.Contains(key))
        {
            return (T)_memoryCache.Get(key);
        }

        T data = getData();
        _memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10));

        return data;
    }

    public static void ClearCache(string key)
    {
        _memoryCache.Remove(key);
    }
}
```

Anda dapat menggunakan kode berikut:

```
public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";

        Func<string> getSampleData = () =>
        {
```

```
            // Replace this with your data retrieval logic
            return "Sample data";
        };

        string data = InMemoryCache.GetData(cacheKey, getSampleData);
        Console.WriteLine("Data: " + data);
    }
}
```

Contoh berikut menunjukkan cara menggunakan LiteDB untuk menyimpan data di penyimpanan lokal. Anda dapat menggunakan LiteDB sebagai alternatif atau pelengkap caching dalam memori. Kode berikut menunjukkan cara menggunakan pustaka LiteDB untuk menyimpan data di penyimpanan lokal. `LocalStorageCache`Kelas berisi fungsi utama untuk mengelola cache.

```
using System;
using LiteDB;

public class LocalStorageCache
{
    private static string _liteDbPath = @"Filename=LocalCache.db";

    public static T GetData<T>(string key, Func<T> getData)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            var item = collection.FindOne(Query.EQ("_id", key));

            if (item != null)
            {
                return item;
            }
        }

        T data = getData();

        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            collection.Upsert(new BsonValue(key), data);
        }

        return data;
```

```
    }

    public static void ClearCache(string key)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection("cache");
            collection.Delete(key);
        }
    }
}

public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";

        Func<string> getSampleData = () =>
        {
            // Replace this with your data retrieval logic
            return "Sample data";
        };

