Kerangka Kerja AWS Well-Architected

Pilar Keberlanjutan

Pilar Keberlanjutan: Kerangka Kerja AWS Well-Architected

Table of Contents

Pilar Berkelanjutan - Kerangka Kerja AWS Well-Architected

Tanggal publikasi: 27 Juni 2024 (Revisi dokumen)

Laporan resmi ini berfokus pada pilar keberlanjutan Kerangka Kerja Amazon Web Services (AWS) Well-Architected. Laporan ini memberikan prinsip desain, panduan operasional, praktik terbaik, potensi kompensasi, dan rencana peningkatan yang dapat Anda gunakan untuk memenuhi target keberlanjutan untuk beban kerja AWS Anda.

Pengantar

Kerangka Kerja AWS Well-Architected akan membantu Anda mengetahui kelebihan dan kekurangan dari keputusan yang Anda ambil saat membangun beban kerja di AWS. Kerangka Kerja ini akan membantu Anda mempelajari praktik-praktik terbaik berkaitan dengan arsitektur untuk mendesain dan mengoperasikan beban kerja yang aman, andal, efisien, hemat biaya, dan ramah lingkungan di AWS Cloud. Kerangka Kerja ini menyediakan cara untuk secara terus menerus menilai arsitektur Anda berdasarkan praktik-praktik terbaik dan mengidentifikasi area yang perlu diperbaiki. Dengan memiliki beban kerja yang dirancang dengan baik, kemampuan Anda untuk mendukung hasil bisnis dapat meningkat pesat.

Enam pilar landasan Kerangka Kerja:

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

Dokumen ini berfokus pada pilar keenam, yakni menitikberatkan pada cakupan keberlanjutan. Dokumen ini dimaksudkan untuk orang-orang yang memiliki peran di bidang teknologi, seperti kepala pejabat teknologi (CTO), arsitek, pengembang, dan para anggota tim operasi.

Setelah membaca dokumen ini, Anda akan memahami saran dan strategi AWS terkini yang bisa digunakan ketika merancang arsitektur cloud dengan mempertimbangkan keberlanjutan. Dengan

mengadopsi praktik-praktik yang diuraikan dalam laporan ini, Anda dapat membangun arsitektur yang memaksimalkan efisiensi dan mengurangi limbah.

Keberlanjutan cloud

Bidang keberlanjutan membahas tentang dampak jangka panjang aktivitas bisnis Anda terhadap lingkungan, ekonomi, dan masyarakat. Komisi Lingkungan dan Pembangunan PBB mendefinisikan pembangunan berkelanjutan sebagai “pembangunan yang memenuhi kebutuhan saat ini tanpa mengorbankan kemampuan generasi mendatang untuk memenuhi kebutuhan mereka sendiri.” Bisnis atau organisasi Anda dapat memberikan dampak negatif terhadap lingkungan seperti emisi karbon langsung atau tidak langsung, limbah yang tidak dapat didaur ulang, dan kerusakan yang dapat terjadi pada sumber daya bersama, misalnya air bersih.

Saat membangun beban kerja cloud, praktik keberlanjutannya adalah memahami dampak layanan yang digunakan, menghitung dampak melalui seluruh siklus hidup beban kerja, dan menerapkan prinsip-prinsip desain dan praktik terbaik untuk mengurangi dampak-dampak ini. Dokumen ini berfokus pada dampak-dampak lingkungan, terutama konsumsi dan efisiensi energi, karena ini merupakan pendorong penting bagi arsitek untuk melandasi tindakan-tindakan langsung mereka untuk mengurangi penggunaan sumber daya.

Saat berfokus pada dampak lingkungan, Anda harus memahami bagaimana biasanya dampak-dampak ini dipertimbangkan serta dampak-dampak susulan terhadap pengukuran emisi organisasi Anda sendiri. Protokol Gas Rumah Kaca mengatur emisi karbon ke dalam cakupan-cakupan berikut ini, beserta contoh-contoh emisi untuk tiap-tiap cakupan yang relevan bagi penyedia cloud seperti AWS:

- Lingkup 1: Semua emisi langsung dari aktivitas suatu organisasi atau di bawah kendalinya. Sebagai contoh, pembakaran gas oleh generator cadangan pusat data.
- Lingkup 2: Emisi tidak langsung dari listrik yang dibeli dan digunakan untuk menghidupkan pusat data dan fasilitas lain. Contohnya adalah emisi dari pembangkit daya komersial.
- Lingkup 3: Semua emisi tidak langsung lainnya yang berasal dari aktivitas suatu organisasi dari sumber yang tidak dikendalikannya. Contoh AWS antara lain emisi yang berhubungan dengan pembangunan pusat data, dan produksi serta pengangkutan perangkat keras IT yang di-deploy di pusat data.

Dari sudut pandang pelanggan AWS, emisi dari beban kerja Anda yang berjalan di AWS dianggap sebagai emisi tidak langsung, dan bagian dari emisi Lingkup 3 Anda. Tiap-tiap beban kerja yang di-deploy menghasilkan sebagian dari total emisi AWS dari tiap-tiap lingkup sebelumnya. Jumlah yang sebenarnya berbeda-beda untuk setiap beban kerja dan bergantung pada sejumlah faktor termasuk

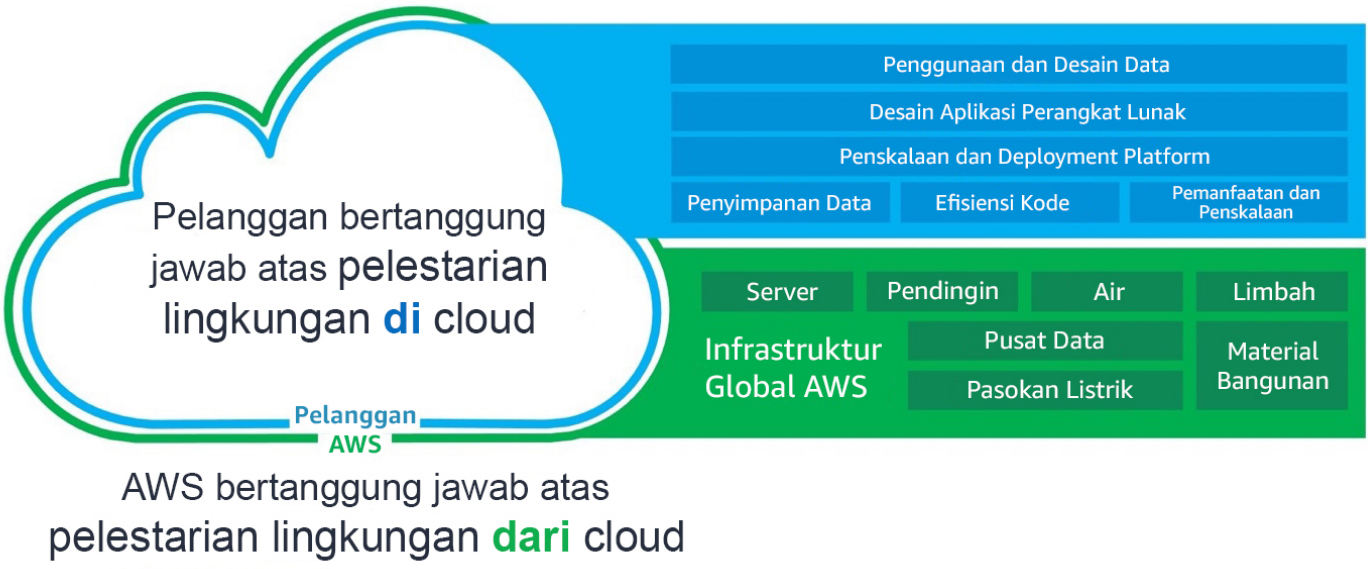
layanan-layanan AWS yang digunakan, energi yang dipakai oleh layanan-layanan tersebut, intensitas karbon jaringan listrik yang mengaliri pusat data AWS tempatnya berjalan, dan pengadaan energi terbarukan oleh AWS.

Dokumen ini terlebih dahulu menjelaskan model tanggung jawab bersama untuk keberlanjutan, lalu menyediakan praktik terbaik arsitektur sehingga Anda dapat meminimalkan dampak beban kerja Anda dengan mengurangi total sumber daya yang diperlukan beban kerja untuk berjalan di pusat data AWS.

Model tanggung jawab bersama

Keberlanjutan adalah tanggung jawab bersama antara pelanggan dan AWS.

- AWS bertanggung jawab untuk mengoptimalkan keberlanjutan cloud - dengan menyediakan infrastruktur bersama yang efisien, penatagunaan air, dan pengadaan energi terbarukan.
- Pelanggan bertanggung jawab untuk keberlanjutan pada cloud - dengan mengoptimalkan pemanfaatan beban kerja dan sumber daya, dan meminimalkan total sumber daya yang diperlukan untuk di-deploy untuk beban kerja Anda.



Model tanggung jawab bersama

Keberlanjutan cloud

Penyedia cloud memiliki jejak karbon yang lebih rendah dan lebih hemat energi daripada alternatif on-premise biasa karena penyedia cloud berinvestasi dalam teknologi energi dan pendingin yang efisien, mengoperasikan kumpulan server yang hemat energi, dan mencapai tingkat pemanfaatan server yang tinggi. Beban kerja cloud engurangi dampak dengan cara memanfaatkan sumber daya bersama, seperti jaringan, daya, pendingin, dan fasilitas fisik. Anda dapat memigrasikan beban kerja cloud Anda ke teknologi yang lebih efisien jika tersedia dan menggunakan layanan berbasis cloud untuk mentransformasikan beban kerja Anda demi keberlanjutan yang lebih baik.

Sumber daya

- Peluang Pengurangan Karbon dengan Berpindah ke Amazon Web Services
- AWS menghadirkan solusi keberlanjutan

Keberlanjutan di cloud

Keberlanjutan di cloud adalah sebuah upaya berkelanjutan yang difokuskan terutama pada pengurangan dan efisiensi energi di semua komponen beban kerja dengan mencapai manfaat maksimum dari sumber daya yang disediakan dan meminimalkan total sumber daya yang diperlukan. Upaya ini bisa mencakup pemilihan bahasa pemrograman yang efisien di awal, adopsi algoritme modern, penggunaan teknik penyimpanan data yang efisien, deployment ke infrastruktur komputasi yang efisien dan terukur dengan tepat, serta meminimalkan kebutuhan perangkat keras pengguna akhir yang berdaya tinggi.

Keberlanjutan melalui cloud

Selain meminimalkan dampak beban kerja yang telah Anda deploy, Anda juga dapat menggunakan AWS Cloud untuk menjalankan beban kerja yang dirancang untuk mendukung tantangan keberlanjutan yang lebih luas yang Anda hadapi. Contoh tantangan ini antara lain adalah pengurangan emisi karbon, penurunan konsumsi energi, daur ulang air, atau pengurangan limbah di area lain dalam bisnis atau organisasi Anda.

Keberlanjutan melalui cloud adalah ketika Anda menggunakan teknologi AWS untuk menyelesaikan tantangan keberlanjutan yang lebih luas. Misalnya, Anda dapat menggunakan layanan machine learning seperti Amazon Monitron untuk mendeteksi perilaku abnormal dalam mesin industri. Dengan data deteksi ini, Anda dapat melakukan pemeliharaan preventif untuk mengurangi risiko terjadinya

insiden lingkungan yang disebabkan oleh kegagalan peralatan yang tidak terduga dan memastikan mesin dapat terus beroperasi dengan efisiensi optimal.

Prinsip-prinsip desain untuk keberlanjutan di cloud

Terapkan prinsip-prinsip desain ini saat merancang beban kerja cloud Anda guna memaksimalkan keberlanjutan dan meminimalkan dampak-dampak yang ditimbulkannya.

- Pahami dampak Anda: Ukur dampak beban kerja cloud Anda dan buat model dampak beban kerja Anda untuk masa mendatang. Sertakan semua sumber dampak, termasuk dampak akibat penggunaan produk Anda oleh pelanggan, serta dampak yang muncul dari penonaktifan dan penghentian produk. Bandingkan output produktif dengan total dampak beban kerja cloud Anda dengan meninjau sumber daya dan emisi yang diperlukan per unit kerja. Gunakan data ini untuk membuat indikator kinerja utama (KPI), dan untuk mengevaluasi cara-cara yang digunakan untuk meningkatkan produktivitas sekaligus mengurangi dampak yang ditimbulkan, serta memperkirakan dampak yang ditimbulkan oleh perubahan-perubahan yang diajukan seiring waktu.
- Tetapkan tujuan keberlanjutan: Untuk tiap-tiap beban kerja cloud, tetapkan tujuan keberlanjutan jangka panjang seperti mengurangi sumber daya komputasi dan penyimpanan yang diperlukan per transaksi. Buat pemodelan laba atas investasi peningkatan keberlanjutan untuk beban kerja yang ada, dan beri pemilik sumber daya yang mereka perlukan untuk berinvestasi dalam tujuan keberlanjutan. Rencanakan pertumbuhan, dan rancang beban kerja Anda agar pertumbuhan menghasilkan penurunan intensitas dampak yang terukur berdasarkan unit yang tepat, seperti per pengguna atau per transaksi. Tujuan ini membantu Anda mendukung tujuan keberlanjutan yang lebih luas untuk bisnis atau organisasi Anda, mengidentifikasi regresi, dan memprioritaskan area-area dengan potensi perbaikan.
- Maksimalkan pemanfaatan: Sesuaikan ukuran beban kerja dan implementasikan desain yang efisien untuk memastikan pemanfaatan yang tinggi dan memaksimalkan efisiensi energi untuk perangkat keras yang digunakan. Dua host yang berjalan dengan pemanfaatan 30% memiliki efisiensi yang lebih rendah daripada satu host dengan pemanfaatan 60% dikarenakan konsumsi daya dasar per host. Pada saat yang sama, singkirkan atau minimalkan sumber daya, pemrosesan, dan penyimpanan yang tidak aktif untuk mengurangi total energi yang diperlukan untuk menjalankan beban kerja Anda.
- Antisipasi dan adopsi penawaran perangkat keras dan perangkat lunak baru yang lebih efisien: Dukung peningkatan hulu yang dilakukan mitra dan pemasok Anda untuk membantu Anda mengurangi dampak beban kerja cloud Anda. Terus pantau dan evaluasi penawaran perangkat

keras dan perangkat lunak baru yang lebih efisien. Rancang desain yang fleksibel untuk memungkinkan pengadopsian teknologi efisien baru secara cepat.

- Gunakan layanan terkelola: Berbagi layanan di seluruh basis pelanggan yang luas dapat membantu memaksimalkan pemanfaatan sumber daya, sehingga mengurangi jumlah infrastruktur yang diperlukan untuk mendukung beban kerja cloud. Sebagai contoh, pelanggan dapat berbagi dampak komponen pusat data yang sama seperti daya dan jaringan dengan memigrasikan beban kerja ke AWS Cloud dan mengadopsi layanan terkelola, seperti AWS Fargate untuk kontainer nirserver, tempat AWS beroperasi pada skala besar dan bertanggung jawab atas efisiensi operasi mereka. Gunakan layanan terkelola yang dapat membantu meminimalkan dampak Anda, seperti memindahkan data yang jarang diakses ke penyimpanan dingin secara otomatis dengan menggunakan konfigurasi Amazon S3 Lifecycle atau Amazon EC2 Auto Scaling untuk menyesuaikan kapasitas guna memenuhi permintaan.
- Kurangi dampak hilir beban kerja cloud Anda: Kurangi jumlah energi atau sumber daya yang diperlukan untuk menggunakan layanan-layanan Anda. Kurangi atau singkirkan keharusan pelanggan untuk meningkatkan perangkat mereka hanya untuk bisa menggunakan layanan Anda. Uji menggunakan device farm untuk memahami dampak yang diperkirakan dan uji dengan pelanggan untuk memahami dampak riil dari penggunaan layanan Anda.

Proses peningkatan

Proses peningkatan arsitektur mencakup pemahaman tentang apa yang Anda miliki dan apa yang dapat Anda lakukan untuk melakukan peningkatan, menyeleksi target peningkatan, menguji peningkatan, mengadopsi peningkatan yang berhasil, menghitung keberhasilan, dan membagikan pelajaran yang Anda dapatkan sehingga dapat direplikasi di tempat lain, lalu mengulangi siklus ini.

Tujuan-tujuan peningkatan Anda antara lain:

- Untuk menghilangkan pemborosan, pemanfaatan yang rendah, dan sumber daya yang tidak aktif atau tidak digunakan
- Untuk memaksimalkan nilai dari sumber daya yang Anda gunakan

> **Note**
>
> Gunakan semua sumber daya yang Anda sediakan, dan selesaikan tugas yang sama dengan sumber daya yang seminimal mungkin.

Pada tahap-tahap awal optimalisasi, terlebih dahulu singkirkan area dengan pemborosan atau pemanfaatan yang rendah, lalu beralihlah ke optimalisasi yang lebih tertarget yang sesuai dengan beban kerja khusus Anda.

Lakukan pemantauan terhadap perubahan-perubahan yang terjadi pada pemakaian sumber daya dari waktu ke waktu. Identifikasi tempat di mana perubahan terkumpul yang mengakibatkan peningkatan pemakaian sumber daya yang tidak efisien atau terlalu besar. Tentukan kebutuhan peningkatan untuk menangani perubahan-perubahan yang terjadi pada pemakaian dan implementasikan peningkatan yang Anda prioritaskan.

Langkah-langkah berikut ini dirancang sebagai proses iteratif yang mengevaluasi, memprioritaskan, menguji, dan melakukan deployment peningkatan untuk beban kerja cloud yang berfokus pada keberlanjutan.

1. Identifikasi sasaran peningkatan: Tinjau beban kerja Anda dengan mengacu praktik terbaik untuk keberlanjutan yang disebutkan dalam dokumen ini, dan identifikasi target-target peningkatan.
2. Evaluasi peningkatan spesifik: Evaluasi perubahan-perubahan khusus untuk mengetahui potensi peningkatan, biaya yang diperkirakan, dan risiko bisnis.

3. Prioritaskan dan rencanakan peningkatan: Prioritaskan perubahan yang menawarkan peningkatan terbesar dengan biaya dan risiko terkecil, dan buat rencana pengujian dan implementasi.
4. Uji dan validasi peningkatan: Implementasikan perubahan di lingkungan pengujian untuk memvalidasi potensi peningkatannya.
5. Menerapkan perubahan ke produksi: Menerapkan perubahan di seluruh lingkungan produksi.
6. Ukur hasil dan replikasi keberhasilan: Cari peluang untuk mereplikasi keberhasilan di beban kerja, dan batalkan perubahan dengan hasil yang tidak dapat diterima.

Contoh skenario

Contoh skenario berikut ini akan disebutkan di bagian lain dokumen ini untuk menggambarkan tiap-tiap langkah dari proses peningkatan.

Perusahaan Anda memiliki beban kerja yang menjalankan manipulasi gambar kompleks di instans Amazon EC2 dan menyimpan file asli dan yang telah dimodifikasi agar dapat diakses pengguna. Aktivitas-aktivitas pemrosesannya memerlukan CPU yang besar, dan file output-nya amat sangat besar.

Identifikasi target peningkatan

Pahami praktik-praktik terbaik yang dapat membantu Anda mencapai tujuan keberlanjutan Anda. Anda dapat menemukan deskripsi terperinci tentang praktik terbaik ini dan rekomendasi untuk perbaikan nanti dalam dokumen ini.

Tinjau beban kerja Anda serta sumber daya yang digunakan. Identifikasi hot spot seperti deployment yang besar dan sumber daya yang sering digunakan. Evaluasi area-area penting ini untuk mengetahui adanya peluang peningkatan pemanfaatan yang efektif pada sumber daya Anda dan untuk mengurangi sumber daya total yang diperlukan untuk mencapai hasil bisnis Anda.

Tinjau beban kerja Anda dengan mengacu pada praktik terbaik, dan identifikasi target-target peningkatan.

Dengan menerapkan langkah ini ke Contoh skenario, Anda mengidentifikasi praktik-praktik terbaik berikut ini sebagai potensi target peningkatan:

- Gunakan perangkat keras dalam jumlah minim untuk memenuhi kebutuhan Anda
- Gunakan teknologi yang mendukung pola akses dan penyimpanan data

Sumber daya

- Mengoptimalkan Infrastruktur AWS untuk Keberlanjutan, Bagian I: Komputasi
- Mengoptimalkan Infrastruktur AWS Anda untuk Keberlanjutan, Bagian II: Penyimpanan
- Mengoptimalkan Infrastruktur AWS Anda untuk Keberlanjutan, Bagian III: Jaringan

Evaluasi peningkatan khusus

Pahami sumber daya yang disediakan oleh beban kerja Anda untuk menyelesaikan sebuah unit kerja. Evaluasi potensi peningkatan, dan perkirakan potensi dampaknya, biaya untuk mengimplementasikannya, dan risiko-risiko terkait.

Untuk mengukur peningkatan dari waktu ke waktu, terlebih dahulu pahami apa yang telah Anda sediakan di AWS dan pahami juga bagaimana sumber daya tersebut digunakan.

Mulailah dengan gambaran umum penuh tentang penggunaan AWS Anda, dan gunakan Laporan Biaya dan Penggunaan AWS untuk membantu Anda mengidentifikasi area-area penting. Gunakan kode contoh AWS ini untuk membantu Anda meninjau dan menganalisis laporan Anda dengan bantuan Amazon Athena.

Metrik proksi

Ketika Anda melakukan evaluasi atas perubahan tertentu, evaluasi juga metrik apa saja yang paling mewakili efek perubahan tersebut pada sumber daya terkait. Metrik-metrik ini disebut metrik proxy. Pilih metrik-mterik proksi yang paling mencerminkan tipe peningkatan yang sedang Anda evaluasi serta sumber daya yang ditargetkan oleh peningkatan. Metrik-metrik ini mungkin berkembang dari waktu ke waktu.

Sumber daya yang disediakan untuk mendukung beban kerja Anda mencakup sumber daya komputasi, penyimpanan, dan jaringan. Evaluasi sumber daya yang disediakan menggunakan metrik-metrik proksi Anda untuk melihat bagaimana sumber daya tersebut digunakan.

Gunakan metrik-metrik proksi Anda untuk mengukur sumber daya yang disediakan untuk mencapai hasil bisnis.

Sumber Daya	Contoh metrik proksi	Tujuan peningkatan
Hitung	Menit vCPU	Memaksimalkan pemanfaatan sumber daya yang disediakan
Penyimpanan	GB yang disediakan	Mengurangi total yang disediakan
Jaringan	GB yang ditransfer atau paket yang ditransfer	Mengurangi total yang ditransfer dan jarak yang ditransfer

Metrik bisnis

Pilih metrik-metrik bisnis untuk menghitung pencapaian hasil bisnis. Metrik-metrik bisnis Anda harus mencerminkan nilai yang disediakan oleh beban kerja Anda, misalnya jumlah pengguna aktif dalam waktu yang sama, panggilan API yang dilayani, atau jumlah transaksi yang diselesaikan. Metrik-metrik ini mungkin berkembang dari waktu ke waktu. Hati-hatilah ketika Anda mengevaluasi metrik-metrik bisnis berbasis keuangan, karena ketidaksesuaian pada nilai transaksi akan menjadikan perbandingan menjadi tidak valid.

