Kerangka Kerja AWS Well-Architected

Pilar Efisiensi Kinerja

Pilar Efisiensi Kinerja: Kerangka Kerja AWS Well-Architected

Table of Contents

Pilar Efisiensi Kinerja - AWS Well-Architected Framework

Tanggal publikasi: 27 Juni 2024 (Revisi dokumen)

Abstrak

Laporan resmi ini berfokus pada pilar efisiensi kinerja Kerangka Kerja AWS Well-Architected. Ruang lingkup dokumen ini adalah untuk memberikan panduan yang membantu pelanggan menggunakan sumber daya cloud secara efisien untuk memenuhi kebutuhan bisnis mereka, dan mempertahankan efisiensi itu seiring perubahan permintaan dan perubahan teknologi.

Pengantar

Dengan Kerangka Kerja AWS Well-Architected , Anda dapat memahami pro dan kontra keputusan yang Anda ambil selama membangun beban kerja di AWS. Penggunaan Kerangka Kerja ini membantu Anda mempelajari praktik terbaik arsitektur untuk mendesain dan mengoperasikan beban kerja yang andal, aman, efisien, hemat biaya, dan ramah lingkungan di cloud. Kerangka Kerja ini menyediakan cara untuk secara terus menerus menilai arsitektur Anda berdasarkan praktik terbaik dan mengidentifikasi area yang perlu diperbaiki. Kami percaya bahwa memiliki beban kerja yang didesain dengan baik akan meningkatkan peluang keberhasilan bisnis.

Enam pilar landasan kerangka kerja:

- Keunggulan Operasional
- Keamanan
- Keandalan
- Efisiensi Kinerja
- Optimasi Biaya
- Pelestarian Lingkungan

Artikel ini berfokus pada penerapan prinsip pilar efisiensi kinerja pada beban kerja Anda. Pada lingkungan on-premise tradisional, meraih kinerja yang tinggi dan bertahan lama merupakan sebuah tantangan. Prinsip-prinsip pada artikel ini akan membantu Anda membangun arsitektur di AWS yang secara efisien menghadirkan kinerja berkelanjutan dari waktu ke waktu. Panduan dan praktik terbaik

dalam dokumen ini tersebar di lima area fokus utama yang berfungsi sebagai prinsip panduan untuk membangun solusi cloud di AWS yang efisien untuk performa. Area-area fokus tersebut adalah:

- Pemilihan arsitektur
- Komputasi dan perangkat keras
- Manajemen Data
- Jaringan dan pengiriman konten
- Proses dan budaya

Artikel ini dimaksudkan untuk orang-orang yang memiliki peran di bidang teknologi, seperti kepala pejabat teknologi (chief technology officer/CTO), arsitek, developer, dan anggota tim operasi. Setelah membaca artikel ini, Anda akan memahami praktik terbaik dan strategi AWS yang digunakan ketika merancang arsitektur cloud berkinerja baik.

Efisiensi kinerja

Pilar efisiensi kinerja berfokus pada penggunaan sumber daya komputasi yang efisien agar memenuhi persyaratan, dan cara memelihara efisiensi seiring dengan perubahan permintaan dan perkembangan teknologi.

Topik

Prinsip desain

Prinsip desain berikut dapat membantu Anda mencapai dan mempertahankan beban kerja yang efisien di cloud.