        string data = LocalStorageCache.GetData(cacheKey, getSampleData);
        Console.WriteLine("Data: " + data);
    }
}
```

Jika Anda memiliki cache statis atau file statis yang tidak sering berubah, Anda juga dapat menyimpan file-file ini di penyimpanan objek Amazon Simple Storage Service (Amazon S3). Aplikasi dapat mengambil file cache statis saat startup untuk digunakan secara lokal. Untuk detail selengkapnya tentang cara mengambil file dari Amazon S3 menggunakan .NET, lihat Mengunduh objek dalam dokumentasi Amazon S3.

Caching dengan DAX

Anda dapat menggunakan lapisan caching yang dapat dibagikan di semua instance aplikasi. DynamoDB Accelerator (DAX) adalah cache dalam memori yang dikelola sepenuhnya dan sangat tersedia untuk DynamoDB yang dapat memberikan peningkatan kinerja sepuluh kali lipat. Anda dapat menggunakan DAX untuk mengurangi biaya dengan mengurangi kebutuhan untuk menyediakan unit

kapasitas baca yang berlebihan dalam tabel DynamoDB. Ini sangat berguna untuk beban kerja yang dibaca berat dan memerlukan pembacaan berulang untuk masing-masing kunci.

DynamoDB diberi harga sesuai permintaan atau dengan kapasitas yang disediakan, sehingga jumlah pembacaan dan penulisan per bulan berkontribusi terhadap biaya. Jika Anda telah membaca beban kerja yang berat, cluster DAX dapat membantu menurunkan biaya dengan mengurangi jumlah pembacaan pada tabel DynamoDB Anda. Untuk petunjuk tentang cara mengatur DAX, lihat Akselerasi dalam memori dengan DynamoDB Accelerator (DAX) dalam dokumentasi DynamoDB. Untuk informasi tentang integrasi aplikasi.NET, tonton Mengintegrasikan Amazon DynamoDB DAX ke dalam Aplikasi ASP.NET Anda di. YouTube

Sumber daya tambahan

- Akselerasi dalam memori dengan DynamoDB Accelerator (DAX) - Amazon DynamoDB (dokumentasi DynamoDB)
- Mengintegrasikan Amazon DynamoDB DAX ke dalam Aplikasi ASP.NET Anda () YouTube
- Mengunduh objek (dokumentasi Amazon S3)

Pertimbangkan .NET tanpa server

Ikhtisar

Komputasi tanpa server telah menjadi pendekatan populer untuk membangun dan menyebarkan aplikasi. Ini terutama karena skalabilitas dan kelincahan yang ditawarkan pendekatan tanpa server ketika membangun arsitektur modern. Namun, penting untuk mempertimbangkan dampak biaya komputasi tanpa server dalam beberapa skenario.

Lambda adalah platform komputasi tanpa server yang memungkinkan pengembang menjalankan kode tanpa memerlukan server khusus. Lambda adalah pilihan yang sangat menarik bagi pengembang.NET yang ingin mengurangi biaya infrastruktur. Dengan Lambda, pengembang .NET dapat mengembangkan dan menyebarkan aplikasi yang sangat skalabel dan berpotensi hemat biaya. Dengan menggunakan pendekatan tanpa server, pengembang tidak lagi menyediakan server untuk menangani permintaan aplikasi. Sebagai gantinya, pengembang dapat membuat fungsi yang dijalankan sesuai permintaan. Ini membuat pendekatan tanpa server lebih terukur, dapat dikelola, dan berpotensi lebih hemat biaya daripada menjalankan, mengelola, dan menskalakan mesin virtual. Akibatnya, Anda hanya membayar sumber daya yang digunakan oleh aplikasi, tanpa harus khawatir tentang sumber daya yang kurang dimanfaatkan atau biaya pemeliharaan server.

Pengembang dapat menggunakan versi .NET lintas platform modern untuk membangun aplikasi tanpa server yang cepat, efisien, dan hemat biaya. .NET Core dan versi yang lebih baru adalah framework gratis dan open-source yang lebih cocok untuk berjalan pada platform tanpa server dibandingkan versi.NET Framework sebelumnya. Hal ini memungkinkan pengembang untuk mengurangi waktu pengembangan dan meningkatkan kinerja aplikasi. .NET modern juga mendukung berbagai bahasa pemrograman, termasuk C # dan F #. Untuk alasan ini, ini adalah pilihan yang menarik bagi pengembang yang ingin membangun arsitektur modern di cloud.

Bagian ini menjelaskan bagaimana Anda dapat mencapai penghematan biaya dengan menggunakan Lambda sebagai opsi tanpa server. Anda dapat mengoptimalkan biaya lebih lanjut dengan menyempurnakan profil eksekusi fungsi Lambda Anda, mengukur alokasi memori fungsi Lambda Anda dengan benar, menggunakan AOT Asli, dan pindah ke fungsi berbasis Graviton.

Dampak biaya

Seberapa banyak Anda dapat mengurangi biaya tergantung pada beberapa faktor, termasuk berapa banyak eksekusi yang akan dijalankan oleh fungsi tanpa server Anda selain jumlah memori yang dialokasikan dan durasi setiap fungsi. AWS Lambda menawarkan tingkat gratis, yang mencakup satu juta permintaan gratis per bulan dan 400.000 GB detik waktu komputasi per bulan. Anda dapat secara signifikan mengurangi biaya bulanan Anda untuk beban kerja yang berada di atau di sekitar batas tingkat gratis ini.