Indikator kinerja utama

Menggunakan rumus berikut ini, bagi sumber daya yang disediakan dengan hasil bisnis yang dicapai untuk menentukan sumber daya yang disediakan untuk setiap unit kerja.

$$\textit{Sumber daya yang disediakan per unit kerja} = \frac{\textit{Metrik proksi untuk sumber daya yang disediakan}}{\textit{Metrik bisnis untuk hasil}}$$

Rumus KPI

Gunakan sumber daya Anda per unit kerja sebagai KPI Anda. Buat garis acuan berdasarkan sumber daya yang disediakan sebagai dasar perbandingan.

Sumber Daya	Contoh KPI	Tujuan peningkatan
Hitung	menit vCPU per transaksi	Memaksimalkan pemanfaatan sumber daya yang disediakan
Penyimpanan	GB per transaksi	Mengurangi total yang disediakan
Jaringan	GB yang ditransfer untuk setiap transaksi atau paket yang ditransfer per transaksi	Mengurangi total yang ditransfer dan jarak yang ditransfer

Menghitung peningkatan

Hitung peningkatan sebagai penurunan kuantitatif pada sumber daya yang disediakan (seperti yang ditunjukkan oleh metrik proksi Anda) serta perubahan persentase dari sumber daya acuan Anda yang disediakan untuk setiap unit kerja.

Sumber Daya	Contoh KPI	Tujuan peningkatan
Hitung	Persentase (%) penurunan menit vCPU per transaksi	Maksimalkan pemanfaatan
Penyimpanan	Persentase (%) penurunan GB per transaksi	Mengurangi total yang disediakan
Jaringan	Persentase (%) penurunan GB yang ditransfer per transaksi atau paket yang ditransfer per transaksi	Mengurangi total yang ditransfer dan jarak yang ditransfer

Evaluasi peningkatan

Lakukan evaluasi terhadap potensi peningkatan berdasarkan manfaat bersih yang diharapkan. Lakukan evaluasi terhadap waktu, biaya, dan tingkat upaya untuk mengimplementasikan dan memelihara, serta risiko-risiko bisnis, misalnya dampak yang tidak terduga.

Peningkatan-peningkatan yang ditargetkan sering kali mewakili kompromi antar tipe sumber daya yang dipakai. Sebagai contoh, untuk mengurangi pemakaian komputasi, Anda dapat menyimpan sebuah hasil, atau untuk membatasi data yang ditransfer, Anda dapat memproses data sebelum mengirimkan hasilnya kepada seorang klien. Kompensasi ini dibahas di detail tambahan yang akan datang.

Sertakan persyaratan non-fungsional ketika mengevaluasi risiko untuk beban kerja Anda, termasuk keamanan, keandalan, efisiensi kinerja, optimalisasi biaya, dan dampak peningkatan terhadap kemampuan Anda untuk mengoperasikan beban kerja.

Dengan menerapkan langkah ini ke Contoh skenario, Anda mengevaluasi target peningkatan dengan hasil-hasil berikut ini:

Praktik terbaik	Peningkatan yang ditargetkan	Potensi	Biaya	Risiko
Gunakan perangkat keras dalam jumlah minim untuk memenuhi kebutuhan Anda	Implementasikan penskalaan prediktif untuk mengurangi periode pemanfaatan yang rendah	Sedang	Rendah	Rendah
Gunakan teknologi yang mendukung pola akses dan penyimpanan data	Implementasikan mekanisme-mekanisme kompresi yang lebih efektif untuk mengurangi total penyimpanan dan waktu untuk mencapainya	Tinggi	Rendah	Rendah

Implementasi penskalaan prediktif dapat mengurangi jam vCPU yang digunakan oleh instans dengan pemanfaatan rendah atau yang tidak digunakan yang menyediakan manfaat sedang dibandingkan

mekanisme penskalaan yang ada dengan estimasi penurunan sumber daya yang digunakan sebesar 11%. Biaya yang terlibat jumlahnya rendah dan mencakup konfigurasi sumber daya cloud dan operasi penskalaan prediktif untuk Amazon EC2 Auto Scaling. Risikonya adalah kinerja yang dibatasi ketika dilakukan penambahan skala (scale-out) secara reaktif untuk merespons permintaan yang melampaui prediksi.

Implementasi kompresi yang lebih efektif dapat memiliki dampak signifikan dengan penurunan yang besar dalam hal ukuran file di semua gambar asli dan manipulasi Anda, dengan estimasi penurunan kebutuhan penyimpanan sebesar 25% di lingkungan produksi. Menimplementasikan algoritme baru adalah pengganti yang mudah dan memiliki sedikit risiko.

Prioritaskan dan rencanakan peningkatan

Prioritaskan peningkatan yang Anda identifikasi berdasarkan dampak paling besar yang diantisipasi dengan biaya paling rendah serta risiko yang dapat diterima.

Putuskan peningkatan-peningkatan mana yang harus difokuskan di awal, dan sertakan peningkatan-peningkatan tersebut dalam perencanaan sumber daya dan roadmap pengembangan Anda.

Dengan menerapkan langkah ini ke Contoh skenario, artinya Anda memprioritaskan target peningkatan sebagai berikut:

Prioritas	Peningkatan	Potensi	Biaya	Risiko
1	Implementasikan mekanisme kompresi yang lebih efektif	Tinggi	Rendah	Rendah
2	Implementasikan penskalaan prediktif	Sedang	Rendah	Rendah

Potensi yang tinggi dan biaya serta risiko yang rendah pada pembaruan kompresi file menjadikannya target yang bernilai tinggi untuk perusahaan Anda dan juga menjadikannya prioritas dibandingkan implementasi penskalaan prediktif. Anda menentukan bahwa mengimplementasikan penskalaan prediktif dengan potensi dampak yang sedang dan biaya serta risiko yang rendah harus menjadi peningkatan prioritas setelah kompresi file selesai.

Anda menugaskan seorang anggota tim untuk mengimplementasikan kompresi file yang telah ditingkatkan dan menambahkan penskalaan prediktif ke backlog Anda.

Uji dan validasikan peningkatan

Lakukan pengujian kecil dengan investasi yang minim untuk mengurangi risiko upaya skala besar.

Implementasikan satu salinan beban kerja yang representatif dalam lingkungan pengujian Anda untuk membatasi biaya dan risiko dalam melakukan pengujian dan validasi. Lakukan rangkaian transaksi uji yang ditetapkan sebelumnya, ukur sumber daya yang disediakan, dan tentukan sumber daya yang digunakan untuk setiap unit kerja untuk membuat garis acuan pengujian.

Implementasikan peningkatan target Anda di lingkungan pengujian dan ulangi pengujian tersebut dengan menggunakan metodologi yang sama dengan kondisi yang sama. Dan kemudian ukur sumber daya yang disediakan serta sumber daya yang digunakan untuk setiap unit kerja dengan peningkatan yang telah diterapkan.

Hitung perubahan persentase dari garis acuan sumber daya yang disediakan untuk setiap unit kerja, dan tentukan penurunan kuantitatif yang diharapkan pada sumber daya yang disediakan di lingkungan produksi Anda. Bandingkan nilai-nilai ini dengan nilai-nilai yang diperkirakan. Tentukan apakah hasilnya adalah peningkatan dengan level yang dapat diterima. Evaluasi apakah setiap kompromi pada sumber daya tambahan yang dipakai menjadikan manfaat bersih dari peningkatan tersebut tidak dapat diterima.

Tentukan apakah peningkatan tersebut berhasil dan apakah sumber daya harus diinvestasikan dalam implementasi perubahan yang dilakukan di lingkungan produksi. Jika setelah dievaluasi perubahan dianggap tidak berhasil, maka arahkan sumber daya Anda untuk menguji dan memvalidasi target berikutnya dan lanjutkan siklus peningkatan Anda.

% Penurunan sumber daya yang disediakan per unit kerja	Penurunan kuantitatif sumber daya yang disediakan	Tindakan
Memenuhi ekspektasi	Memenuhi ekspektasi	Lanjutkan peningkatan
Tidak memenuhi ekspektasi	Memenuhi ekspektasi	Lanjutkan peningkatan
Memenuhi ekspektasi	Tidak memenuhi ekspektasi	Cari peningkatan alternatif
Tidak memenuhi ekspektasi	Tidak memenuhi ekspektasi	Cari peningkatan alternatif

Menerapkan langkah ini ke Contoh skenario, Anda melakukan tes untuk memvalidasi kesuksesan.

Setelah Anda melakukan pengujian pada algoritme kompresi yang ditingkatkan, penurunan persentase pada sumber daya yang disediakan untuk setiap unit kerja (penyimpanan yang diperlukan untuk citra asli dan citra modifikasi) akan memenuhi ekspektasi dengan rata-rata penurunan 30% pada penyimpanan yang disediakan dan penambahan beban komputasi dalam jumlah sangat kecil.

Anda menentukan bahwa sumber daya komputasi tambahan yang diperlukan untuk menerapkan algoritme kompresi yang ditingkatkan ke file yang ada di lingkungan produksi tidaklah signifikan dibandingkan dengan penurunan penyimpanan yang dicapai. Anda mengonfirmasi bahwa keberhasilan penurunan kuantitatif pada sumber daya yang diperlukan (TB penyimpanan), dan peningkatan tersebut disetujui untuk deployment produksi.

Terapkan perubahan ke produksi

Implementasikan peningkatan yang telah diuji, divalidasi, dan disetujui ke produksi. Implementasikan menggunakan deployment terbatas, konfirmasikan fungsionalitas beban kerja Anda, uji penurunan riil pada sumber daya yang disediakan dan sumber daya yang digunakan per unit kerja di dalam deployment terbatas, dan periksa apakah ada konsekuensi yang tidak diinginkan dari perubahan tersebut. Lanjutkan deployment penuh setelah pengujian berhasil.

Batalkan perubahan jika pengujian gagal atau Anda mengalami konsekuensi perubahan yang tidak diinginkan pada level yang tidak dapat diterima.

Menerapkan langkah ini ke Contoh skenario, artinya Anda melakukan langkah berikut.

Anda mengimplementasikan perubahan di lingkungan produksi dengan menggunakan deployment terbatas melalui metodologi deployment blue/green. Pengujian fungsionalitas pada instans yang baru di-deploy berhasil. Anda melihat penurunan rata-rata sebesar 26% pada penyimpanan yang disediakan untuk file citra asli dan citra manipulasi. Anda tidak melihat bukti adanya kenaikan beban komputasi ketika mengompresi file-file baru.

Anda melihat penurunan yang tidak terduga pada waktu proses untuk mengompresi file citra, dan Anda mengaitkan hal ini dengan kode yang sangat dioptimalkan untuk algoritme kompresi baru.

Anda melanjutkan deployment versi baru secara penuh.

Ukur hasil dan replikasi keberhasilan

Ukur hasil dan replikasi keberhasilan dengan cara-cara berikut ini:

- Ukur peningkatan awal pada sumber daya yang disediakan untuk setiap unit kerja dan penurunan kuantitatif pada sumber daya yang disediakan.
- Bandingkan estimasi awal dan hasil pengujian dengan pengukuran produksi Anda. Identifikasi faktor-faktor yang mungkin menjadi penyebab terjadinya selisih tersebut, dan perbarui metodologi estimasi dan pengujian Anda, jika diperlukan.
- Tentukan keberhasilan, serta tingkat keberhasilan, dan bagikan hasilnya kepada para pemangku kepentingan.
- Jika Anda harus membatalkan perubahan dikarenakan pengujian yang gagal atau konsekuensi negatif yang tidak diinginkan dari perubahan tersebut, maka Anda harus mengidentifikasi faktor-faktor penyebabnya. Lakukan pengulangan apabila memungkinkan, atau lakukan evaluasi terhadap pendekatan baru untuk mencapai tujuan-tujuan perubahan.
- Ambil pelajaran yang Anda dapatkan, buat standar, dan terapkan peningkatan yang berhasil ke sistem-sistem lain yang bisa menerima manfaat yang serupa. Rekam dan bagikan metodologi Anda, artefak terkait, dan manfaat bersih kepada seluruh tim dan organisasi sehingga orang lain dapat mengadopsi standar Anda dan mengulang keberhasilan Anda.
- Pantau sumber daya yang disediakan untuk setiap unit kerja dan lacak perubahan serta total dampak dari waktu ke waktu. Perubahan-perubahan pada beban kerja Anda, atau cara pelanggan Anda memakai beban kerja Anda, dapat memengaruhi efektivitas peningkatan Anda. Evaluasi ulang peluang-peluang peningkatan jika Anda melihat terjadi penurunan jangka pendek yang besar pada efektivitas peningkatan Anda atau penurunan efektivitas yang terakumulasi dari waktu ke waktu.
- Hitung manfaat bersih dari peningkatan Anda dari waktu ke waktu (termasuk manfaat yang diterima oleh tim lain yang menerapkan peningkatan Anda, jika ada) untuk menunjukkan laba atas investasi dari aktivitas-aktivitas peningkatan Anda.

Menerapkan langkah ini ke Contoh skenario, artinya Anda mengukur hasil berikut.

Beban kerja Anda menunjukkan adanya peningkatan awal berupa penurunan kebutuhan penyimpanan sebesar 23% setelah melakukan deployment dan menerapkan algoritme kompresi baru ke file-file citra yang ada.

Nilai yang diukur sebagian besar sesuai dengan perkiraan awal (25%), dan selisih signifikan yang dibandingkan dengan pengujian (30%) ditentukan menjadi hasil file citra yang digunakan dalam pengujian tidak mewakili file citra yang ada dalam lingkungan produksi. Anda memodifikasi set citra pengujian agar lebih mencerminkan citra dalam produksi.

Peningkatan dianggap sepenuhnya berhasil. Total penurunan penyimpanan yang disediakan 2% lebih sedikit daripada yang diperkirakan yakni 25%, tetapi 23% tetaplah peningkatan besar dalam hal dampak keberlanjutan, dan disertai dengan penghematan biaya yang setara.

Satu-satunya konsekuensi yang tidak diinginkan dari perubahan ini adalah penurunan waktu proses yang bermanfaat untuk melakukan kompresi dan penurunan setara pada vCPU yang dipakai. Semua peningkatan ini dianggap sebagai hasil dari kode yang sangat dioptimalkan.

Anda membuat sebuah proyek sumber terbuka internal di mana Anda membagikan kode, artefak terkait, panduan cara mengimplementasikan perubahan, dan hasil-hasil dari implementasi Anda. Proyek sumber terbuka internal ini memudahkan tim Anda untuk mengadopsi kode tersebut untuk semua kasus penggunaan penyimpanan file tetap mereka. Tim Anda mengadopsi peningkatan ini sebagai standar. Manfaat sekunder dari proyek sumber terbuka internal ini adalah setiap orang yang mengadopsi solusi tersebut mendapatkan manfaat peningkatan yang dilakukan pada solusi, dan siapa pun dapat memberikan kontribusi peningkatan untuk proyek ini.

Anda menerbitkan keberhasilan Anda dan membagikan proyek sumber terbuka Anda kepada seluruh organisasi Anda. Setiap tim yang mengadopsi solusi ini mereplikasi manfaatnya dengan menanamkan investasi minimum dan menjadi manfaat tambahan pada manfaat bersih yang diterima dari investasi Anda. Anda menerbitkan data ini sebagai kisah keberhasilan yang berkelanjutan.

Anda melanjutkan pemantauan yang dilakukan terhadap dampak peningkatan dari waktu ke waktu dan akan membuat perubahan pada proyek sumber terbuka internal Anda, jika diperlukan.

Keberlanjutan sebagai persyaratan nonfungsional

Penambahan keberlanjutan ke daftar persyaratan bisnis dapat menghasilkan solusi-solusi yang lebih hemat biaya. Berfokus untuk memperoleh lebih banyak nilai dari sumber daya yang Anda gunakan dan menggunakan sumber daya yang lebih sedikit secara langsung akan menghasilkan penghematan biaya di AWS karena Anda hanya membayar sesuai yang Anda gunakan.

Memenuhi target keberlanjutan mungkin tidak memerlukan kompromi setara di satu atau beberapa metrik tradisional seperti waktu aktif, ketersediaan, atau waktu respons. Anda dapat mencapai hasil yang besar dalam hal keberlanjutan tanpa ada dampak yang terukur pada tingkat layanan. Apabila kompromi kecil diperlukan, peningkatan keberlanjutan yang didapatkan dari kompromi tersebut dapat mengungguli perubahan kualitas layanan.

Dorong anggota tim Anda untuk terus bereksperimen dengan peningkatan keberlanjutan saat mereka mengembangkan persyaratan fungsional. Tim juga harus menyematkan metrik-metrik proksi saat menetapkan tujuan untuk memastikan bahwa mereka mengevaluasi intensitas sumber daya saat mengembangkan beban kerja.

Berikut ini adalah contoh kompromi yang dapat mengurangi sumber daya cloud yang Anda pakai:

Sesuaikan kualitas hasilnya: Anda dapat mengorbankan Kualitas Hasil (QoR) demi pengurangan intensitas beban kerja dengan penghitungan perkiraan. Praktik komputasi perkiraan mencari peluang-peluang pemanfaatan celah antara apa yang dibutuhkan pelanggan dan apa yang sebenarnya Anda hasilkan. Sebagai contoh, jika Anda menempatkan data di sebuah struktur data set, Anda dapat menggunakan operator ORDER BY di SQL untuk menghilangkan pemrosesan yang tidak perlu, sehingga dapat menghemat sumber daya sambil tetap menyediakan jawaban yang dapat diterima.

Sesuaikan waktu respons: Jawaban dengan waktu respons yang lebih lambat dapat mengurangi karbon dengan meminimalkan biaya tambahan bersama. Memproses tugas-tugas khusus dan sementara dapat menimbulkan biaya overhead perusahaan rintisan. Kelompokkan dan proses tugas-tugas dalam batch, bukan membayar biaya tambahan setiap kali ada tugas muncul. Pemrosesan batch mengorbankan waktu respons yang lebih cepat demi pengurangan biaya overhead bersama untuk membuat instans, mengunduh kode sumber, dan menjalankan proses.

Sesuaikan ketersediaan: Dengan AWS, Anda dapat menambahkan redundansi dan memenuhi target ketersediaan tinggi hanya dengan beberapa kali klik. Anda dapat meningkatkan redundansi dengan menggunakan teknik-teknik seperti stabilitas statis dengan menyediakan sumber daya tidak aktif

yang selalu menyebabkan pemanfaatan yang lebih rendah. Evaluasi kebutuhan-kebutuhan bisnis saat menetapkan target. Kompromi yang relatif kecil dalam hal ketersediaan dapat mengakibatkan peningkatan pemanfaatan yang jauh lebih besar. Misalnya, pola arsitektur stabilitas statis melibatkan pengadaan kapasitas failover tidak aktif agar dapat langsung mengambil beban setelah terjadi kesalahan komponen. Pelonggaran persyaratan ketersediaan dapat menghilangkan kebutuhan kapasitas online tidak aktif dengan menyediakan waktu otomatisasi untuk melakukan deployment sumber daya pengganti. Penambahan kapasitas failover sesuai permintaan dapat mendorong pemanfaatan yang lebih tinggi secara keseluruhan tanpa mengganggu bisnis selama operasi normal dan memiliki manfaat sekunder berupa penurunan biaya.

Praktik terbaik untuk keberlanjutan di cloud

Optimalkan penempatan beban kerja, dan optimalkan arsitektur untuk permintaan, perangkat lunak, data, perangkat keras, dan proses untuk meningkatkan efisiensi energi. Masing-masing area ini merepresentasikan peluang untuk menerapkan praktik terbaik guna mengurangi dampak beban kerja cloud Anda terhadap keberlanjutan dengan memaksimalkan pemanfaatan, dan meminimalkan limbah serta total sumber daya yang di-deploy dan diberdayakan untuk mendukung beban kerja Anda.

Topik

Pemilihan wilayah

Pilihan Wilayah untuk beban kerja Anda sangat memengaruhi KPI-nya, termasuk kinerja, biaya, dan jejak karbon. Untuk meningkatkan semua KPI ini secara efektif, sebaiknya pilih Wilayah untuk beban kerja Anda berdasarkan persyaratan bisnis dan tujuan keberlanjutan Anda.

Praktik terbaik

SUS01-BP01 Memilih Wilayah berdasarkan persyaratan bisnis dan tujuan-tujuan keberlanjutan

Pilihlah suatu Wilayah untuk beban kerja Anda berdasarkan persyaratan bisnis dan tujuan keberlanjutan Anda untuk mengoptimalkan KPI, termasuk kinerja, biaya, dan jejak karbon.

Anti-pola umum:

- Anda memilih Wilayah beban kerja berdasarkan lokasi Anda sendiri.

- Anda menggabungkan semua sumber daya beban kerja ke dalam satu lokasi geografis.

Manfaat menerapkan praktik terbaik ini: Menempatkan beban kerja berada dalam lokasi yang dekat dengan proyek energi terbarukan Amazon atau Wilayah dengan intensitas karbon diterbitkan yang rendah dapat membantu Anda dalam menurunkan jejak karbon dari beban kerja cloud.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

AWS Cloud adalah sebuah jaringan Wilayah dan titik kehadiran (PoP) yang terus-menerus berekspansi, dengan sebuah infrastruktur jaringan global yang menghubungkan semuanya. Pilihan Wilayah untuk beban kerja Anda sangat memengaruhi KPI-nya, termasuk kinerja, biaya, dan jejak karbon. Untuk meningkatkan semua KPI ini secara efektif, Anda harus memilih Wilayah untuk beban kerja Anda berdasarkan persyaratan bisnis dan tujuan-tujuan keberlanjutan Anda.

Langkah-langkah implementasi

- Ikuti langkah-langkah ini untuk mengevaluasi dan merangkum Wilayah potensial untuk beban kerja Anda berdasarkan persyaratan bisnis Anda, termasuk persyaratan kepatuhan, fitur yang tersedia, biaya, dan latensi:
  - Konfirmasikan bahwa Wilayah-wilayah tersebut mematuhi persyaratan peraturan setempat yang berlaku.
  - Gunakan Daftar Layanan Regional AWS untuk memeriksa apakah Wilayah-Wilayah tersebut memiliki layanan dan fitur yang Anda perlukan untuk menjalankan beban kerja Anda.
  - Hitung biaya beban kerja di setiap Wilayah dengan menggunakan AWS Pricing Calculator.
  - Uji latensi jaringan antara lokasi pengguna akhir Anda dan masing-masing Wilayah AWS.
- Pilih Wilayah yang dekat dengan proyek-proyek energi terbarukan Amazon dan Wilayah dengan jaringan energi yang memiliki intensitas karbon terpublikasi lebih rendah daripada lokasi (atau Wilayah) lain.
  - Identifikasi pedoman keberlanjutan Anda yang relevan untuk melacak dan membandingkan emisi karbon dari tahun ke tahun berdasarkan Protokol Gas Rumah Kaca (metode berbasis pasar dan berbasis lokasi).
  - Pilih wilayah berdasarkan metode yang Anda gunakan untuk melacak emisi karbon. Untuk detail selengkapnya tentang cara memilih Wilayah berdasarkan pedoman keberlanjutan Anda, lihat Cara memilih Wilayah untuk beban kerja Anda berdasarkan tujuan keberlanjutan.