- Demokratisasikan teknologi canggih: Jadikan implementasi teknologi canggih lebih mudah untuk tim Anda dengan mendelegasikan tugas kompleks kepada vendor cloud. Daripada bertanya kepada tim IT Anda tentang hosting dan menjalankan teknologi baru, manfaatkan teknologi sebagai layanan. Misalnya, basis data NoSQL, transkode media, dan machine learning merupakan teknologi yang memerlukan keahlian khusus. Di cloud, teknologi ini menjadi layanan yang digunakan tim Anda, sehingga tim dapat fokus pada pengembangan produk, bukan penyediaan dan manajemen sumber daya.
- Tersebar secara global dalam hitungan menit: : Melakukan deployment beban kerja ke beberapa Wilayah AWS di seluruh dunia untuk menyediakan latensi yang lebih rendah dan pengalaman yang lebih baik untuk pelanggan dengan biaya minimal.
- Gunakan arsitektur nirserver: : Dengan arsitektur nirserver, Anda tidak perlu menjalankan dan memelihara server fisik untuk aktivitas komputasi tradisional. Misalnya, layanan penyimpanan nirserver dapat bertindak sebagai situs web statis (tanpa memerlukan server web) dan layanan peristiwa dapat melakukan hosting kode. Dengan demikian, beban operasional untuk mengelola server fisik tidak lagi ada, dan biaya transaksional berkurang karena layanan terkelola dioperasikan pada skala cloud.
- Bereksperimen lebih sering: : Dengan sumber daya virtual yang dapat diotomatiskan, Anda dapat melakukan pengujian komparatif dengan cepat menggunakan jenis instans, penyimpanan, atau konfigurasi yang berbeda.

- Selaraskan tujuan dengan penggunaan: Gunakan pendekatan teknologi yang paling sesuai dengan tujuan Anda. Misalnya, pertimbangkan pola akses data saat memilih basis data atau penyimpanan untuk beban kerja Anda.

Definisi

Fokus pada area berikut untuk mencapai efisiensi kinerja di cloud:

- Pemilihan arsitektur
- Komputasi dan perangkat keras
- Manajemen Data
- Jaringan dan pengiriman konten
- Proses dan budaya

Gunakan pendekatan berbasis data untuk membangun arsitektur dengan kinerja tinggi. Kumpulkan data tentang semua aspek arsitektur, dari desain tingkat tinggi hingga pemilihan dan konfigurasi jenis sumber daya.

Peninjauan pilihan secara rutin memastikan bahwa Anda memperoleh manfaat dari AWS Cloud yang terus berkembang. Dengan pemantauan, Anda dapat mengidentifikasi penyimpangan apa pun dari kinerja yang diharapkan. Buat kompensasi dalam arsitektur Anda untuk meningkatkan kinerja, seperti menggunakan kompresi atau caching, atau persyaratan konsistensi yang lebih fleksibel.

Pemilihan arsitektur

Solusi yang optimal bervariasi untuk beban kerja tertentu, dan solusi sering kali menggabungkan beberapa pendekatan. Beban kerja Well-Architected menggunakan beberapa solusi dan memungkinkan berbagai fitur guna meningkatkan kinerja.

Sumber daya AWS tersedia dalam berbagai jenis dan konfigurasi, sehingga memudahkan Anda menemukan pendekatan yang sesuai kebutuhan. Anda juga dapat menemukan opsi yang tidak mudah dicapai dengan infrastruktur on-premise. Misalnya, layanan terkelola seperti Amazon DynamoDB menyediakan basis data NoSQL terkelola penuh dengan latensi satu digit milidetik pada skala berapa pun.

Area fokus ini membagikan panduan dan praktik terbaik tentang cara memilih sumber daya cloud dan pola arsitektur yang efisien dan berkinerja tinggi.

Praktik terbaik

- PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia
- PERF01-BP02 Menggunakan panduan dari penyedia cloud Anda atau mitra yang tepat untuk mempelajari pola arsitektur dan praktik terbaik
- PERF01-BP03 Mempertimbangkan biaya dalam keputusan arsitektur
- PERF01-BP04 Mengevaluasi bagaimana kompromi berdampak pada pelanggan dan efisiensi arsitektur
- PERF01-BP05 Menggunakan kebijakan dan arsitektur referensi
- PERF01-BP06 Menggunakan tolok ukur untuk mendorong keputusan arsitektur
- PERF01-BP07 Menggunakan pendekatan berbasis data untuk pilihan arsitektur

PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia

Terus pelajari dan temukan layanan serta konfigurasi yang tersedia yang membantu Anda mengambil keputusan arsitektur yang lebih baik dan meningkatkan efisiensi kinerja dalam arsitektur beban kerja Anda.