Mungkin juga ada biaya tambahan saat menggunakan penyeimbang beban dengan fungsi Lambda sebagai target. Ini dihitung sebagai jumlah data yang diproses oleh penyeimbang beban untuk target Lambda.

Rekomendasi optimisasi biaya

Ukuran yang tepat fungsi Lambda Anda

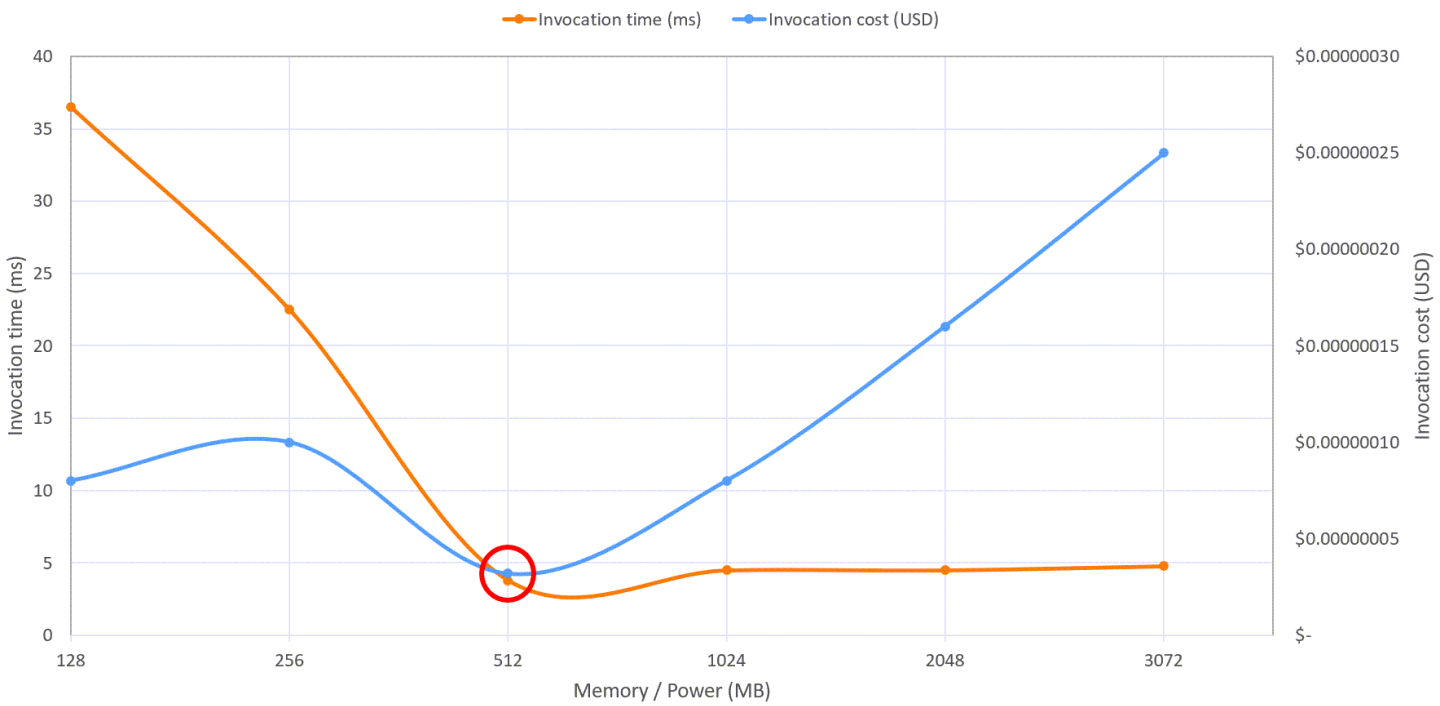
Ukuran yang tepat adalah praktik penting untuk pengoptimalan biaya dalam fungsi Lambda berbasis .net. Proses ini melibatkan identifikasi konfigurasi memori optimal yang menyeimbangkan kinerja dengan efektivitas biaya, tanpa memerlukan perubahan pada kode.

Dengan mengonfigurasi memori untuk fungsi Lambda, mulai dari 128 MB hingga 10.240 MB, Anda juga menyesuaikan jumlah vCPU yang tersedia selama pemanggilan. Hal ini memungkinkan memori atau aplikasi yang terikat CPU untuk mengakses sumber daya tambahan selama eksekusi, yang mengarah ke potensi pengurangan durasi pemanggilan dan biaya keseluruhan.

Namun, mengidentifikasi konfigurasi optimal untuk fungsi Lambda berbasis .NET Anda dapat menjadi proses manual dan intensif waktu, terutama jika perubahan sering terjadi. Alat AWS Lambda Power Tuning dapat membantu Anda mengidentifikasi konfigurasi yang sesuai dengan menganalisis satu set konfigurasi memori terhadap contoh payload.

Misalnya, meningkatkan memori untuk fungsi Lambda berbasis .NET dapat menyebabkan peningkatan total waktu pemanggilan dan mengurangi biaya tanpa mempengaruhi kinerja. Konfigurasi memori optimal untuk suatu fungsi dapat bervariasi. Alat AWS Lambda Power Tuning dapat membantu mengidentifikasi konfigurasi yang paling hemat biaya untuk setiap fungsi.

Dalam bagan contoh berikut, total waktu pemanggilan meningkat seiring dengan meningkatnya memori untuk fungsi Lambda ini. Hal ini menyebabkan pengurangan biaya untuk total eksekusi tanpa mempengaruhi kinerja asli fungsi. Untuk fungsi ini, konfigurasi memori optimal untuk fungsi tersebut adalah 512 MB, karena di sinilah pemanfaatan sumber daya paling efisien untuk total biaya setiap pemanggilan. Ini bervariasi per fungsi, dan menggunakan alat pada fungsi Lambda Anda dapat mengidentifikasi apakah mereka mendapat manfaat dari ukuran yang tepat.



Kami menyarankan Anda menyelesaikan latihan ini secara teratur, sebagai bagian dari pengujian integrasi apa pun saat pembaruan baru dirilis. Jika jarang diperbarui, maka lakukan latihan ini secara berkala untuk memastikan fungsi disetel dan berukuran tepat. Setelah Anda mengidentifikasi pengaturan memori yang sesuai untuk fungsi Lambda Anda, Anda dapat menambahkan ukuran yang tepat ke proses Anda. Alat AWS Lambda Power Tuning menghasilkan output terprogram yang

dapat digunakan oleh alur kerja CI/CD Anda selama rilis kode baru. Ini memungkinkan Anda untuk mengotomatiskan konfigurasi memori.

Anda dapat mengunduh alat AWS Lambda Power Tuning secara gratis. Untuk petunjuk tentang cara menggunakan alat ini, lihat Cara menjalankan mesin status di GitHub.

Lambda juga mendukung AOT asli, yang memungkinkan aplikasi.NET untuk dikompilasi sebelumnya. Ini dapat membantu mengurangi biaya dengan mengurangi waktu eksekusi untuk fungsi.NET. Untuk informasi selengkapnya tentang membuat fungsi AOT asli, lihat fungsi.NET dengan kompilasi AOT asli dalam dokumentasi Lambda.

Hindari waktu tunggu idle

Durasi fungsi Lambda adalah satu dimensi yang digunakan untuk menghitung penagihan. Ketika kode fungsi membuat panggilan pemblokiran, Anda ditagih untuk waktu yang menunggu untuk menerima respons. Waktu tunggu ini dapat tumbuh ketika fungsi Lambda dirantai bersama, atau fungsi bertindak sebagai orkestrator untuk fungsi lainnya. Jika Anda memiliki alur kerja seperti operasi batch atau sistem pengiriman pesanan, ini menambahkan overhead manajemen. Selain itu, tidak mungkin menyelesaikan semua logika alur kerja dan penanganan kesalahan dalam batas waktu Lambda maksimum 15 menit.

Alih-alih menangani logika ini dalam kode fungsi, kami sarankan Anda merancang ulang solusi Anda untuk digunakan AWS Step Functionssebagai orkestrator alur kerja. Saat menggunakan alur kerja standar, Anda ditagih untuk setiap transisi status dalam alur kerja, bukan total durasi alur kerja. Selain itu, Anda dapat memindahkan dukungan untuk percobaan ulang, kondisi tunggu, alur kerja kesalahan, dan panggilan balik ke kondisi status untuk memungkinkan fungsi Lambda Anda fokus pada logika bisnis. Untuk informasi selengkapnya, lihat Mengoptimalkan AWS Lambda biaya Anda — Bagian 2 di Blog AWS Komputasi.

Pindah ke fungsi berbasis Graviton