Sumber daya

Dokumen terkait:

- Memahami perkiraan emisi karbon Anda
- Amazon di Seluruh Dunia
- Metodologi Energi Terbarukan
- Hal-Hal yang Perlu Dipertimbangkan saat Memilih Wilayah untuk Beban Kerja Anda

Video terkait:

- AWS re:Invent 2023 - Inovasi keberlanjutan dalam Infrastruktur Global AWS
- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- AWS re:Invent 2022 - Menghadirkan arsitektur berkelanjutan dan berkinerja tinggi
- AWS re:Invent 2022 - Merancang arsitektur secara berkelanjutan dan mengurangi jejak karbon AWS Anda
- AWS re:Invent 2022 - Keberlanjutan dalam infrastruktur global AWS

Penyelarasan dengan permintaan

Cara pengguna dan aplikasi menggunakan beban kerja Anda dan sumber daya lainnya dapat membantu Anda mengidentifikasi peningkatan untuk memenuhi tujuan keberlanjutan. Skalakan infrastruktur agar dapat terus sesuai dengan permintaan dan pastikan bahwa Anda hanya menggunakan sumber daya minimum yang diperlukan untuk mendukung para pengguna Anda. Selaraskan tingkat layanan dengan kebutuhan para pelanggan. Posisikan sumber daya guna membatasi jaringan yang diperlukan para pengguna dan aplikasi untuk memakainya. Hapus aset yang tidak digunakan. Bekali anggota tim Anda dengan perangkat yang mendukung kebutuhan mereka dan meminimalkan dampak terhadap keberlanjutan.

Praktik terbaik

- SUS02-BP01 Menskalakan infrastruktur beban kerja secara dinamis
- SUS02-BP02 Menyelaraskan SLA dengan tujuan keberlanjutan
- SUS02-BP03 Menghentikan pembuatan dan pemeliharaan aset yang tak terpakai
- SUS02-BP04 Mengoptimalkan penempatan geografis beban kerja berdasarkan persyaratan jaringannya

- SUS02-BP05 Mengoptimalkan sumber daya anggota tim untuk aktivitas yang dijalankan
- SUS02-BP06 Mengimplementasikan buffering atau throttling untuk meratakan kurva permintaan

SUS02-BP01 Menskalakan infrastruktur beban kerja secara dinamis

Gunakan elastisitas cloud dan skalakan infrastruktur Anda secara dinamis untuk menyesuaikan pasokan sumber daya cloud dengan permintaan dan menghindari terjadinya kelebihan penyediaan kapasitas di beban kerja Anda.

Anti-pola umum:

- Anda tidak menskalakan infrastruktur Anda dengan beban pengguna.
- Anda menskalakan secara manual infrastruktur Anda sepanjang waktu.
- Anda membiarkan peningkatan kapasitas setelah terjadi peristiwa penskalaan, bukannya menurunkan kembali skala.

Manfaat menerapkan praktik terbaik ini: Mengkonfigurasi dan menguji elastisitas beban kerja akan Anda membantu dalam mencocokkan pasokan sumber daya cloud secara efisien dengan permintaan dan menghindari terjadinya kelebihan penyediaan kapasitas. Anda dapat memanfaatkan elastisitas di cloud untuk menskalakan kapasitas secara otomatis selama dan setelah terjadi lonjakan permintaan. Hal ini bertujuan untuk memastikan bahwa Anda hanya menggunakan jumlah sumber daya yang benar-benar diperlukan untuk memenuhi persyaratan-persyaratan bisnis Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Cloud menyediakan fleksibilitas untuk memperluas atau mengurangi sumber daya Anda secara dinamis melalui beragam mekanisme untuk memenuhi perubahan-perubahan sesuai dengan permintaan. Menyesuaikan pasokan dengan permintaan secara optimal akan memberikan dampak lingkungan terendah untuk sebuah beban kerja.

Permintaan dapat bersifat tetap atau bervariasi, sehingga akan memerlukan metrik-metrik dan otomatisasi untuk memastikan bahwa manajemen permintaan tersebut tidak akan menyulitkan. Aplikasi dapat diskalakan secara vertikal (naik atau turun) dengan mengubah ukuran instans, secara horizontal (ke dalam atau ke luar) dengan mengubah jumlah instans, atau melakukan kombinasi keduanya.

Anda dapat menggunakan sejumlah pendekatan yang berbeda untuk menyesuaikan pasokan sumber daya dengan permintaan.

- Pendekatan pelacakan target: Pantau metrik penskalaan Anda dan tingkatkan atau turunkan kapasitas secara otomatis sesuai kebutuhan.
- Penskalaan prediktif: Lakukan pengurangan skala (scale in) dalam mengantisipasi tren harian dan mingguan.
- Pendekatan berbasis jadwal: Tetapkan jadwal penskalaan Anda sendiri sesuai dengan perubahan beban yang dapat diprediksi.
- Penskalaan layanan: Pilih layanan (seperti nirserver) yang secara native menskalakan berdasarkan desain atau menyediakan penskalaan otomatis sebagai fitur.

Identifikasi periode penggunaan rendah atau nol dan skalakan sumber daya untuk menghapus kapasitas berlebih dan meningkatkan efisiensi.

Langkah-langkah implementasi

- Elastisitas menyesuaikan pasokan sumber daya yang Anda miliki dengan permintaan untuk sumber daya tersebut. Instans, kontainer, dan fungsi menyediakan mekanisme-mekanisme untuk elastisitas, baik dalam kombinasi dengan penskalaan otomatis maupun sebagai fitur layanan. AWS menyediakan serangkaian mekanisme penskalaan otomatis untuk memastikan bahwa beban kerja tersebut dapat diturunkan skalanya dengan cepat dan mudah selama periode beban pengguna yang rendah. Berikut ini adalah beberapa contoh mekanisme penskalaan otomatis:

Mekanisme penskalaan otomatis	Harus digunakan di mana
Amazon EC2 Auto Scaling	Gunakan untuk memastikan Anda memiliki jumlah instans Amazon EC2 yang tepat untuk menangani beban pengguna aplikasi Anda.
Penskalaan Otomatis Aplikasi	Gunakan untuk secara otomatis menskalak an sumber daya untuk masing-masing layanan AWS di luar Amazon EC2, seperti fungsi Lambda atau layanan Amazon Elastic Container Service (Amazon ECS).

Mekanisme penskalaan otomatis	Harus digunakan di mana
Penskala Otomatis Klaster Kubernetes	Gunakan untuk secara otomatis menskalakan klaster Kubernetes di AWS.

- Penskalaan sering kali dibahas terkait dengan layanan-layanan komputasi seperti instans Amazon EC2 atau fungsi AWS Lambda. Pertimbangkan konfigurasi layanan non-komputasi seperti unit kapasitas baca dan tulis Amazon DynamoDB atau serpihan (shard) Amazon Kinesis Data Streams agar sesuai dengan permintaan.
- Pastikan bahwa metrik-metrik untuk melakukan peningkatan atau penurunan skala telah divalidasi terhadap jenis beban kerja yang di-deploy. Jika Anda men-deploy sebuah aplikasi transkode video, 100% pemanfaatan CPU adalah hal normal dan tidak boleh menjadi metrik primer Anda. Anda dapat menggunakan metrik kustom (seperti pemanfaatan memori) untuk kebijakan penskalaan Anda jika diperlukan. Untuk memilih metrik yang tepat, pertimbangkan panduan berikut untuk Amazon EC2:
  - Metrik tersebut harus merupakan metrik pemanfaatan yang valid dan mendeskripsikan tingkat kesibukan suatu instans.
  - Nilai metrik harus meningkatkan atau menurunkan secara proporsional jumlah instance dalam grup Auto Scaling.
- Gunakan penskalaan dinamis alih-alih penskalaan manual untuk grup Auto Scaling Anda. Kami juga menyarankan agar Anda menggunakan kebijakan penskalaan pelacakan target dalam penskalaan dinamis Anda.
- Pastikan deployment beban kerja dapat menangani peristiwa penambahan skala dan pengurangan skala. Buatlah skenario pengujian untuk peristiwa-peristiwa penambahan skala guna memastikan bahwa beban kerja berperilaku sesuai harapan dan tidak memengaruhi pengalaman pengguna (seperti kehilangan sesi lekat (sticky session)). Anda dapat menggunakan Riwayat aktivitas untuk melakukan verifikasi terhadap aktivitas penskalaan untuk sebuah grup Auto Scaling.
- Lakukan evaluasi terhadap beban kerja Anda untuk memeriksa pola-pola terprediksi dan secara proaktif skalakan saat Anda mengantisipasi perubahan permintaan yang terencana dan terprediksi. Dengan penskalaan prediktif, Anda dapat menghilangkan kebutuhan untuk menyediakan kapasitas secara berlebih. Untuk detail selengkapnya, silakan lihat Penskalaan Prediktif dengan Amazon EC2 Auto Scaling.

Sumber daya

Dokumen terkait:

- Memulai dengan Amazon EC2 Auto Scaling
- Penskalaan Prediktif untuk EC2, Didukung oleh Machine Learning
- Menganalisis perilaku pengguna dengan menggunakan Amazon OpenSearch Service, Amazon Data Firehose dan Kibana
- Apa itu Amazon CloudWatch?
- Memantau muatan DB dengan Wawasan Performa di Amazon RDS
- Memperkenalkan Dukungan Native untuk Penskalaan Prediktif dengan Amazon EC2 Auto Scaling
- Memperkenalkan Karpenter - Penskala Otomatis Klaster Kubernetes Sumber Terbuka dan Performa Tinggi
- Memahami Lebih Dalam Penskalaan Otomatis Klaster Amazon ECS

Video terkait:

- AWS re:Invent 2023 - Menskalakan di AWS untuk 10 juta pengguna pertama
- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- AWS re:Invent 2022 - Membangun lingkungan komputasi yang hemat biaya, energi, dan sumber daya
- AWS re:Invent 2022 - Menskalakan kontainer dari satu pengguna hingga jutaan pengguna
- AWS re:Invent 2023 - Menskalakan inferensi FM ke ratusan model dengan Amazon SageMaker
- AWS re:Invent 2023 - Memanfaatkan kekuatan Karpenter untuk menskalakan, mengoptimalkan & meningkatkan Kubernetes

Contoh terkait:

- Penskalaan otomatis

SUS02-BP02 Menyelaraskan SLA dengan tujuan keberlanjutan

Tinjau dan optimalkan perjanjian tingkat layanan (SLA) beban kerja berdasarkan tujuan keberlanjutan Anda untuk meminimalkan sumber daya yang diperlukan untuk mendukung beban kerja Anda sambil terus memenuhi kebutuhan bisnis.

Anti-pola umum:

- SLA beban kerja tidak diketahui atau ambigu.
- Anda menetapkan SLA hanya demi ketersediaan dan kinerja.
- Anda menggunakan pola desain yang sama (seperti arsitektur Multi-AZ) untuk semua beban kerja Anda.

Manfaat menerapkan praktik terbaik ini: Menyelaraskan SLA dengan tujuan-tujuan keberlanjutan akan mengarahkan Anda pada penggunaan sumber daya yang optimal sekaligus memenuhi kebutuhan bisnis Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

SLA menetapkan tingkat layanan yang diharapkan dari sebuah beban kerja cloud, seperti waktu respons, ketersediaan, dan retensi data. SLA memengaruhi arsitektur, penggunaan sumber daya, dan dampak lingkungan yang ditimbulkan sebuah beban kerja cloud. Secara rutin, tinjau SLA dan buat pilihan kompromi yang secara signifikan akan mengurangi penggunaan sumber daya dengan penurunan tingkat layanan yang dapat diterima.

Langkah-langkah implementasi

- Memahami tujuan keberlanjutan: Identifikasi tujuan-tujuan keberlanjutan yang ditetapkan dalam organisasi Anda, seperti pengurangan karbon atau peningkatan pemanfaatan sumber daya.
- Tinjau SLA: Lakukan evaluasi terhadap SLA Anda untuk menilai apakah mereka mendukung persyaratan-persyaratan bisnis Anda. Jika Anda melampaui SLA, maka lakukan peninjauan lebih lanjut.
- Memahami kompromi: Memahami kompromi yang bisa dilakukan di seluruh kompleksitas beban kerja Anda (seperti pengguna yang menggunakan volume tinggi secara bersamaan), kinerja (seperti latensi), dan dampak keberlanjutan (seperti sumber daya yang diperlukan). Umumnya, ketika dua faktor diprioritaskan, faktor ketiga akan dikorbankan.
- Sesuaikan SLA: Lakukan penyesuaian terhadap SLA Anda dengan mengambil pilihan kompromi yang secara signifikan mengurangi dampak keberlanjutan dengan penurunan yang dapat diterima dalam hal tingkat layanan.
  - Keberlanjutan dan keandalan: Beban kerja dengan ketersediaan yang sangat tinggi cenderung mengkonsumsi lebih banyak sumber daya.
  - Keberlanjutan dan kinerja: Menggunakan lebih banyak sumber daya untuk meningkatkan kinerja cenderung memiliki dampak lingkungan yang lebih tinggi.

  - Keberlanjutan dan keamanan: Beban kerja yang terlalu aman cenderung memiliki dampak lingkungan yang lebih tinggi.
- Tentukan SLA keberlanjutan jika memungkinkan: Sertakan SLA keberlanjutan untuk beban kerja Anda. Misalnya, tentukan tingkat pemanfaatan minimum sebagai SLA keberlanjutan untuk instans komputasi Anda.
- Gunakan pola desain yang efisien: Gunakan pola desain yang mengutamakan fungsi-fungsi bisnis penting seperti layanan mikro di AWS, dan izinkan tingkat layanan (seperti waktu respons atau tujuan waktu pemulihan) yang lebih rendah untuk fungsi-fungsi yang tidak penting.
- Berkomunikasi dan membangun akuntabilitas: Bagikan SLA dengan semua pemangku kepentingan yang relevan, termasuk tim pengembangan Anda dan para pelanggan Anda. Gunakan pelaporan untuk melacak dan memantau SLA. Tetapkan pertanggungjawaban untuk memenuhi target-target keberlanjutan untuk SLA Anda.
- Gunakan insentif dan penghargaan: Gunakan insentif dan penghargaan untuk mencapai atau melampaui SLA yang selaras dengan tujuan-tujuan keberlanjutan.
- Tinjau dan ulangi: Tinjau dan sesuaikan SLA Anda secara teratur untuk memastikannya selaras dengan keberlanjutan dan tujuan-tujuan kinerja yang terus berkembang.

Sumber daya

Dokumen terkait:

- Memahami pola ketahanan dan kompromi untuk merancang secara efisien di cloud
- Pentingnya Perjanjian Tingkat Layanan untuk Penyedia SaaS

Video terkait:

- AWS re:Invent 2023 - Kapasitas, ketersediaan, efisiensi biaya: Pilih ketiganya
- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- AWS re:Invent 2023 - Pola integrasi tingkat lanjut & kompromi untuk sistem yang digabungkan dengan metode penggabungan longgar
- AWS re:Invent 2022 - Menghadirkan arsitektur berkelanjutan dan berkinerja tinggi
- AWS re:Invent 2022 - Membangun lingkungan komputasi yang hemat biaya, energi, dan sumber daya

SUS02-BP03 Menghentikan pembuatan dan pemeliharaan aset yang tak terpakai

Nonaktifkan aset yang tak terpakai di beban kerja Anda untuk mengurangi jumlah sumber daya cloud yang diperlukan untuk mendukung permintaan Anda dan meminimalkan limbah.

Anti-pola umum:

- Anda tidak melakukan analisis terhadap aplikasi Anda untuk mengetahui aset-aset yang redundan atau tidak diperlukan lagi.
- Anda tidak menyingkirkan aset yang redundan atau tidak diperlukan lagi.

Manfaat menerapkan praktik terbaik ini: Menghapus aset yang tidak lagi digunakan akan membebaskan sumber daya dan meningkatkan efisiensi keseluruhan beban kerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Aset yang tak terpakai mengonsumsi sumber daya cloud seperti ruang penyimpanan dan daya komputasi. Dengan mengidentifikasi dan mengeliminasi aset-aset ini, Anda dapat membebaskan berbagai sumber daya ini, sehingga arsitektur cloud akan menjadi lebih efisien. Lakukan analisis terhadap aset-aset aplikasi secara rutin, yakni aset-aset seperti laporan pra-kompilasi, set data, gambar statis, dan pola akses aset untuk mengidentifikasi redundansi, pemanfaatan yang terlalu rendah, dan potensi target penonaktifan. Singkirkan aset-aset redundan tersebut untuk mengurangi limbah sumber daya di beban kerja Anda.

Langkah-langkah implementasi

- Lakukan inventarisasi: Lakukan inventarisasi secara komprehensif untuk mengidentifikasi semua aset yang ada dalam beban kerja Anda.
- Lakukan analisis penggunaan: Gunakan pemantauan terus-menerus untuk mengidentifikasi aset-aset statis yang tidak diperlukan lagi.
- Hapus aset yang tidak digunakan: Buatlah rencana untuk menghapus aset yang tidak lagi diperlukan.
  - Sebelum menyingkirkan aset apa pun, evaluasi terlebih dahulu dampak penyingkirannya di dalam arsitektur.
  - Gabungkan aset tumpang tindih yang dihasilkan untuk menghindari redundansi pemrosesan.

    - Perbarui aplikasi Anda hingga tidak lagi yang membuat dan menyimpan aset -aset yang tidak diperlukan.
- Komunikasikan dengan pihak ketiga: Arahkan pihak ketiga untuk berhenti memproduksi dan menyimpan aset yang dikelola atas nama Anda yang tidak diperlukan lagi. Mintalah untuk menggabungkan aset-aset redundan.
- Gunakan kebijakan siklus hidup: Gunakan kebijakan siklus hidup untuk menghapus aset yang tidak digunakan secara otomatis.
    - Anda dapat menggunakan Siklus Hidup Amazon S3 untuk mengelola objek-objek Anda di sepanjang siklus hidupnya.
    - Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi pembuatan, retensi, dan penghapusan snapshot Amazon EBS dan AMI yang didukung Amazon EBS.
- Tinjau dan optimalkan: Lakukan peninjauan secara teratur terhadap beban kerja Anda untuk mengidentifikasi dan menghapus aset yang tak terpakai.

Sumber daya

Dokumen terkait:

- Mengoptimalkan Infrastruktur AWS Anda untuk Keberlanjutan, Bagian II: Penyimpanan
- Bagaimana cara menghentikan sumber daya aktif yang tidak lagi saya perlukan di Akun AWS?

Video terkait:

- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- AWS re:Invent 2022 - Melestarikan dan memaksimalkan nilai aset media digital menggunakan Amazon S3
- AWS re:Invent 2023 - Mengoptimalkan biaya di lingkungan multi-akun

SUS02-BP04 Mengoptimalkan penempatan geografis beban kerja berdasarkan persyaratan jaringannya

Pilih layanan dan lokasi cloud untuk beban kerja Anda yang mengurangi jarak yang harus ditempuh lalu lintas jaringan dan menurunkan total sumber daya jaringan yang diperlukan untuk mendukung beban kerja Anda.

Anti-pola umum:

- Anda memilih Wilayah beban kerja berdasarkan lokasi Anda sendiri.
- Anda menggabungkan semua sumber daya beban kerja ke dalam satu lokasi geografis.
- Semua lalu lintas mengalir melalui pusat data Anda.

Manfaat menerapkan praktik terbaik ini: Menempatkan beban kerja dekat dengan penggunanya akan menghasilkan latensi terendah sambil mengurangi pergerakan data di seluruh jaringan dan mengurangi dampak lingkungan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Infrastruktur AWS Cloud dibangun di seputar opsi lokasi seperti Wilayah, Zona Ketersediaan, grup penempatan, dan lokasi edge seperti AWS Outposts dan Zona Lokal AWS. Opsi-opsi lokasi ini bertanggung jawab untuk memelihara konektivitas yang ada di antara komponen-komponen aplikasi, layanan cloud, jaringan edge, dan pusat data on-premise.

Lakukan analisis terhadap pola akses jaringan yang ada di beban kerja Anda untuk mengidentifikasi cara menggunakan opsi lokasi cloud ini dan mengurangi jarak yang harus ditempuh lalu lintas jaringan.

Langkah-langkah implementasi

- Lakukan analisis terhadap pola akses jaringan di beban kerja Anda untuk mengidentifikasi cara pengguna menggunakan aplikasi Anda.
  - Gunakan alat pemantauan, seperti Amazon CloudWatch dan AWS CloudTrail, untuk mengumpulkan data tentang aktivitas jaringan.
  - Analisis data untuk mengidentifikasi pola akses jaringan.
- Pilihlah Wilayah untuk deployment beban kerja Anda berdasarkan elemen-elemen utama berikut ini:
  - Tujuan Keberlanjutan Anda: seperti yang dijelaskan dalam Pemilihan Wilayah.
  - Dimana lokasi data Anda: Untuk aplikasi-aplikasi dengan banyak data (seperti big data dan machine learning), kode aplikasi harus dijalankan sedekat mungkin dengan data.
  - Dimana lokasi pengguna Anda: Untuk aplikasi-aplikasi yang ditampilkan kepada pengguna, pilihlah sebuah Wilayah (Wilayah-wilayah) yang dekat dengan para pengguna beban kerja Anda.

  - Kendala lainnya: Pertimbangkan kendala-kendala seperti biaya dan kepatuhan sebagaimana yang dijelaskan dalam Hal-Hal yang Perlu Dipertimbangkan saat Memilih Wilayah untuk Beban Kerja Anda.
- Gunakan caching lokal atau Solusi Penerapan Cache AWS untuk aset-aset yang sering digunakan untuk meningkatkan performa, mengurangi perpindahan data, dan mengurangi dampak pada lingkungan.

Layanan	Kapan harus digunakan
Amazon CloudFront	Gunakan untuk melakukan caching terhadap konten-konten statis seperti gambar, skrip, dan video, serta konten-konten dinamis seperti respons API atau aplikasi web.
Amazon ElastiCache	Gunakan untuk meng-cache konten bagi aplikasi web.
DynamoDB Accelerator	Gunakan untuk menambahkan percepatan dalam memori ke tabel DynamoDB Anda.

- Gunakan layanan-layanan yang dapat membantu Anda menjalankan kode lebih dekat dengan para pengguna beban kerja Anda:

Layanan	Kapan harus digunakan
Lambda@Edge	Gunakan untuk operasi-operasi yang memiliki banyak komputasi yang dimulai saat objek tidak ada dalam cache.
Fungsi Amazon CloudFront	Gunakan untuk kasus-kasus penggunaa n sederhana, misalnya permintaan HTTP atau manipulasi respons yang dapat dimulai oleh fungsi-fungsi yang memiliki masa pakai singkat.