Antipola umum:

- Anda menggunakan cloud sebagai pusat data kolokasi.

- Anda tidak memodernisasi aplikasi Anda setelah migrasi ke cloud.
- Anda hanya menggunakan satu tipe penyimpanan untuk semua hal yang perlu dipertahankan.
- Anda menggunakan tipe instans yang paling sesuai dengan standar Anda saat ini, tetapi lebih besar dari yang diperlukan.
- Anda melakukan deployment dan mengelola teknologi yang tersedia sebagai layanan terkelola.

Manfaat menjalankan praktik terbaik ini: Dengan mempertimbangkan layanan dan konfigurasi baru, Anda mungkin dapat meningkatkan kinerja, mengurangi biaya, dan mengoptimalkan upaya yang diperlukan untuk memelihara beban kerja Anda. Ini juga dapat membantu Anda mempercepat waktu perolehan nilai untuk produk yang didukung cloud.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

AWS terus-menerus merilis layanan dan fitur baru yang dapat meningkatkan kinerja dan mengurangi biaya beban kerja cloud. Mengikuti perkembangan layanan dan fitur baru ini sangat penting untuk menjaga efisiensi kinerja di cloud. Modernisasi arsitektur beban kerja juga membantu Anda mempercepat produktivitas, mendorong inovasi, dan membuka lebih banyak peluang pertumbuhan.

Langkah implementasi

- Buat inventaris arsitektur dan perangkat lunak beban kerja untuk layanan terkait. Tentukan kategori produk mana yang akan dipelajari lebih lanjut.
- Jelajahi penawaran AWS untuk mengidentifikasi dan mempelajari layanan serta opsi konfigurasi yang relevan yang dapat membantu Anda meningkatkan kinerja dan mengurangi biaya serta kompleksitas operasional.
  - Amazon Web Services Cloud
  - AWS Academy
  - Apa yang Baru dengan AWS?
  - Blog AWS
  - AWS Skill Builder
  - Acara dan Webinar AWS
  - AWS Training and Certifications
  - Saluran YouTube AWS

  - Lokakarya AWS
  - Komunitas AWS
- Gunakan lingkungan sandbox (non-produksi) untuk mempelajari dan bereksperimen dengan layanan baru tanpa dikenakan biaya tambahan.
- Terus pelajari layanan dan fitur cloud baru.

Sumber daya

Dokumen terkait:

- Gambaran Umum Amazon Web Services
- Fitur Amazon EC2
- Pelajari langkah demi langkah dengan Rencana Pembelajaran Partner AWS
- AWS Training and Certification
- Jalur pembelajaran saya untuk menjadi arsitek solusi AWS
- Pusat Arsitektur AWS
- AWS Partner Network
- Pustaka Solusi AWS
- Pusat Pengetahuan AWS
- Membangun aplikasi modern di AWS

Video terkait:

- AWS re:Invent 2023 - Apa yang baru dengan Amazon EC2
- AWS re:Invent 2022 - Mengurangi biaya operasional dan infrastruktur Anda dengan Amazon ECS
- AWS re:Invent 2023 - Membangun dengan efisiensi, ketangkasan & inovasi cloud dengan AWS
- AWS re:Invent 2022 - Melakukan deployment model ML untuk inferensi dengan performa tinggi dan biaya rendah
- Ini Arsitektur saya

Contoh terkait:

- Sampel AWS

- Contoh SDK AWS

PERF01-BP02 Menggunakan panduan dari penyedia cloud Anda atau mitra yang tepat untuk mempelajari pola arsitektur dan praktik terbaik

Gunakan sumber daya perusahaan cloud, seperti dokumentasi, arsitek solusi, layanan profesional, atau partner yang tepat untuk memandu keputusan arsitektur Anda. Semua sumber daya ini membantu meninjau dan meningkatkan arsitektur Anda untuk kinerja yang optimal.