Fungsi Lambda yang didukung oleh prosesor Graviton2 generasi berikutnya sekarang tersedia secara umum. Fungsi Graviton2, menggunakan arsitektur prosesor berbasis ARM, dirancang untuk memberikan kinerja hingga 19 persen lebih baik dengan biaya 20 persen lebih rendah untuk berbagai beban kerja tanpa server. Dengan latensi yang lebih rendah dan kinerja yang lebih baik, fungsi yang ditenagai oleh prosesor Graviton2 ideal untuk memberi daya pada aplikasi tanpa server yang sangat penting.

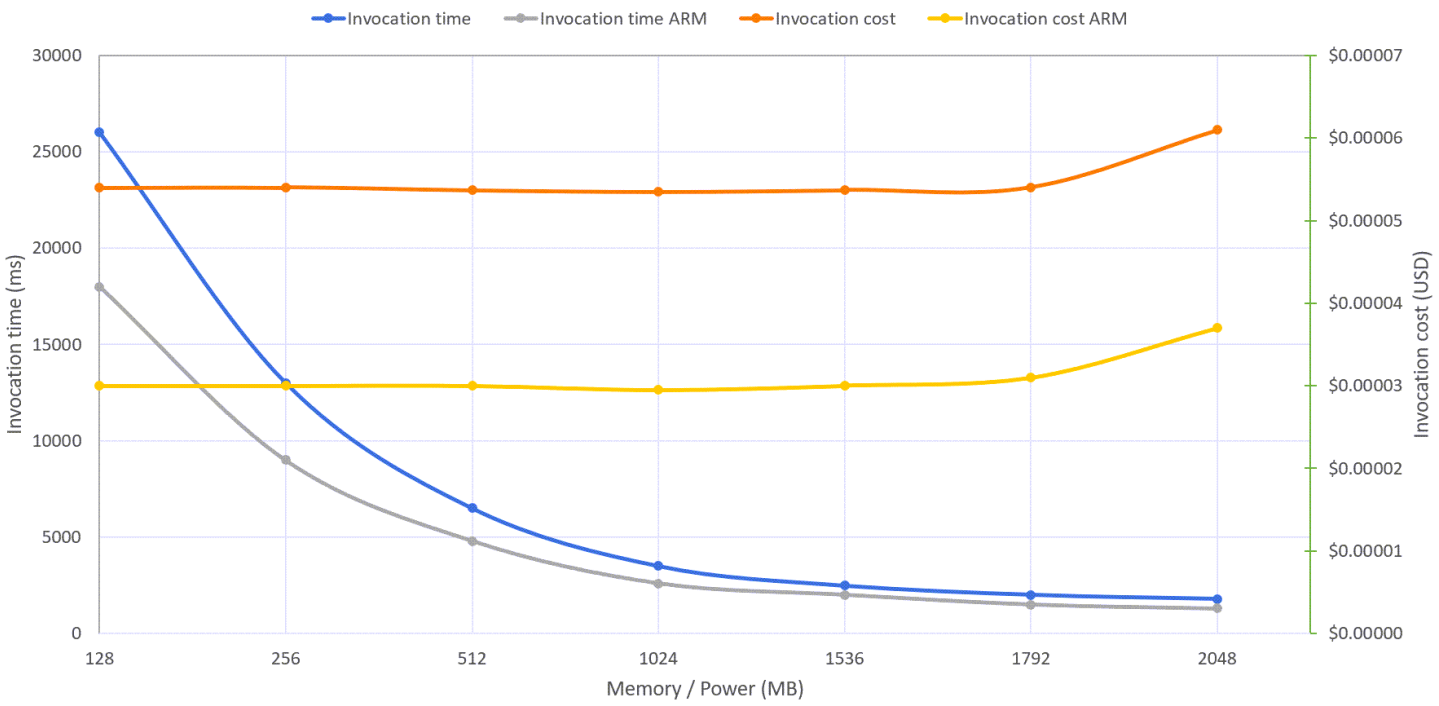
Migrasi ke fungsi Lambda berbasis Graviton dapat menjadi opsi hemat biaya bagi pengembang.NET yang ingin mengoptimalkan biaya Lambda mereka. Fungsi berbasis Graviton menggunakan prosesor

berbasis ARM, bukan prosesor x86 tradisional. Hal ini dapat menyebabkan penghematan biaya yang signifikan tanpa mengorbankan kinerja.

Meskipun ada beberapa manfaat untuk pindah ke fungsi berbasis Graviton, ada juga beberapa tantangan dan pertimbangan yang kami sarankan untuk Anda pertimbangkan. Misalnya, fungsi berbasis Graviton memerlukan penggunaan Amazon Linux 2, yang mungkin tidak kompatibel dengan semua aplikasi.NET. Selain itu, mungkin ada masalah kompatibilitas dengan pustaka atau dependensi pihak ketiga yang tidak kompatibel dengan prosesor berbasis ARM.

Jika Anda menjalankan aplikasi.NET Framework dan ingin memanfaatkan tanpa server dengan Lambda, Anda dapat mempertimbangkan porting aplikasi ke .NET modern dengan menggunakan Porting Assistant untuk.NET. Ini dapat membantu Anda mempercepat porting aplikasi .NET lama ke .NET modern, memungkinkan aplikasi berjalan di Linux.

Bagan berikut membandingkan hasil arsitektur x86 dan ARM/Graviton2 untuk fungsi yang menghitung bilangan prima.



Fungsinya menggunakan satu utas. Durasi terendah untuk kedua arsitektur dilaporkan ketika memori dikonfigurasi dengan 1,8 GB. Di atas itu, fungsi Lambda memiliki akses ke lebih dari 1 vCPU, tetapi dalam hal ini, fungsi tersebut tidak dapat menggunakan daya tambahan. Untuk alasan yang sama, biaya stabil dengan memori hingga 1, 8 GB. Dengan lebih banyak memori, biaya meningkat karena tidak ada manfaat kinerja tambahan untuk beban kerja ini. Prosesor Graviton2 jelas memberikan kinerja yang lebih baik dan biaya yang lebih rendah untuk fungsi komputasi intensif ini.

Untuk mengonfigurasi fungsi Anda untuk menggunakan dan prosesor berbasis ARM dengan Graviton, lakukan hal berikut:

1. Masuk ke Konsol Manajemen AWS dan kemudian buka konsol Lambda.
2. Pilih Buat fungsi.
3. Untuk nama Fungsi, masukkan nama.
4. Untuk Runtime, pilih .NET 6 (C #/PowerShell).
5. Untuk Arsitektur, pilih arm64.
6. Buat konfigurasi tambahan apa pun yang Anda butuhkan, lalu pilih Buat fungsi.

Sumber daya tambahan

- Lambda berfungsi sebagai target (dokumentasi)AWS
- Mengoptimalkan AWS Lambda biaya dan kinerja menggunakan AWS Compute Optimizer (AWS Compute Blog)
- Mengoptimalkan AWS Lambda biaya Anda - Bagian 1 (AWS Compute Blog)
- Mengoptimalkan AWS Lambda biaya Anda - Bagian 2 (AWS Compute Blog)
- Membangun aplikasi.NET tanpa server AWS Lambda menggunakan .NET 7 (AWS Compute Blog)

Pertimbangkan database yang dibuat khusus

Ikhtisar

Salah satu aspek paling mahal dalam menjalankan beban kerja berbasis Microsoft berasal dari lisensi database komersial, seperti SQL Server. Bisnis sering melakukan standarisasi pada SQL Server sebagai platform database pilihan dan menjadi mendarah daging dalam budaya pengembangan organisasi. Pengembang umumnya memilih model berbasis SQL Server relasional terlepas dari kasus penggunaannya. Alasan meliputi:

- Bisnis ini sudah memiliki and/or lisensi instans SQL Server yang tersedia.
- Tim telah terbiasa dengan model pemrograman SQL melalui penggunaan perpustakaan bersama, ORMs, dan logika bisnis.
- Manajemen tidak menyadari alternatif.
- Pengembang tidak mengetahui alternatif.

Database yang dibuat khusus dapat mengakomodasi pola akses data kasus penggunaan Anda. Basis data ini semakin diadopsi oleh bisnis karena mereka mengadopsi arsitektur yang lebih modern (seperti layanan mikro) dan sebagai ruang lingkup aplikasi individu yang sempit.

Database yang dibuat khusus tidak menghalangi model relasional, atau memerlukan model NoSQL (non-relasional). Bahkan, database relasional dianggap dibangun khusus ketika dipilih sebagai respons terhadap kebutuhan spesifik beban kerja. Penggunaan database yang dibuat khusus dapat membantu tim untuk mengurangi biaya database yang terkait dengan aplikasi .NET mereka, sementara juga mendapatkan manfaat cloud standar, seperti skalabilitas, ketahanan, dan pengurangan angkat berat yang tidak berdiferensiasi.

Tabel berikut menunjukkan basis data yang dibuat khusus yang ditawarkan oleh. AWS

Basis Data	Tipe	Karakteristik
Amazon Aurora PostgreSQL atau Amazon Aurora MySQL	Relasional	Gunakan kasus di mana data memiliki struktur tetap Database relasional secara alami menjaga konsistensi data melalui transaksi ACID