Layanan	Kapan harus digunakan
AWS IoT Greengrass	Gunakan untuk menjalankan komputasi lokal, olahpesan, dan caching data untuk perangkat yang terhubung.

- Gunakan pooling koneksi untuk mengizinkan penggunaan ulang koneksi dan mengurangi sumber daya yang diperlukan.
- Gunakan penyimpanan data terdistribusi yang tidak mengandalkan koneksi persisten dan pembaruan-pembaruan selaras untuk mendapatkan konsistensi guna melayani populasi wilayah.
- Ganti kapasitas jaringan statis yang disediakan di awal dengan kapasitas dinamis bersama, dan bagikan dampak keberlanjutan kapasitas jaringan kepada pelanggan lain.

Sumber daya

Dokumen terkait:

- Mengoptimalkan Infrastruktur AWS Anda untuk Keberlanjutan, Bagian III: Jaringan
- Dokumentasi Amazon ElastiCache
- Apa yang dimaksud dengan Amazon CloudFront?
- Fitur Utama Amazon CloudFront
- Infrastruktur Global AWS
- Zona Lokal AWS dan AWS Outposts, memilih teknologi yang tepat untuk beban kerja edge Anda
- Grup penempatan
- Zona Lokal AWS
- AWS Outposts

Video terkait:

- Menjelaskan transfer data di AWS
- Menskalakan kinerja jaringan pada instans Amazon EC2 generasi berikutnya
- Video Penjelas Zona Lokal AWS
- AWS Outposts: Ikhtisar dan Cara Kerjanya
- AWS re:Invent 2023 - Strategi migrasi untuk beban kerja edge dan on-premise

- AWS re:Invent 2021 - AWS Outposts: Membawa pengalaman AWS ke on-premise
- AWS re:Invent 2020 - AWS Wavelength: Menjalankan aplikasi dengan latensi sangat rendah di edge 5G
- AWS re:Invent 2022 - Zona Lokal AWS: Membangun aplikasi untuk edge terdistribusi
- AWS re:Invent 2021 - Membangun situs web latensi rendah dengan Amazon CloudFront
- AWS re:Invent 2022 - Meningkatkan performa dan ketersediaan dengan AWS Global Accelerator
- AWS re:Invent 2022 - Membangun jaringan area luas global dengan menggunakan AWS
- AWS re:Invent 2020: Manajemen lalu lintas global dengan Amazon Route 53

Contoh terkait:

- Lokakarya Jaringan AWS
- Merancang arsitektur untuk keberlanjutan - Meminimalkan pergerakan data lintas jaringan

SUS02-BP05 Mengoptimalkan sumber daya anggota tim untuk aktivitas yang dijalankan

Optimalkan sumber daya yang disediakan bagi anggota tim untuk meminimalkan dampak keberlanjutan sambil mendukung kebutuhan mereka.

Anti-pola umum:

- Anda mengabaikan dampak yang ditimbulkan oleh perangkat yang digunakan oleh anggota tim Anda terhadap efisiensi aplikasi cloud secara keseluruhan.
- Anda mengelola dan memperbarui sumber daya yang digunakan oleh anggota tim secara manual.

Manfaat menerapkan praktik terbaik ini: Mengoptimalkan sumber daya anggota tim akan meningkatkan efisiensi keseluruhan aplikasi yang berkemampuan cloud.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Pahami sumber daya yang digunakan anggota tim Anda untuk mengonsumsi layanan Anda, ekspektasi siklus hidup mereka, dan dampak finansial serta dampak pada keberlanjutan.

Implementasikan strategi untuk melakukan optimalisasi terhadap berbagai sumber daya ini. Sebagai contoh, lakukan operasi yang kompleks, seperti rendering dan kompilasi, pada infrastruktur yang dapat diskalakan dan sangat banyak digunakan, bukan pada sistem pengguna tunggal dan berdaya tinggi namun jarang digunakan.

Langkah-langkah implementasi

- Gunakan workstation hemat energi: Sediakan workstation dan periferal hemat energi kepada anggota tim. Gunakan fitur manajemen daya yang efisien (seperti mode daya rendah) di perangkat-perangkat tersebut untuk mengurangi penggunaan energinya
- gunakan virtualisasi: Gunakan streaming aplikasi dan desktop virtual untuk membatasi persyaratan perangkat dan pemutakhiran.
- Dorong kolaborasi jarak jauh: Dorong anggota tim untuk menggunakan alat kolaborasi jarak jauh seperti Amazon Chime atau AWS Wickr untuk mengurangi kebutuhan akan perjalanan dan emisi karbon terkait.
- Gunakan perangkat lunak hemat energi: Berikan anggota tim perangkat lunak yang hemat energi dengan menghapus atau mematikan fitur dan proses yang tidak perlu.
- Lakukan pengelolaan siklus hidup: Evaluasi dampak proses dan sistem atas siklus hidup perangkat, dan pilih solusi yang meminimalkan persyaratan untuk penggantian perangkat sekaligus memenuhi persyaratan bisnis. Lakukan pemeliharaan dan pembaruan secara rutin terhadap stasiun kerja atau perangkat lunak untuk menjaga dan memperbaiki efisiensi.
- Pengelolaan perangkat jarak jauh: Implementasikan manajemen jarak jauh untuk perangkat guna mengurangi perjalanan bisnis yang diperlukan.
  - AWS Systems Manager Fleet Manager adalah sebuah pengalaman antarmuka pengguna (UI) terpadu yang akan membantu Anda mengelola simpul Anda secara jarak jauh yang beroperasi di AWS atau on-premise.

Sumber daya

Dokumen terkait:

- Apa yang dimaksud dengan Amazon WorkSpaces?
- Pengoptimal Biaya untuk Amazon WorkSpaces
- Dokumentasi Amazon AppStream 2.0
- NICE DCV

Video terkait:

- Mengelola biaya untuk Amazon WorkSpaces di AWS

SUS02-BP06 Mengimplementasikan buffering atau throttling untuk meratakan kurva permintaan

Buffering dan throttling meratakan kurva permintaan dan mengurangi kapasitas tersedia yang diperlukan untuk beban kerja Anda.

Anti-pola umum:

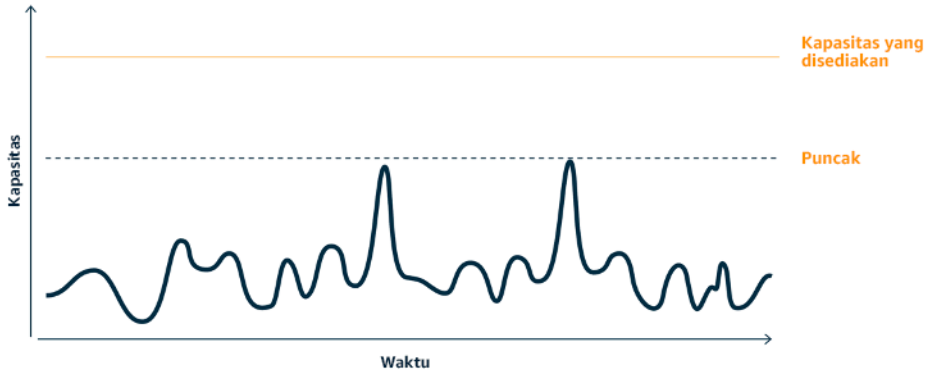
- Anda memproses permintaan klien dengan segera walaupun tidak diperlukan.
- Anda tidak menganalisis persyaratan-persyaratan untuk permintaan klien.

Manfaat menerapkan praktik terbaik ini: Dengan meratakan kurva permintaan Anda akan mengurangi kapasitas yang disediakan untuk beban kerja. Mengurangi kapasitas tersedia artinya konsumsi energi berkurang dan dampak pada lingkungan juga akan berkurang.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

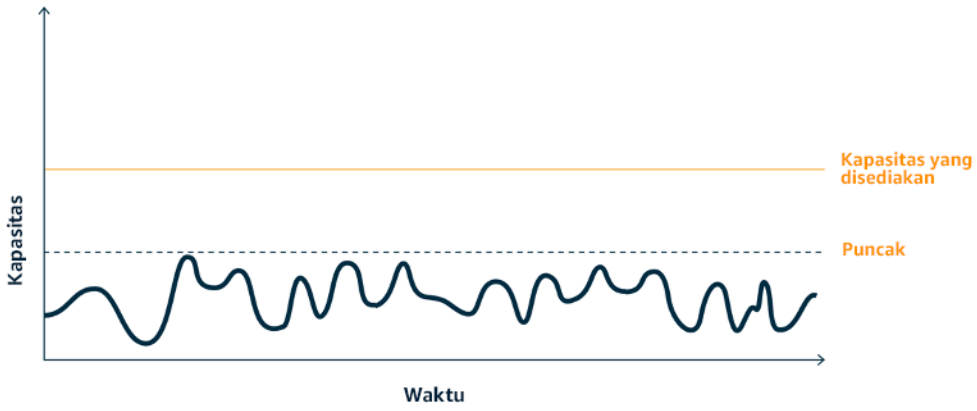
Panduan implementasi

Meratakan kurva permintaan beban kerja dapat membantu Anda mengurangi kapasitas tersedia untuk beban kerja dan mengurangi dampaknya terhadap lingkungan. Asumsikan sebuah beban kerja dengan kurva permintaan seperti yang ditunjukkan pada gambar di bawah ini. Beban kerja ini memiliki dua puncak, dan untuk menangani puncak-puncak ini, disediakan kapasitas sumber daya sebagaimana yang ditunjukkan oleh garis berwarna oranye. Sumber daya dan energi yang digunakan untuk beban kerja ini tidak diindikasikan oleh area di bawah kurva permintaan, tetapi ditunjukkan oleh area di bawah garis kapasitas tersedia, karena kapasitas tersedia diperlukan untuk menangani kedua puncak ini.



Kurva permintaan dengan dua puncak berbeda yang membutuhkan kapasitas penyediaan tinggi.

Anda dapat menggunakan buffering atau throttling untuk melakukan modifikasi terhadap kurva permintaan dan meratakan puncak, yang artinya konsumsi energi menjadi lebih sedikit energi dan penyediaan kapasitas menjadi lebih rendah. Implementasikan throttling ketika klien Anda dapat mencoba ulang. Implementasikan buffering untuk menyimpan permintaan dan menunda pemrosesan ke lain waktu.



Efek throttling pada kurva permintaan dan kapasitas penyediaan.

Langkah-langkah implementasi

- Analisis permintaan klien untuk menentukan cara merespons permintaan. Pertanyaan yang harus dipertimbangkan antara lain:
  - Dapatkah permintaan ini diproses secara tak selaras?
  - Apakah klien memiliki kemampuan untuk melakukan percobaan ulang?

- Jika klien memiliki kemampuan untuk melakukan percobaan ulang, maka Anda dapat mengimplementasikan throttling, yang memberi tahu sumber bahwa jika sumber tidak dapat melayani permintaan pada saat ini maka sumber harus mencoba lagi nanti.
  - Anda dapat menggunakan Amazon API Gateway untuk menerapkan throttling.
- Untuk klien yang tidak dapat melakukan percobaan ulang, buffering harus diimplementasikan untuk meratakan kurva permintaan. Buffering akan menunda pemrosesan permintaan, sehingga aplikasi yang dijalankan pada tingkat yang berlainan dapat berkomunikasi secara efektif. Pendekatan berbasis buffering menggunakan antrean atau aliran untuk menerima pesan dari penghasil pesan. Pesan dibaca oleh konsumen dan diproses, sehingga pesan dapat dijalankan dengan tingkat yang memenuhi persyaratan-persyaratan bisnis konsumen.
  - Amazon Simple Queue Service (Amazon SQS) merupakan sebuah layanan terkelola yang memberikan antrean yang memungkinkan satu konsumen membaca pesan satu-persatu.
  - Amazon Kinesis memberikan aliran yang memungkinkan banyak konsumen untuk membaca pesan yang sama.
- Lakukan analisis terhadap permintaan secara keseluruhan, tingkat perubahan, dan waktu respons yang diperlukan untuk ukuran throttling atau buffering yang tepat.

Sumber daya

Dokumen terkait:

- Mulai menggunakan Amazon SQS
- Integrasi Aplikasi Menggunakan Antrian dan Pesan
- Mengelola dan memantau throttling API di beban kerja Anda
- Melakukan throttling terhadap API REST multi-tenant berjenjang dalam skala besar dengan menggunakan API Gateway
- Integrasi Aplikasi Menggunakan Antrian dan Pesan

Video terkait:

- AWS re:Invent 2022 - Pola integrasi aplikasi untuk layanan mikro
- AWS re:Invent 2023 - Penghematan cerdas: Strategi optimalisasi biaya Amazon EC2
- AWS re:Invent 2023 - Pola integrasi tingkat lanjut & kompromi untuk sistem yang digabungkan dengan metode penggabungan longgar

Perangkat lunak dan arsitektur

Implementasikan pola untuk melancarkan beban dan mempertahankan penggunaan yang tinggi dan konsisten atas sumber daya yang di-deploy guna meminimalkan sumber daya yang dipakai. Komponen dapat menjadi tidak aktif akibat kurangnya pemakaian, karena adanya perubahan perilaku pengguna dari waktu ke waktu. Revisi pola dan arsitektur untuk menggabungkan komponen dengan pemanfaatan rendah guna meningkatkan pemanfaatan secara keseluruhan. Pensiunkan komponen-komponen yang tidak lagi diperlukan. Pahami kinerja dari komponen-komponen beban kerja Anda, dan optimalkan komponen yang memakai sumber daya terbanyak. Ketahui perangkat yang digunakan oleh para pelanggan Anda untuk mengakses layanan Anda, dan implementasikan pola untuk meminimalkan kebutuhan pemutakhiran perangkat.

Praktik terbaik

- SUS03-BP01 Optimalkan perangkat lunak dan arsitektur untuk tugas yang asinkron dan terjadwal
- SUS03-BP02 Menyingkirkan atau memfaktor ulang komponen beban kerja yang jarang atau tidak pernah digunakan
- SUS03-BP03 Mengoptimalkan area kode yang memakai waktu atau sumber daya paling banyak
- SUS03-BP04 Mengoptimalkan dampak pada perangkat dan perlengkapan
- SUS03-BP05 Gunakan pola perangkat lunak dan arsitektur yang paling mendukung pola akses dan penyimpanan data

SUS03-BP01 Optimalkan perangkat lunak dan arsitektur untuk tugas yang asinkron dan terjadwal

Gunakan pola arsitektur dan perangkat lunak yang efisien seperti berbasis antrean untuk mempertahankan pemanfaatan sumber daya ter-deploy yang terus-menerus tinggi.

Anti-pola umum:

- Anda melakukan pengadaan sumber daya secara berlebihan di beban kerja cloud Anda untuk memenuhi lonjakan permintaan yang tidak terduga.
- Arsitektur Anda tidak memisahkan pengirim dan penerima pesan tak selaras berdasarkan komponen pesan.

Manfaat menjalankan praktik terbaik ini:

- Pola arsitektur dan perangkat lunak yang efisien dapat meminimalkan sumber daya tidak terpakai di dalam beban kerja Anda dan akan meningkatkan efisiensi secara keseluruhan.
- Anda dapat menskalakan pemrosesan tanpa terikat penerimaan pesan tak selaras.
- Melalui sebuah komponen perpesanan, Anda memiliki persyaratan ketersediaan yang longgar yang dapat Anda penuhi dengan sumber daya yang lebih sedikit.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Gunakan pola arsitektur yang efisien seperti arsitektur berbasis peristiwa yang dapat menghasilkan pemanfaatan komponen yang merata dan meminimalkan penyediaan berlebihan dalam beban kerja Anda. Menggunakan pola-pola arsitektur yang efisien akan meminimalkan sumber daya tidak aktif akibat kurangnya pemakaian yang disebabkan oleh perubahan permintaan seiring berjalannya waktu.

Pahami persyaratan-persyaratan komponen beban kerja Anda dan adopsi pola-pola arsitektur yang dapat meningkatkan pemanfaatan sumber daya secara keseluruhan. Pensiunkan komponen-komponen yang tidak lagi diperlukan.

Langkah-langkah implementasi

- Analisis permintaan untuk beban kerja Anda guna menentukan cara meresponsnya.
- Untuk permintaan atau tugas-tugas yang tidak memerlukan respons selaras, gunakan arsitektur berbasis antrean dan pekerja penskalaan otomatis untuk memaksimalkan pemanfaatan. Berikut ini adalah beberapa contoh kapan Anda mungkin perlu mempertimbangkan arsitektur berbasis antrean:

Mekanisme antrean	Deskripsi
Antrean tugas AWS Batch	Tugas AWS Batch dikirimkan ke antrean tugas dan menunggu untuk dijadwalkan berjalan di sebuah lingkungan komputasi.
Layanan Antrian Sederhana Amazon dan Instans Spot Amazon EC2	Memasangkan Amazon SQS dan Instans Spot untuk membangun arsitektur yang efisien dan tahan terhadap kesalahan.

- Untuk permintaan atau tugas yang dapat diproses kapan saja, gunakan mekanisme penjadwalan untuk memproses tugas-tugas yang ada dalam batch untuk mendapatkan efisiensi yang lebih tinggi. Berikut beberapa contoh mekanisme penjadwalan di AWS:

Mekanisme penjadwalan	Deskripsi
Penjadwal Amazon EventBridge	Kemampuan dari Amazon EventBridge yang memungkinkan Anda untuk membuat, menjalankan, dan mengelola tugas terjadwal dalam skala besar.
jadwal berbasis waktu AWS Glue	Tentukan jadwal berbasis waktu untuk perayap dan tugas Anda di AWS Glue.
Tugas yang dijadwalkan Amazon Elastic Container Service (Amazon ECS)	Amazon ECS mendukung pembuatan tugas terjadwal. Tugas terjadwal menggunakan aturan Amazon EventBridge untuk menjalank an tugas entah sesuai jadwal atau untuk menanggapi peristiwa EventBridge.
Penjadwal Instans	Konfigurasikan jadwal mulai dan berhenti untuk instans Amazon EC2 dan Amazon Relational Database Service.

- Jika Anda menggunakan mekanisme polling dan webhook dalam arsitektur Anda, maka gantilah dengan mekanisme peristiwa. Gunakan arsitektur berbasis peristiwa untuk membangun beban kerja yang sangat efisien.
- Manfaatkan nirserver di AWS untuk menghilangkan infrastruktur yang disediakan secara berlebihan.
- Sesuaikan ukuran dari setiap komponen yang ada dalam arsitektur Anda untuk menghindari sumber daya yang tidak aktif karena menunggu input.
  - Anda dapat menggunakan Rekomendasi Penyesuaian Ukuran di AWS Cost Explorer atau AWS Compute Optimizer untuk mengidentifikasi peluang penyesuaian ukuran.
  - Untuk detail selengkapnya, lihat Penentuan Ukuran yang Tepat: Menyediakan Instans yang Sesuai dengan Beban Kerja.

Sumber daya

Dokumen terkait:

- Apa itu Amazon Simple Queue Service?
- Apa itu Amazon MQ?
- Penskalaan berdasarkan Amazon SQS
- Apa itu AWS Step Functions?
- Apa itu AWS Lambda?
- Menggunakan AWS Lambda dengan Amazon SQS
- Apa itu Amazon EventBridge?
- Mengelola Alur Kerja Tak Selaras dengan API REST

Video terkait:

- AWS re:Invent 2023 - Menavigasi perjalanan ke arsitektur berbasis peristiwa nirserver
- AWS re:Invent 2023 - Menggunakan nirserver untuk arsitektur berbasis peristiwa & desain berbasis domain
- AWS re:Invent 2023 - Pola yang didorong peristiwa tingkat lanjut dengan Amazon EventBridge
- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- Pola Pesan Asinkron | Peristiwa AWS

Contoh terkait:

- Arsitektur berbasis peristiwa dengan Prosesor AWS Graviton dan Instans Spot Amazon EC2

SUS03-BP02 Menyingkirkan atau memfaktor ulang komponen beban kerja yang jarang atau tidak pernah digunakan

Singkirkan komponen yang tidak digunakan dan sudah tidak diperlukan, dan faktorkan ulang komponen dengan sedikit pemanfaatan, untuk meminimalkan limbah di beban kerja Anda.

Anti-pola umum:

- Anda tidak secara rutin memeriksa tingkat penggunaan masing-masing komponen beban kerja Anda.
- Anda tidak memeriksa dan menganalisis rekomendasi dari alat penyesuaian ukuran AWS seperti AWS Compute Optimizer.

Manfaat menerapkan praktik terbaik ini: Menghapus komponen yang tidak terpakai dapat meminimalkan pemborosan dan meningkatkan efisiensi keseluruhan beban kerja cloud Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Lakukan peninjauan terhadap beban kerja Anda untuk mengidentifikasi komponen-komponen yang pasif atau tidak digunakan. Tindakan ini adalah proses peningkatan yang berulang, yang dapat diinisiasi oleh perubahan-perubahan yang terjadi dalam permintaan atau rilis layanan cloud baru. Misalnya, penurunan waktu fungsi AWS Lambda yang signifikan dapat menjadi indikator yang menunjukkan kebutuhan untuk menurunkan ukuran memori. Selain itu, ketika AWS merilis layanan dan fitur baru, arsitektur dan layanan optimal untuk beban kerja Anda mungkin berubah.

Lakukan pemantauan terus menerus terhadap aktivitas beban kerja Anda dan carilah peluang untuk meningkatkan tingkat pemanfaatan masing-masing komponen. Dengan menyingkirkan komponen-komponen yang pasif dan melakukan aktivitas penyesuaian ukuran, Anda akan memenuhi persyaratan bisnis dengan menggunakan sumber daya cloud sesedikit mungkin.

Langkah-langkah implementasi

- Miliki inventaris sumber daya AWS Anda. Di AWS, Anda dapat mengaktifkan Penjelajah Sumber Daya AWS untuk menjelajahi dan mengatur sumber daya AWS Anda. Untuk detail selengkapnya, lihat AWS re:Invent 2022 - Cara mengelola sumber daya dan aplikasi dalam skala besar di AWS.
- Pantau dan tangkap metrik pemanfaatan untuk komponen penting beban kerja Anda (seperti pemanfaatan CPU, pemanfaatan memori, atau throughput jaringan dalam metrik Amazon CloudWatch).
- Identifikasi komponen-komponen yang tidak digunakan atau kurang dimanfaatkan dalam arsitektur Anda.
  - Untuk beban kerja yang stabil, periksa alat-alat pengatur ukuran AWS seperti AWS Compute Optimizer secara berkala untuk mengidentifikasi komponen yang langsam, tidak digunakan, atau kurang dimanfaatkan.