Antipola umum:

- Anda menggunakan AWS sebagai penyedia cloud umum.
- Anda menggunakan layanan AWS dengan cara yang tidak sesuai dengan tujuan desainnya.
- Anda mengikuti semua panduan tanpa mempertimbangkan konteks bisnis Anda.

Manfaat menjalankan praktik terbaik ini: Menggunakan panduan dari penyedia cloud atau partner yang tepat dapat membantu Anda membuat pilihan arsitektur yang tepat untuk beban kerja Anda dan memberi Anda kepercayaan diri dalam keputusan Anda.

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

Panduan implementasi

AWS menawarkan berbagai panduan, dokumentasi, dan sumber daya yang dapat membantu Anda membangun dan mengelola beban kerja cloud yang efisien. Dokumentasi AWS menyediakan contoh kode, tutorial, dan penjelasan layanan yang mendetail. Selain dokumentasi, AWS menyediakan program pelatihan dan sertifikasi, arsitek solusi, dan layanan profesional yang dapat membantu pelanggan menjelajahi berbagai aspek layanan cloud dan menerapkan arsitektur cloud yang efisien di AWS.

Manfaatkan semua sumber daya ini untuk mendapatkan wawasan tentang pengetahuan dan praktik terbaik yang berharga, menghemat waktu, dan mencapai hasil yang lebih baik di AWS Cloud.

Langkah implementasi

- Tinjau dokumentasi serta panduan AWS dan ikuti praktik terbaik. Semua sumber daya ini dapat membantu Anda memilih dan mengonfigurasi layanan secara efektif dan mencapai kinerja yang lebih baik.
  - Dokumentasi AWS (seperti panduan pengguna dan laporan resmi)
  - Blog AWS
  - AWS Training and Certifications
  - Saluran YouTube AWS
- Bergabunglah dengan acara partner AWS (seperti AWS Global Summits, AWS Re:Invent, grup pengguna, dan lokakarya) untuk belajar dari para ahli AWS tentang praktik terbaik untuk menggunakan layanan AWS.
  - Pelajari langkah demi langkah dengan Rencana Pembelajaran Partner AWS
  - Acara dan Webinar AWS
  - Lokakarya AWS
  - Komunitas AWS
- Hubungi AWS untuk mendapatkan bantuan saat Anda memerlukan panduan tambahan atau informasi produk. Arsitek Solusi AWS dan Layanan Profesional AWS menyediakan panduan untuk implementasi solusi. Partner AWS menyediakan keahlian AWS untuk membantu Anda menghadirkan ketangkasan dan inovasi untuk bisnis Anda.
- Gunakan AWS Support jika Anda membutuhkan dukungan teknis untuk menggunakan layanan secara efektif. Rencana Dukungan kami dirancang untuk memberi Anda perpaduan alat yang tepat dan akses ke keahlian sehingga Anda dapat berhasil dengan AWS sambil mengoptimalkan kinerja, mengelola risiko, dan menjaga biaya tetap terkendali.

Sumber daya

Dokumen terkait:

- Pusat Arsitektur AWS
- AWS Partner Network
- Pustaka Solusi AWS
- Pusat Pengetahuan AWS
- AWS Enterprise Support

Video terkait:

- Ini Arsitektur saya
- AWS re:Invent 2023 - Pola yang didorong peristiwa tingkat lanjut dengan Amazon EventBridge
- AWS re:Invent 2023 - Mengimplementasikan pola desain terdistribusi di AWS
- AWS re:Invent 2023 - Arsitektur aplikasi sebagai kode

Contoh terkait:

- Sampel AWS
- Contoh SDK AWS
- Arsitektur Referensi Analitik AWS

PERF01-BP03 Mempertimbangkan biaya dalam keputusan arsitektur

Pertimbangkan biaya dalam keputusan arsitektur Anda untuk meningkatkan pemanfaatan sumber daya dan efisiensi kinerja beban kerja cloud Anda. Ketika Anda menyadari implikasi biaya dari beban kerja cloud Anda, Anda kemungkinan akan memanfaatkan sumber daya yang efisien dan mengurangi praktik pemborosan.