Amazon DynamoDB	Pasangan nilai kunci	Database NoSQL yang menyimpan data menggunak an struktur data tabel hash Penyimpanan kinerja tinggi dan pengambilan data tidak terstruktur Kasus penggunaan termasuk profil pengguna, status sesi, dan data keranjang belanja
Amazon ElastiCache	Dalam memori	Database NoSQL berkinerja tinggi yang menyimpan data tidak terstruktur dalam memori dengan waktu akses sub-milid etik

Basis Data	Tipe	Karakteristik
		Digunakan untuk data fana yang sering diakses seperti sesi pengguna, dan sebagai lapisan caching di depan penyimpanan data lain yang lebih lambat Termasuk dukungan untuk keduanya ElastiCache (Redis OSS) dan ElastiCache (Memcached)
Amazon MemoryDB	Tahan lama dalam memori	Database buatan khusus yang kompatibel dengan Redis dengan penyimpanan yang tahan lama
Amazon Timestream	Deret waktu	Database dirancang untuk konsumsi data throughput tinggi dalam urutan temporal Kasus penggunaan termasuk aplikasi Internet of Things (IoT) dan menyimpan metrik atau data telemetri
Amazon DocumentDB	Dokumen	Database NoSQL yang menyimpan data tanpa struktur yang ditentukan atau hubungan yang dipaksakan dengan data lain Sering digunakan untuk beban kerja intensif baca seperti katalog produk

Basis Data	Tipe	Karakteristik
Amazon Neptune	Grafik	Database NoSQL yang menyimpan data dan representasi koneksi antar item data Kasus penggunaan termasuk deteksi penipuan, mesin rekomendasi, dan aplikasi sosial
Keyspaces Amazon	Kolom lebar	Database terdistribusi kinerja tinggi berdasarkan Apache Cassandra Kasus penggunaan termasuk aplikasi IoT, pemrosesan acara, dan aplikasi game

Pendorong signifikan adopsi basis data yang dibangun khusus dapat dikaitkan dengan penghapusan lisensi komersial. Namun, kemampuan auto-scaling database seperti DynamoDB (termasuk mode sesuai permintaan), Aurora, Amazon Neptune, dan Amazon Keyspaces memungkinkan Anda menyediakan kapasitas untuk kasus rata-rata, bukan untuk penggunaan puncak. Database yang dibuat khusus, seperti Timestream, tanpa server dan secara otomatis menskalakan untuk memenuhi permintaan tanpa pra-penyediaan.

AWS menawarkan Babelfish untuk Aurora PostgreSQL jika Anda ingin menggunakan database relasional yang kompatibel dengan sumber terbuka yang dibuat khusus, tetapi tidak dapat atau tidak mau membuat perubahan kode yang signifikan pada aplikasi Anda. Dalam beberapa kasus, Babelfish memungkinkan Anda untuk menggunakan kode akses SQL Server yang ada, dengan hampir tidak ada perubahan.

Saat memilih database relasional yang dibuat khusus untuk aplikasi, penting untuk mempertahankan fitur yang sama (atau setara secara fungsional) yang Anda perlukan untuk aplikasi Anda. Rekomendasi ini membahas database yang dibangun khusus sebagai penyimpanan data utama untuk aplikasi. Aplikasi khusus (seperti caching) dibahas dalam rekomendasi lain.

Dampak biaya

Mengadopsi basis data yang dibuat khusus untuk beban kerja.NET, meskipun tidak mungkin mempengaruhi komputasi consumption/cost secara langsung, dapat secara langsung mempengaruhi biaya layanan database yang dikonsumsi oleh aplikasi.NET. Bahkan, penghematan biaya mungkin menjadi tujuan sekunder, jika dibandingkan dengan manfaat tambahan kelincahan, skalabilitas, ketahanan, dan daya tahan data.

Ini di luar cakupan panduan ini untuk menjelaskan proses penuh memilih database yang dibangun khusus untuk aplikasi dan merancang ulang strategi data untuk menggunakannya secara efektif. Untuk informasi selengkapnya, lihat Database yang dibuat khusus di Direktori Tutorial. AWS

Tabel berikut menunjukkan beberapa contoh bagaimana penggantian SQL Server dengan database yang dibangun khusus dapat mengubah biaya aplikasi. Perhatikan bahwa ini hanyalah perkiraan kasar. Benchmark dan optimalisasi beban kerja aktual diperlukan untuk menghitung biaya produksi yang tepat.

Ini adalah beberapa perkiraan basis data yang dibuat khusus yang umum digunakan yang mencakup komputasi sesuai permintaan dan 100 GB SSD, database instance tunggal di. `us-east-1` Biaya lisensi termasuk lisensi SQL Server ditambah jaminan perangkat lunak.

Tabel berikut menunjukkan perkiraan biaya untuk contoh database komersial.

Mesin database	Model perizinan	Jenis/spesifikasi instans	AWS menghitung + biaya penyimpanan	Biaya lisensi	Total biaya bulanan
SQL Server edisi Standar di Amazon EC2	Lisensi disertakan	r6i.2xbesar (8 CPU/64 GB RAM)	$1,345.36	$0,00	$1,345.36
Edisi SQL Server Enterprise di Amazon EC2	Lisensi disertakan	r6i.2xbesar (8 CPU/64 GB RAM)	$2.834,56	$0,00	$2.834,56

Mesin database	Model perizinan	Jenis/spesifikasi instans	AWS menghitung + biaya penyimpanan	Biaya lisensi	Total biaya bulanan