  - Untuk beban kerja sementara, lakukan evaluasi terhadap metrik-metrik pemanfaatan untuk mengidentifikasi komponen yang langsam, tidak digunakan, atau kurang dimanfaatkan.
- Pensiunkan komponen dan aset terkait (seperti gambar Amazon ECR) yang tidak diperlukan lagi.
  - Pembersihan Otomatis Citra yang Tidak Digunakan di Amazon ECR
  - Hapus volume Amazon Elastic Block Store (Amazon EBS) yang tidak terpakai dengan menggunakan AWS Config dan AWS Systems Manager
- Faktor ulang atau gabungkan komponen-komponen yang kurang dimanfaatkan dengan sumber daya lain untuk meningkatkan efisiensi pemanfaatan. Misalnya, Anda dapat menyediakan beberapa basis data kecil pada satu instans basis data Amazon RDS alih-alih menjalankan basis data pada instans individual yang kurang dimanfaatkan.
- Pahami sumber daya yang disediakan oleh beban kerja Anda untuk menyelesaikan sebuah unit kerja.

Sumber daya

Dokumen terkait:

- AWS Trusted Advisor
- Apa itu Amazon CloudWatch?
- Penentuan Ukuran yang Tepat: Menyediakan Instans yang Sesuai dengan Beban Kerja
- Mengoptimalkan biaya Anda dengan Rekomendasi Penentuan Ukuran yang Tepat

Video terkait:

- AWS re:Invent 2023 - Kapasitas, ketersediaan, efisiensi biaya: Pilih ketiganya

Contoh terkait:

- Optimalkan Pola Perangkat Keras dan Lakukan Observasi Terhadap KPI Keberlanjutan

SUS03-BP03 Mengoptimalkan area kode yang memakai waktu atau sumber daya paling banyak

Optimalkan kode Anda yang dijalankan di dalam berbagai macam komponen arsitektur Anda untuk meminimalkan penggunaan sumber daya sambil memaksimalkan performa.

Anti-pola umum:

- Anda mengabaikan optimalisasi kode Anda untuk penggunaan sumber daya.
- Anda biasanya merespons masalah-masalah performa dengan meningkatkan sumber daya.
- Proses pengembangan dan peninjauan kode Anda tidak melacak perubahan-perubahan performa.

Manfaat menerapkan praktik terbaik ini: Menggunakan kode yang efisien dapat meminimalkan penggunaan sumber daya dan meningkatkan kinerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Setiap area fungsional harus diperiksa, termasuk kode untuk aplikasi dengan arsitektur cloud, untuk mengoptimalkan penggunaan sumber daya dan performanya. Lakukan pemantauan terhadap performa beban kerja Anda secara terus menerus di lingkungan pembangunan dan produksi dan identifikasi peluang-peluang untuk meningkatkan snippet kode yang memiliki penggunaan sumber daya sangat tinggi. Adopsi proses peninjauan secara teratur untuk mengidentifikasi bug atau anti-pola yang ada di dalam kode Anda yang menggunakan sumber daya secara tidak efisien. Manfaatkan algoritme sederhana dan efisien yang memberikan hasil yang sama untuk kasus penggunaan Anda.

Langkah-langkah implementasi

- Gunakan bahasa pemrograman yang efisien: Gunakan sistem operasi dan bahasa pemrograman yang efisien untuk beban kerja. Untuk informasi mendetail tentang bahasa pemrograman yang hemat energi (termasuk Rust), lihat Keberlanjutan dengan Rust.
- Gunakan pendamping pengkodean AI: Pertimbangkan untuk menggunakan pendamping pengkodean AI seperti Amazon CodeWhisperer untuk menulis kode secara efisien.
- Lakukan otomatisasi peninjauan kode: Saat mengembangkan beban kerja Anda, adopsi proses peninjauan kode otomatis untuk meningkatkan kualitas dan mengidentifikasi bug dan antipola.
  - Lakukan otomatisasi peninjauan kode dengan Amazon CodeGuru Reviewer

  - Mendeteksi bug konkurensi dengan Amazon CodeGuru
  - Meningkatkan kualitas kode untuk aplikasi Python menggunakan Amazon CodeGuru
- Gunakan profiler kode: Gunakan profiler kode untuk mengidentifikasi area kode yang menggunakan waktu atau sumber daya paling banyak sebagai target optimasi.
  - Mengurangi jejak karbon organisasi Anda dengan Amazon CodeGuru Profiler
  - Memahami penggunaan memori di aplikasi Java Anda dengan Amazon CodeGuru Profiler
  - Meningkatkan pengalaman pelanggan dan mengurangi biaya dengan Amazon CodeGuru Profiler
- Pantau dan optimalkan: Gunakan sumber daya pemantauan berkelanjutan untuk mengidentifikasi komponen dengan persyaratan sumber daya yang tinggi atau konfigurasi mendekati optimal.
  - Ganti algoritme yang banyak memerlukan komputasi dengan versi algoritme yang lebih sederhana dan lebih efisien, yang akan memberikan hasil yang sama.
  - Singkirkan kode yang tidak perlu, seperti kode penyortiran dan pemformatan.
- Gunakan pemfaktoran ulang kode atau transformasi: Jelajahi kemungkinan transformasi kode Amazon Q untuk pemeliharaan dan peningkatan aplikasi.
  - Tingkatkan versi bahasa dengan Transformasi Kode Amazon Q
  - AWS re:Invent 2023 - Otomatiskan peningkatan & pemeliharaan aplikasi menggunakan Transformasi Kode Amazon Q

Sumber daya

Dokumen terkait:

- Apa itu Amazon CodeGuru Profiler?
- Instans FPGA
- SDK AWS di Alat-alat untuk Membangun di AWS

Video terkait:

- Tingkatkan Efisiensi Kode Menggunakan Amazon CodeGuru Profiler
- AWS re:Invent 2023 - Praktik terbaik untuk Amazon CodeWhisperer
- Lakukan Otomatisasi Peninjauan Kode dan Rekomendasi Kinerja Aplikasi dengan Amazon CodeGuru

Contoh terkait:

- Mengoptimalkan Kode dengan Amazon CodeGuru

SUS03-BP04 Mengoptimalkan dampak pada perangkat dan perlengkapan

Pahami perangkat dan perlengkapan yang digunakan dalam arsitektur Anda dan gunakan strategi untuk mengurangi penggunaannya. Tindakan ini dapat meminimalkan dampak beban kerja cloud Anda pada lingkungan secara keseluruhan.

Anti-pola umum:

- Anda mengabaikan dampak yang ditimbulkan oleh perangkat yang digunakan oleh pelanggan Anda terhadap lingkungan.
- Anda mengelola dan memperbarui sumber daya yang digunakan oleh pelanggan secara manual.

Manfaat menerapkan praktik terbaik ini: Menerapkan pola dan fitur perangkat lunak yang sudah dioptimalkan untuk perangkat pelanggan dapat mengurangi dampak lingkungan yang ditimbulkan beban kerja cloud Anda secara keseluruhan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Mengimplementasikan fitur dan pola perangkat lunak yang dioptimalkan untuk perangkat pelanggan dapat mengurangi dampaknya terhadap lingkungan dengan beberapa cara:

- Mengimplementasikan fitur baru yang kompatibel dengan versi lama dapat mengurangi jumlah penggantian perangkat keras.
- Mengoptimalkan aplikasi untuk beroperasi secara efisien di perangkat dapat membantu Anda mengurangi pemakaian energi dan memperpanjang masa pakai baterai (jika menggunakan tenaga baterai).
- Mengoptimalkan aplikasi untuk perangkat dapat juga mengurangi transfer data lewat jaringan.

Pahami perangkat dan perlengkapan yang digunakan dalam arsitektur Anda, siklus hidupnya yang diharapkan, dan dampak dari penggantian komponen-komponen tersebut. Implementasikan fitur dan pola perangkat lunak yang dapat membantu Anda meminimalkan pemakaian energi perangkat,

keharusan pelanggan untuk mengganti perangkat dan melakukan pemutakhiran perangkat secara manual.

Langkah-langkah implementasi

- Lakukan inventarisasi: Buatlah inventarisasi perangkat yang digunakan dalam arsitektur Anda. Perangkat dapat berupa perangkat seluler, tablet, perangkat IOT, lampu pintar, atau bahkan perangkat pintar yang dalam sebuah pabrik.
- Gunakan perangkat hemat energi: Pertimbangkan untuk menggunakan perangkat hemat energi dalam arsitektur Anda. Gunakan konfigurasi manajemen daya pada perangkat untuk masuk ke mode daya rendah saat tidak digunakan.
- Jalankan aplikasi yang efisien: Optimalkan aplikasi yang berjalan di perangkat:
  - Gunakan strategi seperti menjalankan tugas di latar belakang untuk mengurangi pemakaian energi.
  - Perhitungkan bandwidth jaringan dan latensi saat membangun payload, dan implementasikan kemampuan yang dapat membantu aplikasi bekerja dengan baik pada tautan yang memiliki bandwidth rendah dan latensi tinggi.
  - Ubah format payload dan file ke format optimal yang diperlukan oleh perangkat. Misalnya, Anda dapat menggunakan Amazon Elastic Transcoder atau AWS Elemental MediaConvert untuk mengonversi file media digital yang besar dan berkualitas tinggi ke dalam format yang dapat diputar ulang pengguna di perangkat seluler, tablet, browser web, dan televisi yang terhubung.
  - Lakukan aktivitas yang membutuhkan banyak komputasi di sisi server (seperti melakukan rendering gambar), atau gunakan streaming aplikasi untuk meningkatkan pengalaman pengguna pada perangkat yang lebih lama.
  - Segmentasikan dan beri nomor halaman pada output, terutama untuk sesi-sesi interaktif, guna mengelola payload dan membatasi persyaratan penyimpanan lokal.
- Libatkan pemasok: Bekerja samalah dengan pemasok perangkat yang menggunakan bahan berkelanjutan dan memberikan transparansi dalam rantai pasokan dan sertifikasi lingkungan mereka.
- Gunakan pembaruan lewat udara (OTA): Gunakan mekanisme otomatis lewat udara (OTA) untuk melakukan deployment pembaruan ke satu atau beberapa perangkat.
  - Anda dapat menggunakan sebuah pipeline CI/CD untuk memperbarui aplikasi seluler.
  - Anda dapat menggunakan AWS IoT Device Management untuk mengelola perangkat yang terhubung dari jarak jauh dalam skala besar.

- Gunakan device farm terkelola: Untuk menguji fitur baru dan pembaruan, gunakan device farm terkelola dengan set perangkat keras representatif dan ulang pengembangan untuk memaksimalkan perangkat yang didukung. Untuk detail selengkapnya, lihat SUS06-BP04 Menggunakan device farm terkelola untuk pengujian.
- Pemantau dan peningkatan terus-menerus: Lacak penggunaan energi perangkat untuk mengidentifikasi area-area yang bisa diperbaiki. Gunakan teknologi atau praktik terbaik terbaru untuk memperbaiki dampak lingkungan yang ditimbulkan oleh perangkat-perangkat tersebut.

Sumber daya

Dokumen terkait:

- Apa itu AWS Device Farm?
- Dokumentasi AppStream 2.0
- NICE DCV
- Tutorial OTA untuk memperbarui firmware pada perangkat yang menjalankan FreeRTOS
- Mengoptimalkan Perangkat IoT Anda untuk Kelestarian Lingkungan

Video terkait:

- AWS re:Invent 2023 - Tingkatkan kualitas aplikasi seluler dan web Anda dengan menggunakan AWS Device Farm

SUS03-BP05 Gunakan pola perangkat lunak dan arsitektur yang paling mendukung pola akses dan penyimpanan data

Pahami bagaimana data digunakan di dalam beban kerja Anda, dipakai oleh pengguna Anda, ditransfer, dan disimpan. Gunakan pola perangkat lunak dan arsitektur yang paling mendukung akses dan penyimpanan data untuk meminimalkan sumber daya komputasi, jaringan, dan penyimpanan yang diperlukan untuk mendukung beban kerja.

Anti-pola umum:

- Anda berasumsi bahwa semua beban kerja memiliki pola penyimpanan data dan akses data yang serupa.

- Anda hanya menggunakan satu tingkat penyimpanan, dengan anggapan semua beban kerja masuk dalam tingkat tersebut.
- Anda berasumsi bahwa pola akses data tidak akan berubah.
- Arsitektur Anda mendukung potensi lonjakan akses data yang tinggi, yang dapat mengakibatkan sumber daya tetap menjadi tidak aktif dalam sebagian besar waktu.

Manfaat menerapkan praktik terbaik ini: Memilih dan mengoptimalkan arsitektur Anda berdasarkan pola akses data dan pola penyimpanan data akan membantu Anda untuk mengurangi kompleksitas pengembangan dan meningkatkan pemanfaatan secara keseluruhan. Memahami kapan harus menggunakan tabel global, partisi data, dan caching akan membantu Anda untuk mengurangi biaya operasional dan menskalakan sesuai kebutuhan beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Gunakan pola-pola arsitektur dan perangkat lunak yang paling sesuai dengan karakteristik data dan pola akses Anda. Misalnya, gunakan arsitektur data modern di AWS yang memungkinkan Anda untuk menggunakan layanan yang dibuat khusus yang dioptimalkan untuk kasus penggunaan analitik unik Anda. Pola-pola arsitektur ini akan memungkinkan pemrosesan data yang efisien dan mengurangi penggunaan sumber daya.

Langkah-langkah implementasi

- Lakukan analisis terhadap karakteristik data dan pola akses Anda untuk mengidentifikasi konfigurasi yang tepat untuk sumber daya cloud Anda. Karakteristik utama yang perlu dipertimbangkan antara lain:
  - Jenis data: terstruktur, semi-terstruktur, tidak terstruktur
  - Pertumbuhan data: terbatas, tidak terbatas
  - Ketahanan data: persisten, sementara, transien
  - Pola akses baca atau tulis, frekuensi pembaruan, berfluktuasi, atau konsisten
- Gunakan pola-pola arsitektur yang paling mendukung pola akses dan penyimpanan data.
  - Pola untuk memungkinkan persistensi data
  - Mari Merancang! Arsitektur data modern
  - Basis data di AWS: Alat yang Tepat untuk Tugas yang Tepat
- Gunakan teknologi yang berfungsi secara native dengan data terkompresi.

  - Format file Dukungan Kompresi Athena
  - Opsi Format untuk Input dan Output ETL di AWS Glue
  - Memuat file data terkompresi dari Amazon S3 dengan Amazon Redshift
- Gunakan layanan analitik yang dibuat khusus untuk pemrosesan data dalam arsitektur Anda. Untuk detail tentang layanan analitik yang dibuat khusus AWS, lihat AWS re:Invent 2022 - Membangun arsitektur data modern di AWS.
- Gunakan mesin basis data yang paling mendukung pola-pola kueri dominan Anda. Kelola indeks basis data Anda untuk memastikan pembuatan kueri yang efisien. Untuk detail lebih lanjut, lihat Basis Data AWS dan AWS re:Invent 2022 - - Melakukan modernisasi aplikasi dengan basis data yang dibuat khusus.
- Pilihlah protokol-protokol jaringan yang dapat mengurangi jumlah kapasitas jaringan yang dipakai di arsitektur Anda.

Sumber daya

Dokumen terkait:

- MENYALIN dari format data kolom dengan Amazon Redshift
- Mengonversi Format Catatan Input Anda di Firehose
- Meningkatkan kinerja kueri di Amazon Athena dengan Mengonversi ke Format Kolom
- Memantau muatan DB dengan Wawasan Performa di Amazon Aurora
- Memantau muatan DB dengan Wawasan Performa di Amazon RDS
- Kelas penyimpanan Amazon S3 Intelligent-Tiering
- Bangun toko peristiwa CQRS dengan Amazon DynamoDB

Video terkait:

- AWS re:Invent 2022 - Membangun arsitektur jala data di AWS
- AWS re:Invent 2023 - Memahami lebih dalam tentang Amazon Aurora dan inovasinya
- AWS re:Invent 2023 - Meningkatkan efisiensi Amazon EBS dan menjadi lebih hemat biaya
- AWS re:Invent 2023 - Mengoptimalkan harga dan kinerja penyimpanan dengan Amazon S3
- AWS re:Invent 2023 - Membangun dan mengoptimalkan danau data di Amazon S3
- AWS re:Invent 2023 - Pola yang didorong peristiwa tingkat lanjut dengan Amazon EventBridge

Contoh terkait:

- Lokakarya Basis Data yang Dibuat Khusus AWS
- Hari Imersi Arsitektur Data Modern AWS
- Membangun Jala Data di AWS

Manajemen data

Implementasikan praktik manajemen data untuk mengurangi penyediaan penyimpanan yang diperlukan untuk mendukung beban kerja Anda, serta untuk mengurangi sumber daya yang diperlukan untuk menggunakannya. Pahami data Anda, dan gunakan konfigurasi dan teknologi penyimpanan penggunaan yang paling sesuai untuk mendukung nilai bisnis data dan cara data digunakan. Buat siklus hidup data agar penyimpanan menjadi lebih efisien dan memiliki kinerja lebih rendah ketika persyaratan berkurang, dan hapus data yang tidak lagi diperlukan.

Praktik terbaik

- SUS04-BP01 Mengimplementasikan kebijakan klasifikasi data
- SUS04-BP02 Menggunakan teknologi yang mendukung pola akses dan penyimpanan data
- SUS04-BP03 Menggunakan kebijakan untuk mengelola siklus hidup set data Anda
- SUS04-BP04 Menggunakan elastisitas dan otomatisasi untuk memperluas sistem file atau penyimpanan blok
- SUS04-BP05 Menyingkirkan data yang tidak diperlukan atau redundan
- SUS04-BP06 Menggunakan sistem file atau penyimpanan bersama untuk mengakses data umum
- SUS04-BP07 Meminimalkan perpindahan data di jaringan
- SUS04-BP08 Hanya mencadangkan data saat sulit untuk dibuat ulang

SUS04-BP01 Mengimplementasikan kebijakan klasifikasi data

Kelompokkan data untuk memahami tingkat kekritisannya terhadap hasil bisnis dan pilih tingkat penyimpanan hemat energi yang tepat untuk menyimpan data.

Anti-pola umum:

- Anda tidak mengidentifikasi aset data yang memiliki karakteristik serupa (seperti sensitivitas, kekritisan bisnis, atau persyaratan peraturan) yang diproses atau disimpan.

- Anda belum mengimplementasikan katalog data untuk menginventarisasi aset data Anda.

Manfaat menerapkan praktik terbaik ini: Menerapkan kebijakan klasifikasi data memungkinkan Anda menentukan tingkat penyimpanan data yang paling hemat energi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Klasifikasi data melibatkan identifikasi jenis-jenis data yang sedang diproses dan disimpan dalam sistem informasi yang dimiliki atau dioperasikan oleh organisasi. Klasifikasi data juga melibatkan penentuan tingkat kekritisan data dan kemungkinan dampak-dampak yang ditimbulkan saat terjadi peretasan, kehilangan, atau penyalahgunaan data.

Implementasikan kebijakan klasifikasi data dengan bekerja mundur dari penggunaan data kontekstual dan membuat sebuah skema kategorisasi yang mempertimbangkan tingkat kekritisan set data tertentu terhadap operasi organisasi.

Langkah-langkah implementasi

- Lakukan inventarisasi data: Lakukan inventarisasi atas berbagai jenis data yang ada untuk beban kerja Anda.
- Kelompokkan data: Tentukan kekritisan, kerahasiaan, integritas, dan ketersediaan data berdasarkan risiko terhadap organisasi. Gunakan persyaratan-persyaratan ini untuk mengelompokkan data ke dalam satu tingkat klasifikasi data yang Anda adopsi. Sebagai contoh, lihat Empat langkah sederhana untuk mengklasifikasikan data Anda dan mengamankan perusahaan rintisan Anda.
- Tentukan tingkat dan kebijakan klasifikasi data: Untuk masing-masing kelompok data, tentukan tingkat klasifikasi datanya (misalnya, publik atau rahasia) dan kebijakan penanganannya. Berikan tag pada data sebagaimana mestinya. Untuk detail selengkapnya tentang kategori data, lihat laporan resmi Klasifikasi Data.
- Tinjau secara berkala: Lakukan peninjauan dan audit terhadap lingkungan Anda secara berkala untuk data yang tidak diberi tag dan tidak diklasifikasikan. Gunakan otomatisasi untuk mengidentifikasi data ini, dan klasifikasikan serta berikan tag pada data dengan semestinya. Sebagai contoh, lihat Katalog Data dan perayap di AWS Glue.
- Buat katalog data: Buatlah katalog data yang menyediakan kemampuan audit dan tata kelola.
- Dokumentasi: Buatlah dokumentasi kebijakan klasifikasi data dan prosedur penanganan untuk setiap kelas data.

Sumber daya

Dokumen terkait:

- Memanfaatkan AWS Cloud untuk Mendukung Klasifikasi Data
- Kebijakan pemberian tag dari AWS Organizations

Video terkait:

- AWS re:Invent 2022 - Mengaktifkan kelincahan dengan tata kelola data di AWS
- AWS re:Invent 2023 - Perlindungan dan ketahanan data dengan penyimpanan AWS

SUS04-BP02 Menggunakan teknologi yang mendukung pola akses dan penyimpanan data

Gunakan teknologi penyimpanan yang paling mendukung cara data Anda diakses dan disimpan untuk meminimalkan sumber daya yang disediakan sambil mendukung beban kerja Anda.

Anti-pola umum:

- Anda berasumsi bahwa semua beban kerja memiliki pola penyimpanan data dan akses data yang serupa.
- Anda hanya menggunakan satu tingkat penyimpanan, dengan anggapan semua beban kerja masuk dalam tingkat tersebut.
- Anda berasumsi bahwa pola akses data tidak akan berubah.

Manfaat menerapkan praktik terbaik ini: Memilih dan mengoptimalkan teknologi penyimpanan Anda berdasarkan pola akses dan penyimpanan data akan membantu Anda mengurangi sumber daya cloud yang diperlukan untuk memenuhi kebutuhan bisnis dan meningkatkan keseluruhan efisiensi beban kerja cloud.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Pilih solusi penyimpanan yang paling sesuai dengan pola akses Anda, atau pertimbangkan untuk mengubah pola akses tersebut agar sesuai dengan solusi penyimpanan untuk meraih efisiensi kinerja yang maksimal.

Langkah-langkah implementasi

- Lakukan evaluasi terhadap karakteristik data dan akses: Evaluasi karakteristik dan pola akses data Anda untuk mengumpulkan karakteristik utama kebutuhan penyimpanan Anda. Karakteristik utama yang perlu dipertimbangkan antara lain:
  - Jenis data: terstruktur, semi-terstruktur, tidak terstruktur
  - Pertumbuhan data: terbatas, tidak terbatas
  - Ketahanan data: persisten, sementara, transien
  - Pola akses baca atau tulis, frekuensi, berfluktuasi, atau konsisten
- Pilih teknologi penyimpanan yang tepat: Migrasikan data ke teknologi penyimpanan yang tepat yang mendukung karakteristik dan pola akses data Anda. Berikut adalah beberapa contoh teknologi penyimpanan AWS serta karakteristik utamanya:

Tipe	Teknologi	Karakteristik utama
Penyimpanan objek	Amazon S3	Sebuah layanan penyimpan an objek yang memiliki skalabilitas tak terbatas, ketersediaan tinggi, dan berbagai opsi aksesibilitas. Mentransfer dan mengakses objek-objek yang masuk dan keluar dari Amazon S3 dapat dilakukan dengan menggunak an sebuah layanan, seperti Akselerasi Transfer atau Titik Akses, untuk mendukung lokasi, kebutuhan keamanan, dan pola akses Anda.