SQL Server edisi Standar di Amazon EC2	BYOL	r6i.2xbesar (8 CPU/64 GB RAM)	$644,56	$456.00	$1,100.56
Edisi SQL Server Enterprise di Amazon EC2	BYOL	r6i.2xbesar (8 CPU/64 GB RAM)	$644,56	$1,750.00	$2.394,56
SQL Server edisi Standar di Amazon RDS		db.r6i.2xlarge (8 CPU/64 GB RAM)	$2.318.30	$0,00	$2.318.30
Edisi SQL Server Enterprise di Amazon RDS		db.r6i.2xlarge (8 CPU/64 GB RAM)	$3.750,56	$0,00	$3.750,56

Tabel berikut menunjukkan perkiraan biaya untuk contoh yang dibuat khusus.

Mesin database	Jenis/spesifikasi instans	AWS menghitung + biaya penyimpanan	Biaya lisensi	Total biaya bulanan
Amazon Aurora PostgreSQL	r6g.2xbesar (8 CPU/64 GB RAM)	$855,87	$0,00	$855,87

Mesin database	Jenis/spesifikasi instans	AWS menghitung + biaya penyimpanan	Biaya lisensi	Total biaya bulanan
DynamoDB	Basis yang disediakan 100 WCU/400 RCU	$72,00		$72,00
Amazon DocumentDB	db.r6i.2xlarge (8 CPU/64 GB RAM)	$778,60		$778,60

> **Important**
>
> Tabel ini didasarkan pada perkiraan biaya lisensi untuk SQL Server dengan Jaminan Perangkat Lunak, selama tiga tahun pertama pembelian. Untuk SQL Server edisi Standar: $4,100, 2 paket inti, 3 tahun. Untuk edisi SQL Server Enterprise: $15.700, paket 2 inti, 3 tahun.

Kami menyarankan Anda mempertimbangkan implikasi biaya sebelum Anda mengadopsi basis data yang dibuat khusus. Misalnya, biaya untuk memperbarui aplikasi untuk menggunakan database yang dibangun khusus terkait dengan kompleksitas aplikasi dan basis data sumber. Pastikan untuk memperhitungkan total biaya kepemilikan saat merencanakan sakelar arsitektur ini. Ini termasuk memfaktorkan ulang aplikasi Anda, meningkatkan keterampilan staf tentang teknologi baru, dan merencanakan kinerja dan konsumsi yang diantisipasi dengan cermat untuk setiap beban kerja. Dari sana, Anda dapat menentukan apakah investasi tersebut sepadan dengan penghematan biaya. Dalam kebanyakan kasus, memelihara suatu end-of-support produk adalah risiko keamanan dan kepatuhan, dan biaya untuk memulihkannya sepadan dengan usaha dan investasi awal.

Rekomendasi optimisasi biaya

Untuk aplikasi.NET yang mengakses SQL Server, ada pustaka pengganti untuk database relasional yang dibangun khusus. Anda dapat mengimplementasikan pustaka ini dalam aplikasi Anda untuk menggantikan fungsionalitas aplikasi SQL Server yang serupa.

Tabel berikut menyoroti beberapa pustaka yang dapat digunakan dalam banyak skenario umum.

Perpustakaan	Basis Data	Penggantian untuk	Kompatibilitas kerangka kerja
Penyedia Inti Kerangka Entitas Npgsql	Amazon Aurora PostgreSQL	Kerangka Entitas Penyedia Server SQL Inti	.NET modern
Penyedia Kerangka Entitas Npgsql 6	Amazon Aurora PostgreSQL	Kerangka Entitas 6.0 Penyedia Server SQL	.NET Framework
Npgsql (perpustakaan PostgreSQL kompatibel ADO.NET)	Amazon Aurora PostgreSQL	ADO.NET	Framework/ Modern .NET
Penyedia Inti Kerangka Entitas MySQL	Amazon Aurora MySQL	Kerangka Entitas Penyedia Server SQL Inti	.NET modern
Pomelo. EntityFrameworkCore. MySql	Amazon Aurora MySQL	Kerangka Entitas Penyedia Server SQL Inti	.NET modern

Menghubungkan ke Amazon Aurora PostgreSQL dengan menggunakan Babelfish tidak memerlukan pengkodean khusus untuk terhubung. Namun, semua kode harus diuji secara menyeluruh sebelum digunakan.

Basis data lain yang dibuat khusus memiliki pustaka untuk mengakses pustaka yang kompatibel .NET yang memungkinkan Anda mengakses basis data yang dibuat khusus. Contohnya termasuk:

- Menggunakan database Amazon DynamoDB NoSQL (dokumentasi)AWS SDK untuk .NET
- MongoDB C # Driver (dokumentasi MongoDB)
- .NET (Dokumentasi Timestream)
- Menggunakan driver klien Cassandra .NET Core untuk mengakses Amazon Keyspaces secara terprogram (dokumentasi Amazon Keyspaces)
- Menggunakan .NET untuk terhubung ke instans DB Neptunus (dokumentasi Neptunus)

Jika Anda bermigrasi ke basis data bawaan tujuan, Anda dapat menggunakan alat ini AWS untuk membantu proses migrasi:

- AWS Schema Conversion Tool (AWS SCT) dapat membantu Anda mengubah skema SQL Server ke Amazon Aurora dan Amazon DynamoDB.
- AWS Database Migration Service (AWS DMS) dapat membantu Anda memigrasikan data, baik satu kali atau secara berkelanjutan, dari SQL Server ke Aurora atau DynamoDB.
- Babelfish Compass dapat membantu Anda memeriksa kompatibilitas database SQL Server Anda untuk digunakan dengan Babelfish untuk Aurora PostgreSQL.

Sumber daya tambahan

- Panduan untuk memigrasi SQL Server ke Amazon Aurora PostgreSQL (Blog Database)AWS
- Hari Perendaman Modernisasi APP Babelfish (Studio Lokakarya)AWS
- .NET Immersion Day (Studio AWS Lokakarya)
- Memulai Amazon Timestream dengan.NET () GitHub
- Database yang dibuat khusus untuk aplikasi.NET modern pada (presentasi) AWSAWS

Langkah berikutnya