Tipe	Teknologi	Karakteristik utama
Penyimpanan pengarsipan	Amazon S3 Glacier	Kelas penyimpanan Amazon S3 yang dibuat untuk pengarsipan data.
Sistem file bersama	Amazon Elastic File System (Amazon EFS)	Sistem file yang dapat dipasang dan dapat diakses oleh berbagai jenis solusi komputasi. Amazon EFS secara otomatis akan meningkatkan dan menyusutkan penyimpanan serta meningkatkan optimalisasi kinerjanya untuk menghasilkan latensi rendah yang konsisten.
Sistem file bersama	Amazon FSx	Dibangun berdasarkan solusi komputasi AWS terbaru untuk mendukung empat sistem file yang umum digunakan: NetApp ONTAP, OpenZFS, Windows File Server, dan Lustre. Latensi, throughput, dan IOPS Amazon FSx berbeda-beda untuk setiap sistem file dan hal ini harus Anda pertimbangkan saat memilih sistem file yang tepat untuk kebutuhan beban kerja Anda.

Tipe	Teknologi	Karakteristik utama
Penyimpanan blok	Amazon Elastic Block Store (Amazon EBS)	Layanan penyimpanan blok yang dapat diskalakan dan mempunyai performa tinggi yang dirancang untuk Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS mencakup penyimpanan yang didukung SSD untuk beban kerja transaksional, intensif IOPS, dan penyimpanan yang didukung HDD untuk beban kerja yang intensif throughput.
Basis data relasional	Amazon Aurora, Amazon RDS, Amazon Redshift	Didesain untuk mendukung transaksi-transaksi ACID (atomisitas, konsistensi, isolasi, durabilitas), dan mempertahankan integritas referensial serta konsistensi data yang tinggi. Banyak sekali aplikasi tradisional, perencanaan sumber daya perusahaan (ERP), manajemen hubungan pelanggan (CRM), dan sistem perdagangan elektronik yang menggunakan basis data relasional untuk menyimpan data mereka.

Tipe	Teknologi	Karakteristik utama
Basis data nilai-kunci	Amazon DynamoDB	Dioptimalkan untuk pola akses umum, biasanya digunakan untuk menyimpan dan mengambil data dalam volume besar. Aplikasi web dengan lalu lintas tinggi, sistem perdagangan elektronik, dan aplikasi gaming merupakan kasus penggunaan umum untuk basis data nilai kunci.

- Otomatiskan alokasi penyimpanan: Untuk sistem penyimpanan dengan ukuran tetap, seperti Amazon EBS atau Amazon FSx, pantau ruang penyimpanan yang tersedia dan otomatiskan alokasi penyimpanan saat ambang batas tercapai. Anda dapat memanfaatkan Amazon CloudWatch untuk mengumpulkan dan menganalisis berbagai metrik untuk Amazon EBS dan Amazon FSx.
- Pilih kelas penyimpanan yang tepat: Pilihlah kelas penyimpanan yang sesuai untuk data Anda.
  - Kelas penyimpanan Amazon S3 dapat dikonfigurasi pada tingkat objek. Satu bucket dapat berisi objek-objek yang disimpan di semua kelas penyimpanan.
  - Anda dapat menggunakan kebijakan Siklus Hidup Amazon S3 untuk mengalihkan objek secara otomatis antar kelas-kelas penyimpanan atau menghapus data tanpa perubahan aplikasi apa pun. Secara umum, Anda harus memilih mana yang paling penting (melakukan kompromi) antara efisiensi sumber daya, latensi akses, dan keandalan saat mempertimbangkan semua mekanisme penyimpanan ini.

Sumber daya

Dokumen terkait:

- Tipe volume Amazon EBS
- Penyimpanan instans Amazon EC2
- Amazon S3 Intelligent-Tiering
- Karakteristik I/O Amazon EBS

- Menggunakan kelas penyimpanan Amazon S3
- Apa Itu Amazon S3 Glacier?

Video terkait:

- AWS re:Invent 2023 - Meningkatkan efisiensi Amazon EBS dan menjadi lebih hemat biaya
- AWS re:Invent 2023 - Mengoptimalkan harga dan kinerja penyimpanan dengan Amazon S3
- AWS re:Invent 2023 - Membangun dan mengoptimalkan danau data di Amazon S3
- AWS re:Invent 2022 - Membangun arsitektur data modern di AWS
- AWS re:Invent 2022 - Melakukan modernisasi aplikasi dengan basis data yang dibuat khusus
- AWS re:Invent 2022 - Membangun arsitektur jala data di AWS
- AWS re:Invent 2023 - Memahami lebih dalam tentang Amazon Aurora dan inovasinya
- AWS re:Invent 2023 - Pemodelan data tingkat lanjut dengan Amazon DynamoDB

Contoh terkait:

- Contoh-contoh Amazon S3
- Lokakarya Basis Data yang Dibuat Khusus AWS
- Basis Data untuk Pengembang
- Hari Imersi Arsitektur Data Modern AWS
- Membangun Jala Data di AWS

SUS04-BP03 Menggunakan kebijakan untuk mengelola siklus hidup set data Anda

Kelola siklus hidup semua data Anda dan terapkan penghapusan secara otomatis untuk meminimalkan total penyimpanan yang diperlukan untuk beban kerja Anda.

Anti-pola umum:

- Anda menghapus data secara manual.
- Anda tidak menghapus data beban kerja Anda sama sekali.
- Anda tidak mengalihkan data ke tingkat penyimpanan yang lebih hemat energi berdasarkan persyaratan retensi dan aksesnya.

Manfaat menerapkan praktik terbaik ini: Menggunakan kebijakan siklus hidup data memastikan akses dan retensi data yang efisien dalam beban kerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Set data biasanya memiliki persyaratan retensi dan akses data yang berbeda-beda selama masa hidupnya. Misalnya, aplikasi Anda mungkin memerlukan akses yang sering ke sejumlah set data dalam jangka waktu terbatas. Setelah itu, set-set data tersebut jarang diakses.

Untuk mengelola set data Anda secara efisien di sepanjang siklus hidupnya, konfigurasikan kebijakan siklus hidup, yakni aturan yang menetapkan cara menangani set data.

Dengan Aturan konfigurasi siklus hidup, Anda dapat meminta layanan penyimpanan tertentu untuk mengalihkan suatu set data ke tingkat penyimpanan yang lebih hemat energi, mengarsipkannya, atau menghapusnya.

Langkah-langkah implementasi

- Klasifikasikan set data dalam beban kerja Anda.
- Tetapkan prosedur penanganan untuk setiap kelas data.
- Atur kebijakan siklus hidup otomatis untuk menegakkan aturan siklus hidup. Berikut ini adalah beberapa cara menyiapkan kebijakan siklus hidup otomatis untuk berbagai layanan penyimpanan AWS:

Layanan penyimpanan	Cara menyetel kebijakan siklus hidup otomatis
Amazon S3	Anda dapat menggunakan Siklus Hidup Amazon S3 untuk mengelola objek-objek Anda di sepanjang siklus hidupnya. Jika pola akses Anda tidak diketahui, berubah, atau tidak dapat diprediksi, Anda dapat menggunakan Amazon S3 Intelligent-Tiering, yang akan memantau pola akses dan secara otomatis memindahkan objek yang belum diakses ke tingkat akses yang berbiaya lebih rendah. Anda dapat memanfaatkan metrik Lensa Penyimpanan Amazon S3 untuk melakukan

Layanan penyimpanan	Cara menyetel kebijakan siklus hidup otomatis
	identifikasi terhadap peluang dan celah pengoptimalan dalam manajemen siklus hidup.
Amazon Elastic Block Store	Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi pembuatan, retensi, dan penghapusan snapshot Amazon EBS dan AMI yang didukung Amazon EBS.
Sistem File Elastis Amazon	Manajemen siklus hidup Amazon EFS secara otomatis akan mengelola penyimpanan file untuk sistem file Anda.
Amazon Elastic Container Registry	Kebijakan siklus hidup Amazon ECR akan melakukan otomatisasi pembersihan image kontainer Anda berdasarkan image yang kedaluwarsa berdasarkan usia atau hitungan.
AWS Elemental MediaStore	Anda dapat menggunakan kebijakan siklus hidup objek yang mengatur berapa lama objek harus disimpan dalam kontainer MediaStore.

- Hapus volume, snapshot, dan data yang tidak digunakan yang sudah melebihi masa retensinya. Manfaatkan fitur-fitur layanan native seperti Amazon DynamoDB Time To Live atau retensi log Amazon CloudWatch untuk penghapusan.
- Lakukan agregasi dan kompresi data, jika memungkinkan, berdasarkan aturan siklus hidup.

Sumber daya

Dokumen terkait:

- Optimalkan aturan Siklus Hidup Amazon S3 Anda dengan Analisis Kelas Penyimpanan Amazon S3
- Mengevaluasi Sumber Daya dengan Aturan AWS Config

Video terkait:

- AWS re:Invent 2021 - Praktik terbaik Siklus Hidup Amazon S3 untuk mengoptimalkan pengeluaran penyimpanan Anda
- AWS re:Invent 2023 - Mengoptimalkan harga dan kinerja penyimpanan dengan Amazon S3
- Sederhanakan Siklus Hidup Data Anda dan Optimalkan Biaya Penyimpanan Dengan Siklus Hidup Amazon S3
- Menggunakan Lensa Penyimpanan Amazon S3 untuk Mengurangi Biaya Penyimpanan Anda

SUS04-BP04 Menggunakan elastisitas dan otomatisasi untuk memperluas sistem file atau penyimpanan blok

Gunakan elastisitas dan otomatisasi untuk memperluas sistem file atau penyimpanan blok seiring pertumbuhan data untuk meminimalkan total penyimpanan yang disediakan.

Anti-pola umum:

- Anda membeli sistem file atau penyimpanan blok besar untuk keperluan di waktu mendatang.
- Anda memberikan persediaan berlebih untuk operasi input dan output per detik (IOPS) sistem file Anda.
- Anda tidak memantau pemanfaatan volume data Anda.

Manfaat menerapkan praktik terbaik ini: Meminimalkan penyediaan berlebih untuk sistem penyimpanan akan mengurangi sumber daya yang tidak digunakan dan meningkatkan efisiensi keseluruhan beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Buat sistem file dan penyimpanan blok dengan alokasi ukuran, throughput, dan latensi yang sesuai untuk beban kerja Anda. Gunakan elastisitas dan otomatisasi untuk memperluas sistem file atau penyimpanan blok seiring pertumbuhan data tanpa perlu melakukan penyediaan layanan penyimpanan yang berlebihan.

Langkah-langkah implementasi

- Untuk penyimpanan yang berukuran tetap, seperti Amazon EBS, verifikasi bahwa Anda memantau kapasitas penyimpanan yang digunakan dengan keseluruhan ukuran penyimpanan, dan buat

otomatisasi jika memungkinkan untuk meningkatkan ukuran penyimpanan saat mencapai ambang batas.

- Gunakan volume elastis dan layanan data blok terkelola untuk mengotomatisasi alokasi penyimpanan tambahan seiring tumbuhnya data persisten Anda. Sebagai contoh, Anda dapat menggunakan Volume Elastis Amazon EBS untuk mengubah ukuran volume, mengubah tipe volume, atau menyesuaikan performa volume Amazon EBS Anda.
- Pilih kelas penyimpanan, mode performa, dan mode throughput yang tepat untuk sistem file Anda guna memenuhi kebutuhan bisnis, tidak melebihinya.
  - Performa Amazon EFS
  - Performa volume Amazon EBS pada instans Linux
- Atur target tingkat pemanfaatan untuk volume data Anda, dan ubah ukuran volume di luar rentang yang diperkirakan.
- Sesuaikan ukuran volume hanya-baca agar sesuai dengan data.
- Migrasikan data ke penyimpanan objek untuk menghindari penyediaan kapasitas yang berlebihan dari ukuran volume tetap di penyimpanan blok.
- Secara rutin tinjau volume elastis dan sistem file untuk menghentikan volume yang tidak aktif dan memperkecil sumber daya dengan penyediaan berlebihan agar sesuai dengan ukuran data saat ini.

Sumber daya

Dokumen terkait:

- Perluas sistem file setelah mengubah ukuran volume EBS
- Modifikasi volume EBS menggunakan Volume Elastis Amazon EBS
- Dokumentasi Amazon FSx
- Apa itu Sistem File Elastis Amazon?

Video terkait:

- Memahami Lebih Dalam Volume Elastis Amazon EBS
- Amazon EBS dan Strategi Optimalisasi Snapshot untuk Kinerja dan Penghematan Biaya yang Lebih Baik
- Mengoptimalkan Amazon EFS untuk biaya dan kinerja, menggunakan praktik terbaik

SUS04-BP05 Menyingkirkan data yang tidak diperlukan atau redundan

Hapus data yang tidak diperlukan atau redundan untuk meminimalkan sumber daya penyimpanan yang diperlukan untuk menyimpan set data Anda.

Anti-pola umum:

- Anda menduplikasi data yang dapat diperoleh atau dibuat ulang dengan mudah.
- Anda mencadangkan semua data tanpa mempertimbangkan tingkat kekritisannya.
- Anda menghapus data tidak rutin, hanya pada peristiwa operasional, atau tidak menghapusnya sama sekali.
- Anda menyimpan data secara redundan dengan mengabaikan durabilitas layanan penyimpanan.
- Anda mengaktifkan penentuan versi Amazon S3 tanpa alasan bisnis apa pun.

Manfaat menerapkan praktik terbaik ini: Menghapus data yang tidak dibutuhkan akan mengurangi ukuran penyimpanan yang diperlukan untuk beban kerja Anda dan dampak lingkungan yang ditimbulkan beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Jangan menyimpan data yang tidak Anda perlukan. Otomatiskan penghapusan data yang tidak diperlukan. Gunakan teknologi yang menghilangkan data ganda pada tingkat file dan blok. Manfaatkan fitur replikasi dan redundansi data native dari layanan.

Langkah-langkah implementasi

- Evaluasi apakah Anda dapat menghindari penyimpanan data dengan menggunakan set data yang tersedia untuk umum yang ada di dalam AWS Data Exchange dan Data Terbuka di AWS.
- Gunakan mekanisme yang dapat membatalkan duplikasi data pada tingkat blok dan objek. Berikut ini adalah beberapa contoh cara membatalkan duplikasi data di AWS:

Layanan penyimpanan	Mekanisme deduplikasi
Amazon S3	Gunakan AWS Lake Formation FindMatches untuk menemukan rekaman yang cocok di seluruh set data (termasuk yang tanpa

Layanan penyimpanan	Mekanisme deduplikasi
	pengenal) dengan menggunakan FindMatches MLTransform yang baru.
Amazon FSx	Gunakan deduplikasi data di Amazon FSx untuk Windows.
Snapshot Amazon Elastic Block Store	Snapshot bertahap pencadangan, yang berarti bahwa hanya blok di perangkat yang diubah setelah snapshot terbaru Anda disimpan.

- Analisis akses data untuk mengidentifikasi data yang tidak diperlukan. Otomatiskan kebijakan siklus hidup. Manfaatkan fitur-fitur layanan bawaan native seperti Amazon DynamoDB Time To Live, Siklus Hidup Amazon S3, atau retensi log Amazon CloudWatch untuk dihapus.
- Gunakan kemampuan virtualisasi data di AWS untuk mempertahankan data di sumbernya dan menghindari duplikasi data.
  - Virtualisasi Data Cloud Native di AWS
  - Optimalkan Pola Data menggunakan Pembagian Data Amazon Redshift
- Gunakan teknologi pencadangan yang dapat membuat cadangan bertahap.
- Manfaatkan daya tahan Amazon S3 dan replikasi Amazon EBS untuk memenuhi tujuan daya tahan Anda alih-alih teknologi yang dikelola sendiri (seperti susunan disk independen (RAID) yang redundan).
- Pusatkan log dan lacak data, batalkan duplikasi entri log yang identik, dan buat mekanisme untuk menyesuaikan verbositas saat diperlukan.
- Pra-isi cache hanya saat ada alasan yang dibenarkan.
- Lakukan pemantauan dan otomatisasi cache untuk menyesuaikan ukuran cache dengan tepat.
- Singkirkan deployment dan aset usang dari penyimpanan objek dan cache edge saat mendorong versi baru untuk beban kerja Anda.

Sumber daya

Dokumen terkait:

- Mengubah retensi data log di CloudWatch Logs
- Pembatalan duplikasi data di Amazon FSx for Windows File Server

- Fitur Amazon FSx untuk ONTAP meliputi pembatalan duplikasi data
- Membatalkan Validasi File di Amazon CloudFront
- Menggunakan AWS Backup untuk mencadangkan dan memulihkan sistem file Amazon EFS
- Apa itu Log Amazon CloudWatch?
- Bekerja dengan cadangan di Amazon RDS
- Mengintegrasikan dan membatalkan duplikasi set data menggunakan AWS Lake Formation

Video terkait:

- Kasus Penggunaan Berbagi Data Amazon Redshift

Contoh terkait:

- Bagaimana cara menganalisis log akses server Amazon S3 menggunakan Amazon Athena?

SUS04-BP06 Menggunakan sistem file atau penyimpanan bersama untuk mengakses data umum

Adopsi sistem file atau penyimpanan bersama untuk menghindari duplikasi data dan memungkinkan infrastruktur yang lebih efisien untuk beban kerja Anda.

Anti-pola umum:

- Anda menyediakan penyimpanan untuk setiap klien secara individu.
- Anda tidak melepaskan volume data dari klien yang tidak aktif.
- Anda tidak memberikan akses ke penyimpanan di semua platform dan sistem.

Manfaat menerapkan praktik terbaik ini: Menggunakan sistem file atau penyimpanan bersama akan memungkinkan Anda untuk berbagi data ke satu atau beberapa konsumen tanpa harus menyalin data. Hal ini membantu mengurangi sumber daya penyimpanan yang diperlukan untuk beban kerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Jika Anda memiliki beberapa pengguna atau aplikasi yang mengakses set data yang sama, penggunaan teknologi penyimpanan bersama sangatlah penting untuk menggunakan infrastruktur yang efisien untuk beban kerja Anda. Teknologi penyimpanan bersama memberikan lokasi sentral untuk menyimpan dan mengelola set data dan menghindari duplikasi data. Teknologi ini juga memastikan konsistensi data di berbagai sistem yang berlainan. Lebih lanjut, teknologi penyimpanan bersama memungkinkan penggunaan daya komputasi yang lebih efisien, karena beberapa sumber daya komputasi dapat mengakses dan memproses data pada saat yang sama secara paralel.

Hanya ambil data dari layanan penyimpanan bersama ini sesuai kebutuhan dan lepaskan volume yang tidak digunakan untuk membebaskan sumber daya.

Langkah-langkah implementasi

- Migrasikan data ke penyimpanan bersama ketika data memiliki beberapa pemakai. Berikut beberapa contoh teknologi penyimpanan bersama di AWS:

Opsi penyimpanan	Kapan harus digunakan
Amazon EBS Multi-Lampiran	Dengan Multi-Lampiran Amazon EBS, Anda dapat memasang satu volume SSD IOPS yang Disediakan (io1 atau io2) ke banyak instans yang berada dalam Zona Ketersedi aan yang sama.
Amazon EFS	Lihat Kapan Harus Memilih Amazon EFS.
Amazon FSx	Lihat Memilih Sistem File Amazon FSx.
Amazon S3	Aplikasi yang tidak memerlukan struktur sistem file dan didesain untuk berfungsi dengan penyimpanan objek dapat menggunak an Amazon S3 sebagai solusi penyimpanan objek yang dapat diskalakan besar-besaran, tahan lama, dan murah.

- Salin data ke sistem file atau ambil data dari sistem file bersama hanya jika dibutuhkan. Sebagai contoh, Anda dapat membuat sistem file Amazon FSx for Lustre yang didukung oleh Amazon S3 dan hanya memuat subset data yang diperlukan untuk memproses pekerjaan ke Amazon FSx.
- Hapus data yang sesuai untuk pola penggunaan Anda seperti yang diuraikan dalam SUS04-BP03 Menggunakan kebijakan untuk mengelola siklus hidup set data Anda.
- Lepaskan volume dari klien yang tidak menggunakannya secara aktif.

Sumber daya

Dokumen terkait:

- Menautkan sistem file Anda ke bucket Amazon S3
- Menggunakan Amazon EFS untuk AWS Lambda di aplikasi nirserver Anda
- Amazon EFS Intelligent-Tiering Mengoptimalkan Biaya Beban Kerja dengan Mengubah Pola Akses
- Menggunakan Amazon FSx dengan repositori data on-premise

video terkait:

- Optimalisasi biaya penyimpanan dengan Amazon EFS
- AWS re:Invent 2023 - Yang baru dengan penyimpanan file AWS
- AWS re:Invent 2023 - Penyimpanan file untuk builder dan ilmuwan data di Sistem File Elastis Amazon

SUS04-BP07 Meminimalkan perpindahan data di jaringan

Gunakan sistem file atau penyimpanan objek bersama untuk mengakses data umum dan meminimalkan total sumber daya jaringan yang diperlukan untuk mendukung perpindahan data beban kerja Anda.

Anti-pola umum:

- Anda menyimpan semua data di Wilayah AWS yang sama terlepas di mana pengguna data berada.
- Anda tidak mengoptimalkan format dan ukuran data sebelum memindahkannya melalui jaringan.

Manfaat menerapkan praktik terbaik ini: Mengoptimalkan perpindahan data di seluruh jaringan mengurangi total sumber daya jaringan yang diperlukan untuk beban kerja dan memperkecil dampaknya pada lingkungan.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Memindahkan data ke berbagai bagian dalam organisasi memerlukan sumber daya komputasi, jaringan, dan penyimpanan. Gunakan teknik untuk meminimalkan perpindahan data dan meningkatkan efisiensi beban kerja Anda secara keseluruhan.

Langkah-langkah implementasi

- Pertimbangkan kedekatan dengan data atau pengguna sebagai faktor dalam pengambilan keputusan saat memilih Wilayah untuk beban kerja Anda.
- Partisi layanan yang digunakan secara Regional sehingga data khusus Wilayahnya disimpan di Wilayah tempat penggunaan data.
- Gunakan format file yang efisien (seperti Parquet atau ORC) dan kompresi data sebelum memindahkannya melalui jaringan.
- Jangan pindahkan data yang tidak digunakan. Beberapa contoh tindakan yang dapat membantu Anda menghindari pemindahan data yang tidak digunakan:
  - Kurangi respons API untuk data yang relevan saja.
  - Kumpulkan data apabila terperinci (informasi tingkat catatan tidak diperlukan).
  - Lihat Lab Well-Architected - Mengoptimalkan Pola Data Menggunakan Fitur Berbagi Data Amazon Redshift.
  - Pertimbangkan Berbagi data lintas akun di AWS Lake Formation.
- Gunakan layanan yang dapat membantu Anda menjalankan kode lebih dekat dengan pengguna beban kerja Anda.

Layanan	Kapan harus digunakan
Lambda@Edge	Gunakan untuk operasi dengan banyak komputasi yang dijalankan saat objek tidak ada dalam cache.

Layanan	Kapan harus digunakan
Fungsi CloudFront	Gunakan untuk kasus penggunaan sederhana seperti permintaan HTTP/manipulasi respons yang dapat dimulai oleh fungsi dengan masa pakai singkat.
AWS IoT Greengrass	Jalankan komputasi lokal, olahpesan, dan caching data untuk perangkat yang terhubung.

Sumber daya

Dokumen terkait:

- Mengoptimalkan Infrastruktur AWS Anda untuk Keberlanjutan, Bagian III: Jaringan
- Infrastruktur Global AWS
- Fitur Utama Amazon CloudFront termasuk CloudFront Global Edge Network
- Mengompresi permintaan HTTP di Amazon OpenSearch Service
- Kompresi data menengah dengan Amazon EMR
- Memuat file data terkompresi dari Amazon S3 ke Amazon Redshift
- Melayani file terkompresi dengan Amazon CloudFront

Video terkait:

- Menjelaskan transfer data di AWS

Contoh terkait:

- Merancang arsitektur untuk keberlanjutan - Meminimalkan pergerakan data lintas jaringan

SUS04-BP08 Hanya mencadangkan data saat sulit untuk dibuat ulang

Hindari mencadangkan data yang tidak memiliki nilai bisnis untuk meminimalkan persyaratan sumber daya penyimpanan untuk beban kerja Anda.

Anti-pola umum:

- Anda tidak memiliki strategi cadangan untuk data Anda.
- Anda mencadangkan data yang dapat dibuat ulang dengan mudah.

Manfaat menerapkan praktik terbaik ini: Menghindari cadangan data non-kritis dapat mengurangi sumber daya penyimpanan yang diperlukan untuk beban kerja dan menurunkan dampak lingkungan yang ditimbulkannya.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Menghindari pencadangan data yang tidak perlu dapat membantu menurunkan biaya dan mengurangi sumber daya penyimpanan yang digunakan oleh beban kerja. Hanya cadangkan data yang memiliki nilai bisnis atau yang diperlukan untuk memenuhi persyaratan kepatuhan. Periksa kebijakan pencadangan dan jangan sertakan penyimpanan sementara yang tidak memberikan nilai dalam skenario pemulihan.

Langkah-langkah implementasi

- Menerapkan kebijakan klasifikasi data sebagaimana diuraikan dalam SUS04-BP01 Mengimplementasikan kebijakan klasifikasi data.
- Gunakan kekritisan klasifikasi data Anda dan rancang strategi cadangan berdasarkan sasaran waktu pemulihan (RTO) dan sasaran titik pemulihan (RPO) Anda. Hindari mencadangkan data yang tidak penting.
  - Jangan sertakan data yang dapat dibuat ulang dengan mudah.
  - Jangan sertakan data sementara dari cadangan Anda.
  - Jangan sertakan salinan lokal data, kecuali apabila waktu yang diperlukan untuk memulihkan data tersebut dari lokasi umum melebihi perjanjian tingkat layanan (SLA) Anda.
- Gunakan solusi otomatis atau layanan terkelola untuk mencadangkan data yang penting bagi bisnis.
  - AWS Backup adalah sebuah layanan pencadangan terkelola sepenuhnya yang Akan memudahkan dalam melakukan pemusatan dan otomatisasi pencadangan data di seluruh layanan AWS di cloud dan on-premise. Untuk panduan langsung tentang cara membuat cadangan otomatis dengan menggunakan AWS Backup, lihat Lab Well-Architected - Menguji Cadangan dan Penyimpanan Kembali Data.
  - Otomatiskan pencadangan dan optimalkan biaya pencadangan untuk Amazon EFS dengan menggunakan AWS Backup.

Sumber daya

Praktik-praktik terbaik terkait:

- REL09-BP01 Mengidentifikasi dan mencadangkan data yang perlu dicadangkan, atau melakukan reproduksi ulang data dari sumber
- REL09-BP03 Melakukan pencadangan data secara otomatis
- REL13-BP02 Menggunakan strategi pemulihan untuk memenuhi sasaran pemulihan

Dokumen terkait:

- Menggunakan AWS Backup untuk mencadangkan dan memulihkan sistem file Amazon EFS
- Snapshot Amazon EBS
- Bekerja dengan backup di Amazon Relational Database Service
- Partner APN: partner yang dapat membantu terkait pencadangan
- AWS Marketplace: produk yang dapat digunakan untuk pencadangan
- Mencadangkan Amazon EFS
- Mencadangkan Amazon FSx for Windows File Server
- Pencadangan dan pemulihan untuk Amazon ElastiCache (Redis OSS)

Video terkait:

- AWS re:Invent 2023 - Strategi backup dan pemulihan bencana untuk meningkatkan ketahanan
- AWS re:Invent 2023 - Yang baru dengan AWS Backup
- AWS re:Invent 2021 - Pencadangan, pemulihan bencana, dan perlindungan ransomware dengan AWS

Contoh terkait:

- Lab Well-Architected - Data cadangan

Perangkat keras dan layanan

Cari peluang untuk mengurangi dampak beban kerja terhadap keberlanjutan dengan membuat perubahan pada praktik manajemen perangkat keras Anda. Minimalkan jumlah perangkat keras yang

perlu disediakan dan di-deploy, serta pilih perangkat keras dan layanan yang paling efisien untuk setiap beban kerja Anda.

Praktik terbaik

- SUS05-BP01 Menggunakan perangkat keras dalam jumlah minimum untuk memenuhi kebutuhan Anda
- SUS05-BP02 Menggunakan jenis instans dengan dampak paling sedikit
- SUS05-BP03 Menggunakan layanan terkelola
- SUS05-BP04 Mengoptimalkan penggunaan akselerator komputasi berbasis perangkat keras

SUS05-BP01 Menggunakan perangkat keras dalam jumlah minimum untuk memenuhi kebutuhan Anda

Gunakan perangkat keras dalam jumlah minimum untuk beban kerja Anda guna memenuhi kebutuhan-kebutuhan bisnis secara efisien.

Anti-pola umum:

- Anda tidak memantau pemanfaatan sumber daya.
- Anda memiliki sumber daya dengan tingkat pemanfaatan rendah di arsitektur Anda.
- Anda tidak meninjau pemanfaatan perangkat keras statis untuk menentukan apakah ukurannya harus diubah atau tidak.
- Anda tidak menetapkan target pemanfaatan perangkat keras untuk infrastruktur komputasi Anda berdasarkan KPI bisnis.

Manfaat menerapkan praktik terbaik ini: Mengatur ukuran sumber daya cloud Anda dengan tepat akan membantu mengurangi dampak lingkungan beban kerja, menghemat uang, dan mempertahankan tolok ukur kinerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Pilih secara optimal jumlah total perangkat keras yang diperlukan untuk beban kerja Anda guna meningkatkan efisiensi beban kerja secara keseluruhan. AWS Cloud menyediakan fleksibilitas untuk

memperluas atau mengurangi jumlah sumber daya secara dinamis melalui beragam mekanisme, seperti AWS Auto Scaling, dan memenuhi perubahan sesuai permintaan. Ini juga menyediakan API dan SDK yang memungkinkan sumber daya dimodifikasi hanya dengan sedikit usaha. Gunakan kemampuan ini untuk membuat perubahan pada implementasi beban kerja Anda dengan interval waktu yang rapat. Selain itu, gunakan panduan penyesuaian ukuran dari alat AWS untuk secara efisien mengoperasikan sumber daya cloud Anda dan memenuhi kebutuhan bisnis Anda.

Langkah-langkah implementasi

- Pilih jenis instans: Pilih jenis instans yang tepat yang paling sesuai dengan kebutuhan Anda. Untuk mempelajari cara memilih instans Amazon Elastic Compute Cloud dan menggunakan mekanisme seperti pemilihan instans berbasis atribut, lihat dokumentasi berikut ini:
  - Bagaimana cara memilih jenis instans Amazon EC2 yang tepat untuk beban kerja saya?
  - Pemilihan jenis instans berbasis atribut untuk Amazon EC2 Fleet.
  - Membuat grup Auto Scaling menggunakan pemilihan jenis instans berbasis atribut.
- Skalakan: Gunakan peningkatan kecil untuk menskalakan beban kerja variabel.
- Gunakan beberapa opsi pembelian komputasi: Seimbangkan fleksibilitas instans, skalabilitas, dan penghematan biayanya dengan menggunakan beberapa opsi pembelian komputasi.
  - Instans Sesuai Permintaan Amazon EC2 paling cocok untuk beban kerja baru, stateful, dan berfluktuasi yang tidak dapat berupa jenis instans, lokasi, atau waktu yang fleksibel.
  - Instans Spot Amazon EC2 adalah cara yang bagus untuk melengkapi opsi-opsi lain untuk aplikasi yang toleran terhadap kesalahan dan fleksibel.
  - Manfaatkan Compute Savings Plans untuk beban kerja steady state yang memungkinkan fleksibilitas jika kebutuhan Anda (seperti AZ, Region, keluarga instans, atau jenis instans) berubah.
- Gunakan keragaman instans dan Zona Ketersediaan: Maksimalkan ketersediaan aplikasi dan manfaatkan kelebihan kapasitas dengan mendiversifikasi instans dan Zona Ketersediaan Anda.
- Instans pengatur ukuran yang tepat: Gunakan rekomendasi penyesuaian ukuran yang tepat dari alat AWS untuk melakukan penyesuaian pada beban kerja Anda. Untuk informasi selengkapnya, lihat Mengoptimalkan biaya Anda dengan Rekomendasi Penentuan Ukuran yang Tepat dan Penentuan Ukuran yang Tepat: Menyediakan Instans yang Sesuai dengan Beban Kerja
  - Gunakan rekomendasi pengatur ukuran yang tepat dalam AWS Cost Explorer atau AWS Compute Optimizer untuk mengidentifikasi peluang pengaturan ukuran yang tepat.
- Negosiasikan perjanjian tingkat layanan (SLA): Negosiasikan SLA yang mengizinkan pengurangan kapasitas sementara di saat otomatisasi melakukan deployment sumber daya pengganti.

Sumber daya

Dokumen terkait:

- Mengoptimalkan Infrastruktur AWS untuk Keberlanjutan, Bagian I: Komputasi
- Pemilihan Jenis Instans berbasis Atribut untuk Penskalaan Otomatis untuk Amazon EC2 Fleet
- Dokumentasi AWS Compute Optimizer
- Mengoperasikan: Optimasi kinerja
- Dokumentasi Penskalaan Otomatis

Video terkait:

- AWS re:Invent 2023 - Yang baru dengan Amazon EC2
- AWS re:Invent 2023 - Penghematan cerdas: Strategi optimalisasi Amazon Elastic Compute Cloud
- AWS re:Invent 2022 - Mengoptimalkan Amazon Elastic Kubernetes Service untuk kinerja dan biaya di AWS
- AWS re:Invent 2023 - Komputasi berkelanjutan: mengurangi biaya dan emisi karbon dengan AWS

SUS05-BP02 Menggunakan jenis instans dengan dampak paling sedikit

Terus pantau dan gunakan jenis instans baru untuk memanfaatkan peningkatan penghematan energi.

Anti-pola umum:

- Anda hanya menggunakan satu keluarga instans.
- Anda hanya menggunakan instans x86.
- Anda menentukan satu jenis instans dalam konfigurasi Amazon EC2 Auto Scaling Anda.
- Anda menggunakan instans AWS dengan cara yang tidak dirancang untuk instans tersebut (misalnya, Anda menggunakan instans komputasi yang dioptimalkan untuk beban kerja yang intensif memori).
- Anda tidak mengevaluasi jenis instans baru secara teratur.
- Anda tidak melihat rekomendasi dari alat pengatur ukuran yang tepat AWS seperti AWS Compute Optimizer.

Manfaat menerapkan praktik terbaik ini: Dengan memanfaatkan instans hemat energi dan berukuran tepat, Anda dapat jauh mengurangi dampak lingkungan dan biaya beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Menggunakan instans yang efisien dalam beban kerja cloud sangat penting untuk menurunkan penggunaan sumber daya dan menghemat biaya. Terus lakukan pemantauan terhadap rilis instans jenis baru dan manfaatkan peningkatan penghematan energi, termasuk jenis instans yang dirancang untuk mendukung beban kerja spesifik seperti pelatihan dan inferensi machine learning, serta transkode video.

Langkah-langkah implementasi

- Pelajari dan jelajahi jenis instans: Temukan jenis instans yang dapat menurunkan dampak lingkungan beban kerja Anda.
  - Berlangganan Yang Baru AWS untuk tetap mendapatkan informasi terbaru tentang teknologi AWS dan instans terbaru.
  - Pelajari tentang jenis instans AWS yang berbeda-beda.
  - Pelajari tentang instans berbasis AWS Graviton yang menawarkan kinerja terbaik per watt penggunaan energi di Amazon EC2 dengan menonton re:Invent 2020 - Memahami lebih dalam tentang instans Amazon EC2 yang ditenagai prosesor AWS Graviton2 dan Memahami lebih dalam tentang instans AWS Graviton3 dan Amazon EC2 C7g.
- Gunakan jenis instans dengan dampak paling kecil: Rencanakan dan transisikan beban kerja Anda ke jenis instans dengan dampak paling kecil.
  - Tentukan sebuah proses untuk mengevaluasi fitur atau instans baru untuk beban kerja Anda. Manfaatkan ketangkasan di cloud untuk menguji dengan cepat bagaimana jenis instans baru dapat meningkatkan keberlanjutan beban kerja Anda. Gunakan metrik-metrik proksi untuk mengukur berapa banyak sumber daya yang Anda perlukan untuk menyelesaikan satu unit pekerjaan.
  - Jika memungkinkan, ubah beban kerja agar bisa menggunakan jumlah vCPU yang berbeda-beda dan jumlah memori yang berbeda-beda guna memaksimalkan pilihan jenis instans.
  - Pertimbangkan untuk mengalihkan beban kerja Anda ke instans berbasis Graviton guna meningkatkan efisiensi performa beban kerja Anda. Untuk informasi selengkapnya tentang memindahkan beban kerja ke AWS Graviton, lihat Mulai Cepat AWS Graviton dan Pertimbangan

saat mentransisikan beban kerja ke instans Amazon Elastic Compute Cloud berbasis AWS Graviton.

- Pertimbangkan untuk memilih opsi AWS Graviton dalam penggunaan layanan terkelola AWS.
- Migrasikan beban kerja ke Wilayah yang menawarkan instans dengan dampak paling sedikit terhadap keberlanjutan dan masih memenuhi kebutuhan bisnis Anda.
- Untuk beban kerja machine learning, manfaatkan perangkat keras yang dibuat khusus untuk beban kerja Anda, seperti AWS Trainium, AWS Inferentia, dan Amazon EC2 DL1. AWS Instans-instans Inferentia seperti instans Inf2 menawarkan kinerja per watt hingga 50% lebih baik daripada instans Amazon EC2 yang setara.
- Gunakan Amazon SageMaker Inference Recommender untuk menentukan titik akhir inferensi ML ukuran yang tepat.
- Untuk beban kerja yang berfluktuasi (beban kerja yang jarang memerlukan kapasitas tambahan), gunakan instans performa yang dapat melonjak.
- Gunakan instans Spot Amazon EC2 untuk beban kerja yang toleran terhadap kesalahan dan stateless guna meningkatkan pemanfaatan cloud secara keseluruhan, serta mengurangi dampak terhadap keberlanjutan dari sumber daya yang tidak digunakan.

- Operasikan dan optimalkan: Operasikan dan optimalkan instans beban kerja Anda.
  - Untuk beban kerja sementara, lakukan evaluasi terhadap metrik instans Amazon CloudWatch seperti `CPUUtilization` untuk mengidentifikasi apakah instans tersebut tidak digunakan atau kurang begitu dimanfaatkan.
  - Untuk beban kerja stabil, periksa alat penyesuaian ukuran AWS seperti AWS Compute Optimizer secara berkala untuk mengidentifikasi adanya peluang melakukan optimalisasi dan penyesuaian ukuran sumber daya komputasi dengan tepat. Untuk contoh dan rekomendasi lebih lanjut, silakan lihat lab berikut:
    - Lab Well-Architected - Rekomendasi Penyesuaian Ukuran
    - Lab Well-Architected - Penyesuaian Ukuran dengan Pengoptimal Komputasi
    - Lab Well-Architected - Optimisasi Pola Perangkat Keras dan Pengamatan KPI Keberlanjutan

Sumber daya

Dokumen terkait:

- Mengoptimalkan Infrastruktur AWS untuk Keberlanjutan, Bagian I: Komputasi
- AWS Graviton

- Amazon EC2 DL1
- Armada Reservasi Kapasitas Amazon EC2
- Armada Spot Amazon EC2
- Fungsi: Konfigurasi Fungsi Lambda
- Pemilihan jenis instans berbasis atribut untuk Amazon EC2 Fleet
- Membangun Aplikasi yang Berkelanjutan, Efisien, dan Dioptimalkan untuk Biaya di AWS
- Bagaimana Dasbor Keberlanjutan Contino Membantu Pelanggan Mengoptimalkan Jejak Karbon Mereka

Video terkait:

- AWS re:Invent 2023 - AWS Graviton: Performa harga terbaik untuk beban kerja AWS Anda
- AWS re:Invent 2023 - Kemampuan AI generatif Amazon Elastic Compute Cloud baru di AWS Management Console
- AWS re:Invent 2023 = Yang baru dengan Amazon Elastic Compute Cloud
- AWS re:Invent 2023 - Penghematan cerdas: Strategi optimalisasi Amazon Elastic Compute Cloud
- AWS re:Invent 2021 - Memahami Lebih Dalam tentang instans AWS Graviton3 dan Amazon EC2 C7g
- AWS re:Invent 2022 - Membangun lingkungan komputasi yang hemat biaya, energi, dan sumber daya

Contoh terkait:

- Solusi: Panduan untuk Mengoptimalkan Beban Kerja Deep Learning untuk Keberlanjutan di AWS
- Migrasi Basis Data Amazon Relational Database Service ke Graviton

SUS05-BP03 Menggunakan layanan terkelola

Gunakan layanan-layanan terkelola untuk beroperasi dengan lebih efisien di cloud.

Anti-pola umum:

- Anda menggunakan instans Amazon EC2 dengan pemanfaatan rendah untuk menjalankan aplikasi Anda.

- Tim internal Anda hanya mengelola beban kerja, tanpa ada waktu untuk berfokus melakukan inovasi atau simplifikasi.
- Anda melakukan deployment dan memelihara teknologi untuk tugas-tugas yang dapat dijalankan dengan lebih efisien di layanan-layanan terkelola.

Manfaat menjalankan praktik terbaik ini:

- Menggunakan layanan terkelola akan mengalihkan tanggung jawab ke AWS, yang memiliki wawasan atas jutaan pelanggan yang dapat membantu Anda mendorong efisiensi dan inovasi baru.
- Layanan terkelola mendistribusikan dampak lingkungan dari layanan ke banyak pengguna karena bidang kontrol multi-prinsip.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Layanan terkelola mengalihkan tanggung jawab ke AWS untuk mempertahankan pemanfaatan tinggi dan optimisasi keberlanjutan dari perangkat keras yang di-deploy. Layanan-layanan terkelola juga dapat menghilangkan beban administratif dan operasional yang muncul dalam pemeliharaan layanan, sehingga tim Anda dapat memiliki lebih banyak waktu dan berfokus untuk membuat inovasi.

Tinjau beban kerja Anda untuk mengidentifikasi komponen-komponen yang dapat digantikan oleh layanan-layanan terkelola AWS. Sebagai contoh, Amazon RDS, Amazon Redshift, dan Amazon ElastiCache menyediakan layanan basis data terkelola. Amazon Athena, Amazon EMR, dan Amazon OpenSearch Service menyediakan layanan analitik terkelola.

Langkah-langkah implementasi

1. Buat inventarisasi beban kerja Anda: Inventarisasikan beban kerja Anda untuk layanan dan komponen.
2. Identifikasi kandidat: Nilai dan identifikasi komponen yang dapat digantikan oleh layanan terkelola. Berikut ini adalah beberapa contoh kapan Anda mungkin harus mempertimbangkan untuk menggunakan layanan terkelola:

Tugas	Apa yang harus digunakan pada AWS
Meng-hosting basis data	Gunakan instans Amazon Relational Database Service (Amazon RDS) terkelola , alih-alih mempertahankan instans Amazon RDS Anda sendiri di Amazon Elastic Compute Cloud (Amazon EC2).
Hosting beban kerja kontainer	Gunakan AWS Fargate, alih-alih mengimple mentasikan infrastruktur kontainer Anda sendiri.
Hosting aplikasi web	Gunakan AWS Amplify Hosting sebagai CI/ CD dan layanan hosting terkelola sepenuhny a untuk situs web statis dan aplikasi web yang dirender sisi server.

3. Buat rencana migrasi: Identifikasi dependensi dan buatlah sebuah rencana migrasi. Perbarui runbook dan playbook sesuai dengannya.
    - AWS Application Discovery Service secara otomatis mengumpulkan dan menyajikan informasi terperinci tentang dependensi dan pemanfaatan aplikasi untuk membantu Anda membuat keputusan yang lebih tepat saat merencanakan migrasi Anda
4. Lakukan pengujian Uji layanan sebelum migrasi ke layanan terkelola.
5. Ganti layanan yang dihosting sendiri: Gunakan paket migrasi Anda untuk mengganti layanan-layanan yang dihosting sendiri dengan layanan terkelola.
6. Pantau dan sesuaikan: Terus pantau layanan setelah migrasi selesai untuk membuat penyesuaian sebagaimana diperlukan dan optimalkan layanan.

Sumber daya

Dokumen terkait:

- Produk AWS Cloud
- Kalkulator Total Biaya Kepemilikan (TCO) AWS
- Amazon DocumentDB

- Amazon Elastic Kubernetes Service (EKS)
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Video terkait:

- AWS re:Invent 2021 - Operasi cloud dalam skala besar dengan AWS Managed Services
- AWS re:Invent 2023 - Praktik terbaik untuk beroperasi di AWS

SUS05-BP04 Mengoptimalkan penggunaan akselerator komputasi berbasis perangkat keras

Optimalkan penggunaan instans komputasi terakselerasi Anda untuk mengurangi permintaan-permintaan infrastruktur fisik beban kerja Anda.

Anti-pola umum:

- Anda tidak memantau penggunaan GPU.
- Anda menggunakan instans tujuan umum untuk beban kerja, padahal instans yang dibuat khusus dapat menghadirkan kinerja yang lebih tinggi, biaya yang lebih rendah, dan kinerja per watt yang lebih baik.
- Anda menggunakan akselerator komputasi berbasis perangkat keras untuk tugas-tugas yang akan lebih efisien jika dikerjakan menggunakan alternatif berbasis CPU.