Setelah Anda selesai meninjau panduan ini, kami sarankan Anda mengambil langkah-langkah berikut selanjutnya untuk menerapkan MACO:

1. Hubungi ahli MACO. Seorang ahli MACO dapat membantu menjawab pertanyaan Anda dan mengatasi masalah Anda. Jika Anda sudah bekerja dengan tim AWS akun, hubungi tim dan minta bantuan dari pakar MACO. Jika Anda tidak memiliki tim akun, hubungi optimize-microsoft@amazon.com.
2. Terapkan rekomendasi. Terapkan rekomendasi, praktik terbaik, dan strategi yang Anda pelajari dalam panduan ini dan dari berbicara dengan ahli MACO.
3. Lacak perubahan biaya.Tandai beban kerja Anda dan gunakan layanan seperti AWS Cost Explorer dan AWS Budgets untuk pelacakan, pemantauan, dan kontrol biaya terperinci.

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
Pembaruan SQL Server	Kami memperbarui bagian Optimalkan CPU untuk beban kerja SQL Server untuk menambahkan informasi selengkapnya tentang fitur Optimalkan CPU untuk instans Amazon EC2 .	Oktober 22, 2025
Pembaruan SQL Server	Kami memperbarui bagian Optimalkan CPU untuk beban kerja SQL Server.	Oktober 25, 2024
Pembaruan SQL Server dan Container	Kami menambahkan ukuran Optimalkan SQL Server dengan menggunakan Compute Optimizer, Trusted Advisor Meninjau rekomendasi untuk beban kerja SQL Server, dan aplikasi Windows Replatform dengan bagian App2Container.	Juni 29, 2024
Pengoptimalan lisensi SQL Server	Kami menambahkan lisensi Optimize SQL Server dengan menggunakan bagian Compute Optimizer.	22 Mei 2024
Publikasi awal	—	21 Desember 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 — Pindahkan aplikasi dan modifikasi 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 Edition. PostgreSQL-Compatible
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di. AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke. Salesforce.com
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di. AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

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

A

A2A () Agent-to-Agent

Protokol stateful untuk kolaborasi agen-ke-agen yang mendukung delegasi tugas dan transfer negara.

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.

Agen

Sistem AI yang dapat secara mandiri bernalar, merencanakan, dan mengambil tindakan menggunakan alat untuk mencapai tujuan.

Agen Ops

Praktik operasional untuk membangun, menguji, menyebarkan, dan menjalankan agen AI dalam produksi dalam skala besar.

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.

blue/green penyebaran

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. AWS Well-Architected

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.

Pengembang Warga

Pengguna bisnis yang membuat aplikasi AI menggunakan platform tanpa code/low kode tanpa keterampilan teknis khusus.

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 AWS Cloud Perusahaan. Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat panduan kesiapan migrasi.

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori dikhususkan untuk satu bagian fungsionalitas. Satu CI/CD pipa dapat menggunakan beberapa repositori.

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat bahasa definisi database.

ansambel yang dalam

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

pembelajaran mendalam

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

pertahanan-mendalam

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, pendekatan defense-in-depth mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi 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 setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

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

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat Apa itu Pertukaran Data Elektronik.

enkripsi

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

kunci enkripsi

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

endianness

Urutan byte disimpan dalam memori komputer. Big-endian sistem menyimpan byte paling signifikan terlebih dahulu. Little-endian sistem menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.

- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang dapat diakses oleh pengguna akhir. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.
- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

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

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

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

FM

Lihat model pondasi.

model pondasi (FM)

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

Gerbang FM

Perantara terpusat yang mengontrol dan menormalkan akses ke model pondasi. Juga dikenal sebagai gateway LLM.

G

AI generatif

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