Manfaat menerapkan praktik terbaik ini: Dengan mengoptimalkan penggunaan akselerator berbasis perangkat keras, Anda dapat mengurangi permintaan infrastruktur fisik untuk beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Jika Anda memerlukan kemampuan pemrosesan tinggi, Anda dapat memanfaatkan instans komputasi terakselerasi, yang menyediakan akses ke akselerator komputasi berbasis perangkat keras seperti unit pemrosesan grafis (GPU) dan field programmable gate array (FPGA). Akselerator perangkat keras ini menjalankan fungsi-fungsi tertentu seperti pemrosesan grafis atau pencocokan pola data secara lebih efisien daripada alternatif yang berbasis CPU. Banyak beban kerja terakselerasi, seperti perenderan, transkode, dan machine learning, yang memiliki variabel tinggi

sehubungan dengan penggunaan sumber daya. Jalankan perangkat keras ini hanya ketika diperlukan, dan non-aktifkan instans GPU secara otomatis saat tidak diperlukan, guna meminimalkan sumber daya yang digunakan.

Langkah-langkah implementasi

- Identifikasi instans komputasi terakselerasi yang dapat memenuhi kebutuhan Anda.
- Untuk beban kerja machine learning, manfaatkan perangkat keras yang dibuat khusus untuk beban kerja Anda, misalnya AWS Trainium, AWS Inferentia, dan Amazon EC2 DL1. AWS Instans-instans Inferentia seperti instans Inf2 menawarkan kinerja per watt hingga 50% lebih baik daripada instans Amazon EC2 yang setara.
- Kumpulkan metrik penggunaan untuk instans komputasi terakselerasi Anda. Misalnya, Anda dapat menggunakan agen CloudWatch untuk mengumpulkan metrik seperti metrik `utilization_gpu`dan `utilization_memory` untuk GPU Anda seperti yang ditunjukkan dalam Mengumpulkan metrik GPU NVIDIA dengan Amazon CloudWatch.
- Optimalkan kode, operasi jaringan, dan pengaturan akselerator perangkat keras untuk memastikan perangkat keras yang mendasarinya dimanfaatkan sepenuhnya.
  - Mengoptimalkan pengaturan GPU
  - Pemantauan dan Pengoptimalan GPU dalam AMI Deep Learning
  - Mengoptimalkan I/O untuk penyetelan kinerja GPU pelatihan deep learning di Amazon SageMaker
- Gunakan driver GPU dan pustaka berkinerja tinggi terbaru.
- Gunakan otomatisasi untuk melepaskan instans GPU ketika tidak digunakan.

Sumber daya

Dokumen terkait:

- Komputasi yang Dipercepat
- Mari Merancang! Merancang arsitektur dengan chip dan akselerator kustom
- Bagaimana cara memilih jenis instans Amazon EC2 yang tepat untuk beban kerja saya?
- Instans Amazon EC2 VT1
- Pilih akselerator AI dan kompilasi model terbaik untuk inferensi penglihatan komputer dengan Amazon SageMaker

Video terkait:

- AWS re:Invent 2021 - Cara memilih instans GPU Amazon EC2 untuk deep learning
- Pembicaraan Teknologi Online AWS - Men-deploy Inferensi Deep Learning yang Hemat Biaya
- AWS re:Invent 2023 - AI mutakhir dengan AWS dan NVIDIA
- AWS re:Invent 2022 - [PELUNCURAN BARU!] Memperkenalkan instans Amazon EC2 Inf2 berbasis AWS Inferensia2
- AWS re:Invent 2022 - Mempercepat deep learning dan berinovasi lebih cepat dengan AWS Trainium
- AWS re:Invent 2022 - Deep learning di AWS dengan NVIDIA: Dari pelatihan hingga deployment

Proses dan budaya

Cari peluang untuk mengurangi dampak operasi Anda terhadap keberlanjutan dengan membuat perubahan pada praktik deployment, pengujian, dan pengembangan.

Praktik terbaik

- SUS06-BP01 Mengadopsi metode yang dapat menghadirkan peningkatan keberlanjutan dengan cepat
- SUS06-BP02 Selalu pastikan beban kerja Anda mutakhir
- SUS06-BP03 Meningkatkan pemanfaatan lingkungan build
- SUS06-BP04 Menggunakan device farm terkelola untuk pengujian

SUS06-BP01 Mengadopsi metode yang dapat menghadirkan peningkatan keberlanjutan dengan cepat

Adopsi metode dan proses untuk memvalidasi potensi peningkatan, meminimalkan biaya pengujian, dan memberikan peningkatan-peningkatan kecil.

Anti-pola umum:

- Meninjau aplikasi Anda untuk keberlanjutan adalah tugas yang dilakukan hanya satu kali pada awal proyek.
- Beban kerja Anda telah kedaluwarsa, karena proses rilis terlalu merepotkan guna melakukan perubahan kecil untuk efisiensi sumber daya.

- Anda tidak memiliki mekanisme untuk meningkatkan beban kerja untuk keberlanjutan.

Manfaat menerapkan praktik terbaik ini: Dengan membangun proses untuk memperkenalkan dan melacak peningkatan keberlanjutan, Anda akan dapat terus mengadopsi fitur dan kemampuan baru, menghilangkan masalah, dan meningkatkan efisiensi beban kerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Sedang

Panduan implementasi

Uji dan validasi potensi peningkatan keberlanjutan sebelum melakukan deployment peningkatan ini ke produksi. Pertimbangkan biaya pengujian saat menghitung potensi manfaat dari sebuah peningkatan di masa mendatang. Kembangkan metode-metode pengujian berbiaya rendah untuk memberikan peningkatan kecil.

Langkah-langkah implementasi

- Memahami dan mengkomunikasikan tujuan keberlanjutan organisasi Anda: Memahami tujuan keberlanjutan organisasi Anda, seperti pengurangan karbon atau pengelolaan air. Terjemahkan tujuan tersebut menjadi persyaratan keberlanjutan untuk beban kerja cloud Anda. Komunikasikan persyaratan-persyaratan tersebut kepada para pemangku kepentingan utama.
- Tambahkan persyaratan keberlanjutan ke backlog Anda: Tambahkan persyaratan untuk melakukan peningkatan keberlanjutan ke backlog pengembangan Anda.
- Ulang-ulang dan tingkatkan: Gunakan proses perbaikan berulang untuk mengidentifikasi, mengevaluasi, memprioritaskan, menguji, dan men-deploy peningkatan ini.
- Uji menggunakan produk layak minimum (MVP): Mengembangkan dan menguji potensi peningkatan menggunakan komponen perwakilan minimum yang layak untuk mengurangi biaya dan dampak lingkungan dari pengujian.
- Sederhanakan prosesnya: Terus tingkatkan dan sederhanakan proses pengembangan Anda. Sebagai contoh, Lakukan otomastisasi terhadap proses pengiriman perangkat lunak Anda menggunakan pipeline integrasi dan pengiriman berkelanjutan (CI/CD) untuk menguji dan melakukan deployment pada peningkatan-peningkatan potensial untuk mengurangi tingkat upaya dan membatasi kesalahan yang disebabkan oleh proses manual.
- Pelatihan dan kesadaran: Jalankan program pelatihan untuk para anggota tim Anda untuk mendidik mereka tentang keberlanjutan dan bagaimana aktivitas mereka memengaruhi tujuan keberlanjutan organisasi Anda.
- Nilai dan sesuaikan: Terus nilai dampak peningkatan dan buat penyesuaian jika diperlukan.

Sumber daya

Dokumen terkait:

- AWS menghadirkan solusi keberlanjutan
- Praktik pengembangan tangkas yang dapat diskalakan berdasarkan AWS CodeCommit

Video terkait:

- AWS re:Invent 2023 - Arsitektur berkelanjutan: Masa lalu, sekarang, dan masa depan
- AWS re:Invent 2022 - Menghadirkan arsitektur berkelanjutan dan berkinerja tinggi
- AWS re:Invent 2022 - Merancang arsitektur secara berkelanjutan dan mengurangi jejak karbon AWS Anda
- AWS re:Invent 2022 - Keberlanjutan dalam infrastruktur global AWS
- AWS re:Invent 2023 - Apa yang baru dengan observabilitas dan operasi AWS

Contoh terkait:

- Lab Well-Architected - Mengubah laporan biaya & penggunaan menjadi laporan efisiensi

SUS06-BP02 Selalu pastikan beban kerja Anda mutakhir

Selalu pastikan beban kerja Anda mutakhir untuk mengadopsi fitur yang efisien, menghilangkan masalah, dan meningkatkan efisiensi beban kerja Anda secara keseluruhan.

Anti-pola umum:

- Anda berasumsi bahwa arsitektur Anda saat ini adalah arsitektur statis dan tidak akan diperbarui seiring waktu.
- Anda tidak memiliki sistem atau koordinasi rutin untuk mengevaluasi apakah perangkat lunak dan paket-paket yang diperbarui kompatibel dengan beban kerja Anda.

Manfaat menerapkan praktik terbaik ini: Dengan menetapkan proses untuk menjaga agar beban kerja Anda tetap yang terbaru, Anda akan dapat menerapkan fitur dan kemampuan baru, menyelesaikan masalah, dan meningkatkan efisiensi beban kerja.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Sistem operasi, runtime, perangkat lunak perantara (middleware), pustaka, dan aplikasi yang mutakhir dapat meningkatkan efisiensi beban kerja serta memudahkan Anda melakukan adopsi teknologi yang lebih efisien. Perangkat lunak yang mutakhir juga dapat menyertakan fitur-fitur untuk mengukur dampak beban kerja terhadap keberlanjutan secara lebih akurat, mengingat vendor juga menghadirkan fitur-fitur untuk memenuhi tujuan keberlanjutan mereka sendiri. Secara teratur jaga agar beban kerja Anda diperbarui dengan fitur dan rilis terbaru.

Langkah-langkah implementasi

- Tentukan sebuah proses: Gunakan sebuah proses dan jadwal untuk mengevaluasi fitur atau instans baru untuk beban kerja Anda. Manfaatkan ketangkasan di cloud untuk menguji dengan cepat bagaimana fitur-fitur baru dapat meningkatkan beban kerja Anda untuk:
  - Mengurangi dampak keberlanjutan.
  - Memperoleh efisiensi performa.
  - Menghilangkan penghalang untuk peningkatan terencana.
  - Meningkatkan kemampuan Anda dalam mengukur dan mengelola dampak terhadap keberlanjutan.
- Buat inventaris beban kerja: Buatlah inventaris perangkat lunak dan arsitektur beban kerja Anda dan identifikasi komponen yang perlu diperbarui.
  - Anda dapat menggunakan AWS Systems Manager Inventory untuk mengumpulkan metadata sistem operasi (OS), aplikasi, dan metadata instans dari instans Amazon EC2 dan secara cepat memahami instans mana yang menjalankan perangkat lunak dan konfigurasi yang diperlukan oleh kebijakan perangkat lunak Anda dan instans mana yang perlu diperbarui.
- Pelajari prosedur pembaruan: Pahami cara memperbarui komponen beban kerja Anda.

Komponen Beban Kerja	Cara memperbarui
Citra mesin	Gunakan EC2 Image Builder untuk mengelola pembaruan Amazon Machine Image (AMI) untuk image server Linux atau Windows.
Gambar kontainer	Gunakan Amazon Elastic Container Registry (Amazon ECR) dengan pipeline yang ada

Komponen Beban Kerja	Cara memperbarui
	sekarang untuk mengelola image Amazon Elastic Container Service (Amazon ECS).
AWS Lambda	AWS Lambda menyertakan fitur manajemen versi.

- Gunakan otomatisasi: Lakukan otomatisasi terhadap proses pembaruan guna mengurangi tingkat upaya dalam melakukan deployment fitur baru dan membatasi kesalahan yang disebabkan oleh proses manual.
  - Anda dapat menggunakan CI/CD untuk secara otomatis memperbarui AMI, image kontainer, dan artefak lain yang terkait dengan aplikasi cloud Anda.
  - Anda dapat menggunakan alat seperti AWS Systems Manager Patch Manager untuk melakukan otomatisasi proses pembaruan sistem, dan menjadwalkan aktivitas menggunakan AWS Systems Manager Windows Maintenance.

Sumber daya

Dokumen terkait:

- Pusat Arsitektur AWS
- Yang Baru dengan AWS
- Alat Pengembang AWS

Video terkait:

- AWS re:Invent 2022 - Optimalkan beban kerja AWS Anda dengan panduan praktik terbaik
- All Things Patch: AWS Systems Manager

Contoh terkait:

- Lab Well-Architected - Manajemen Inventaris dan Patch
- Lab: AWS Systems Manager

SUS06-BP03 Meningkatkan pemanfaatan lingkungan build

Tingkatkan pemanfaatan sumber daya untuk mengembangkan, menguji, dan membangun beban kerja Anda.

Anti-pola umum:

- Anda menyediakan atau menghentikan lingkungan build Anda secara manual.
- Anda mempertahankan lingkungan-lingkungan build terus berjalan terlepas dari aktivitas pengujian, build, atau rilis (misalnya, menjalankan lingkungan di luar jam kerja anggota tim pengembangan Anda).
- Anda menyediakan terlalu banyak sumber daya untuk lingkungan build Anda.

Manfaat menerapkan praktik terbaik ini: Dengan meningkatkan pemanfaatan lingkungan build, Anda dapat meningkatkan efisiensi keseluruhan beban kerja cloud Anda sekaligus mengalokasikan sumber daya untuk kepada builder untuk melakukan pengembangan, pengujian, dan pembangunan secara efisien.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Gunakan otomatisasi dan infrastruktur sebagai kode untuk mengaktifkan lingkungan build saat diperlukan dan menonaktifkannya saat tidak digunakan. Hal yang umum dilakukan adalah menjadwalkan periode ketersediaan yang bertepatan dengan jam kerja anggota tim pengembangan. Lingkungan pengujian Anda harus sangat mirip dengan konfigurasi lingkungan produksi. Tetapi, cari peluang untuk menggunakan jenis instans dengan kapasitas lonjakan, Instans Spot Amazon EC2, layanan basis data penskalaan otomatis, kontainer, dan teknologi nirserver untuk menyesuaikan pengembangan dan menguji kapasitas dengan penggunaan. Batasi volume data untuk tepat memenuhi persyaratan pengujian. Jika Anda menggunakan data produksi dalam pengujian, jelajahi kemungkinan berbagi data dari produksi dan tidak memindahkan data ke mana-mana.

Langkah-langkah implementasi

- Menggunakan infrastruktur sebagai kode: Gunakan infrastruktur sebagai kode untuk menyediakan lingkungan build Anda.
- Gunakan otomatisasi: Gunakan otomatisasi untuk mengelola siklus hidup pengembangan dan menguji lingkungan serta memaksimalkan efisiensi sumber daya build Anda.

- Maksimalkan pemanfaatan: Gunakan strategi untuk memaksimalkan pemanfaatan lingkungan pengembangan dan pengujian.
  - Gunakan lingkungan representatif yang dapat digunakan pada tingkat minimum untuk mengembangkan dan menguji potensi peningkatan yang mungkin dilakukan.
  - Gunakan teknologi nirserver jika mungkin.
  - Gunakan Instans Sesuai Permintaan untuk membantu perangkat-perangkat pengembang Anda.
  - Gunakan jenis instans dengan kapasitas lonjakan, Instans Spot, dan teknologi-teknologi lainnya untuk menyesuaikan kapasitas build dengan penggunaan.
  - Adopsi layanan-layanan cloud native untuk akses shell instans yang aman daripada melakukan deployment armada host bastion.
  - Menskalakan secara otomatis sumber daya build Anda sesuai dengan tugas build Anda.

Sumber daya

Dokumen terkait:

- Manajer Sesi AWS Systems Manager
- Instans performa yang dapat melonjak Amazon EC2
- Apa itu AWS CloudFormation?
- Apa itu AWS CodeBuild?
- Penjadwal Instans di AWS

Video terkait:

- AWS re:Invent 2023 - Integrasi dan pengiriman berkelanjutan untuk AWS

SUS06-BP04 Menggunakan device farm terkelola untuk pengujian

Gunakan device farm terkelola untuk secara efisien menguji fitur baru pada serangkaian perangkat keras representatif.

Anti-pola umum:

- Anda menguji dan melakukan deployment terhadap aplikasi Anda di masing-masing perangkat fisik secara manual.

- Anda tidak menggunakan layanan pengujian aplikasi untuk menguji dan berinteraksi dengan aplikasi Anda (contohnya, Android, iOS, dan aplikasi web) pada perangkat fisik dan nyata.

Manfaat menerapkan praktik terbaik ini: Menggunakan device farm terkelola untuk menguji aplikasi berkemampuan cloud akan memberikan Anda sejumlah manfaat:

- Device farm terkelola disertai dengan fitur yang lebih efisien untuk menguji aplikasi di berbagai macam perangkat.
- Device farm terkelola menghilangkan kebutuhan akan infrastruktur internal untuk melakukan pengujian.
- Device farm terkelola menawarkan berbagai macam jenis perangkat, termasuk perangkat keras yang lebih lama dan kurang populer, yang menghilangkan kebutuhan akan pemutakhiran perangkat yang tidak perlu.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: Rendah

Panduan implementasi

Menggunakan device farm terkelola dapat membantu Anda untuk menyederhanakan proses pengujian untuk fitur baru di serangkaian perangkat keras representatif. Device farm terkelola menawarkan berbagai jenis perangkat, termasuk perangkat keras yang lebih lama dan kurang populer, serta menghindari dampak keberlanjutan pelanggan akibat pemutakhiran perangkat yang tidak perlu.

Langkah-langkah implementasi

- Tentukan persyaratan pengujian: Tetapkan persyaratan pengujian Anda dan rencanakan (seperti jenis pengujian, sistem operasi, dan jadwal pengujian).
  - Anda dapat menggunakan Amazon CloudWatch RUM untuk mengumpulkan dan menganalisis data sisi klien serta membentuk paket pengujian Anda.
- Pilih device farm terkelola: Pilih device farm terkelola yang dapat mendukung persyaratan pengujian Anda. Misalnya, Anda dapat menggunakan AWS Device Farm untuk menguji dan memahami dampak perubahan yang terjadi pada perangkat keras yang representatif.
- Gunakan otomatisasi: Gunakan otomatisasi dan integrasi berkelanjutan/deployment berkelanjutan (CI/CD) untuk menjadwalkan dan menjalankan pengujian Anda.
  - Mengintegrasikan AWS Device Farm dengan pipeline CI/CD Anda untuk menjalankan pengujian Selenium lintas-browser

  - Membangun dan menguji aplikasi iOS dan iPadOS dengan AWS DevOps dan layanan seluler
- Tinjau dan sesuaikan: Terus tinjau hasil pengujian Anda dan buat peningkatan yang perlu.

Sumber daya

Dokumen terkait:

- daftar perangkat AWS Device Farm
- Melihat dasbor CloudWatch RUM

Video terkait:

- AWS re:Invent 2023 - Tingkatkan kualitas aplikasi seluler dan web Anda dengan menggunakan Device Farm AWS
- AWS re:Invent 2021 - Optimalkan aplikasi melalui wawasan pengguna akhir dengan Amazon CloudWatch RUM

Contoh terkait:

- Aplikasi Sampel AWS Device Farm untuk Android
- Aplikasi Sampel AWS Device Farm untuk iOS
- Tes Appium Web untuk AWS Device Farm

Kesimpulan

Jumlah organisasi yang menetapkan target keberlanjutan kian bertambah. Hal ini disebabkan perubahan peraturan pemerintah, keuntungan kompetitif, dan permintaan investor, pegawai, serta pelanggan. CTO, arsitek, developer. dan anggota tim operasi tengah mencari cara agar mereka dapat terlibat secara langsung dalam mencapai tujuan keberlanjutan organisasi. Dengan menggunakan prinsip desain dan praktik terbaik yang didukung AWS ini, Anda dapat membuat keputusan yang matang terkait penyeimbangan keamanan, biaya, kinerja, keandalan, dan keunggulan operasional dengan hasil keberlanjutan untuk beban kerja AWS Cloud Anda. Setiap tindakan yang Anda ambil untuk meminimalkan penggunaan sumber daya dan meningkatkan efisiensi di seluruh beban kerja turut berkontribusi pada pengurangan dampak lingkungan dan berkontribusi pada tujuan keberlanjutan yang lebih luas milik organisasi Anda.

Kontributor

Para kontributor untuk dokumen ini antara lain:

- Sam Mokhtari, Senior Efficiency Lead Solutions Architect, Amazon Web Services
- Brendan Sisson, Arsitek Solusi Keberlanjutan Utama, Amazon Web Services
- Margaret O'Toole, Sustainability Tech Leader, Amazon Web Services
- Steffen Grunwald, Arsitek Solusi Keberlanjutan Utama, Amazon Web Services
- Ryan Eccles, Rekayasawan Utama, Keberlanjutan, Amazon
- Rodney Lester, Arsitek Utama, Amazon Web Services
- Adrian Cockcroft, VP Sustainability Architecture, Amazon Web Services
- Ian Meyers, Director of Technology, Solutions Architecture, Amazon Web Services

Sumber bacaan lebih lanjut

Untuk informasi tambahan, baca:

- AWS Well-Architected
- Pusat Arsitektur AWS
- Keberlanjutan di Cloud
- AWS menghadirkan solusi keberlanjutan
- Sumpah Iklim
- Tujuan Pembangunan Keberlanjutan Perserikatan Bangsa-Bangsa
- Protokol Gas Rumah Kaca

Revisi dokumen

Untuk mengetahui jika ada perubahan pada laporan resmi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Panduan praktik terbaik yang sudah diperbarui	Perubahan kecil throughput di seluruh pilar.	27 Juni 2024
Tingkat risiko yang diperbarui	Pembaruan kecil untuk tingkat risiko praktik terbaik.	3 Oktober 2023
Panduan praktik terbaik yang sudah diperbarui	Praktik terbaik diperbaru i dengan panduan baru di bidang-bidang berikut: Keselarasan dengan permintaan, Perangkat Lunak dan arsitektur, Data, dan Perangkat Keras dan layanan.	13 Juli 2023
Diperbarui untuk Kerangka Kerja baru	Praktik terbaik diperbarui dengan panduan preskript if dan praktik terbaik baru ditambahkan.	10 April 2023
Laporan resmi diperbarui	Praktik terbaik sudah diperbarui dengan panduan implementasi yang baru.	15 Desember 2022
Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan sudah ditambahkan.	20 Oktober 2022
Publikasi awal	Pilar Keberlanjutan - Kerangka Kerja AWS Well-Architected diterbitkan.	2 Desember 2021

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya disediakan sebagai informasi, (b) berisi AWS penawaran produk dan praktik saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak menjadi komitmen atau jaminan apa pun dari AWS dan afiliasi, pemasok, atau pemberi lisensinya. Produk atau layanan AWS diberikan “apa adanya” tanpa jaminan, pernyataan, atau syarat apa pun, baik secara tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

Daftar istilah AWS

Untuk terminologi AWS terbaru, lihat Daftar istilah AWS di Referensi Glosarium AWS.