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

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

strategi greenfield

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

pagar pembatas

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

pagar pembatas (AI)

Mekanisme keamanan yang menyaring, memvalidasi, dan membatasi input dan output agen untuk membantu memastikan perilaku AI yang bertanggung jawab dan aman.

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

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

manusia-dalam-lingkaran (HiTL)

Pola alur kerja di mana eksekusi agen berhenti untuk peninjauan dan persetujuan manusia pada titik keputusan kritis.

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 in the Framework. AWS Well-Architected

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 lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

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

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

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

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

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

cabang utama

Lihat cabang.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

MCP

Lihat Protokol Konteks Model.

Protokol Konteks Model (MCP)

Protokol stateless untuk komunikasi agen -to- alat.

Server MCP

Layanan yang mengekspos satu atau lebih alat melalui Protokol Konteks Model.

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 selengkapnya, lihat Membangun mekanisme dalam AWS Well-Architected Kerangka Kerja.

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 mesin-ke-mesin (M2M) yang ringan, berdasarkan pola publish/subscribe, 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

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat pembelajaran mesin.

modernisasi

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

penilaian kesiapan modernisasi

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

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat Menguraikan 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 mesin-ke-mesin (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 selengkapnya, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja. AWS Well-Architected

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 `PUT` dan `DELETE` permintaan ke bucket S3.

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.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

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

predikat pushdown

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

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona 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.

rantai cepat

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

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat Retrieval Augmented Generation.

ransomware

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

Matriks RASCI

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

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

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

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat 7 Rs.

Region

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

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

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

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

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

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations 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.

Bayangan AI

Aplikasi AI yang tidak sah dibuat atau digunakan di luar saluran yang diatur dalam suatu organisasi.

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.

model split-and-seed

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 ASP.NET Microsoft (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

Key-value pasangan yang bertindak sebagai metadata untuk mengatur sumber daya Anda AWS . Tag membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai sumber daya AWS.

variabel target

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

daftar tugas

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

lingkungan uji

Lihat lingkungan.

pelatihan

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

alat

Fungsi atau API yang dapat dipanggil agen untuk melakukan operasi di sistem eksternal.

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.

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 AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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