Antipola umum:

- Anda hanya menggunakan satu kelompok instans.
- Anda tidak mengevaluasi solusi berlisensi dibandingkan dengan solusi sumber terbuka.
- Anda tidak menentukan kebijakan siklus hidup penyimpanan.
- Anda tidak meninjau layanan dan fitur baru dari AWS Cloud.
- Anda hanya menggunakan penyimpanan blok.

Manfaat menjalankan praktik terbaik ini: Dengan mempertimbangkan biaya dalam pengambilan keputusan, Anda dapat menggunakan sumber daya yang lebih efisien dan mengeksplorasi investasi lainnya.

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

Panduan implementasi

Mengoptimalkan beban kerja untuk biaya dapat meningkatkan pemanfaatan sumber daya dan menghindari pemborosan dalam beban kerja cloud. Mempertimbangkan biaya dalam keputusan arsitektur biasanya mencakup penyesuaian ukuran komponen beban kerja dan menghadirkan elastisitas, yang menghasilkan peningkatan efisiensi kinerja beban kerja cloud.

Langkah implementasi

- Tetapkan sasaran biaya seperti batas anggaran untuk beban kerja cloud Anda.
- Identifikasi komponen utama (seperti instans dan penyimpanan) yang menambah biaya beban kerja Anda. Anda dapat menggunakan AWS Pricing Calculator dan AWS Cost Explorer untuk mengidentifikasi pendorong biaya utama dalam beban kerja Anda.
- Pahami model harga di cloud, seperti Sesuai Permintaan, Instans Terpesan, Savings Plans, dan Instans Spot.
- Gunakan Praktik terbaik pengoptimalan biaya Well-Architected untuk mengoptimalkan komponen kunci ini untuk biaya.
- Teruslah memantau dan menganalisis biaya untuk mengidentifikasi peluang pengoptimalan biaya dalam beban kerja Anda.
  - Gunakan AWS Budgets untuk mendapatkan pemberitahuan adanya biaya yang tidak dapat diterima.
  - Gunakan AWS Compute Optimizer atau AWS Trusted Advisor untuk mendapatkan rekomendasi pengoptimalan biaya.
  - Gunakan AWS Cost Anomaly Detection untuk mendapatkan deteksi anomali biaya dan analisis akar masalah secara otomatis.

Sumber daya

Dokumen terkait:

- Apa Itu Manajemen Penagihan dan Biaya AWS?
- Optimisasi Biaya dengan AWS
- Memilih strategi manajemen biaya AWS
- Panduan Manajemen Biaya AWS bagi Pemula
- Tinjauan Mendetail tentang Dasbor Inteligensi Biaya

- Pusat Arsitektur AWS
- Pustaka Solusi AWS
- Pusat Pengetahuan AWS

Video terkait:

- Ini Arsitektur saya
- AWS re:Invent 2023 - Apa yang baru dengan optimisasi biaya AWS
- AWS re:Invent 2023 - Mengoptimalkan biaya dan kinerja serta melacak kemajuan menuju mitigasi
- AWS re:Invent 2023 - Praktik terbaik optimisasi biaya penyimpanan AWS
- AWS re:Invent 2023 - Mengoptimalkan biaya di lingkungan multiakun

Contoh terkait:

- Kode AWS Compute Optimizer Demo
- Lokakarya Optimisasi Biaya
- Panduan Implementasi Teknis Manajemen Keuangan Cloud
- Pengoptimalan perusahaan rintisan: Menyetel kinerja aplikasi untuk efisiensi maksimum
- Lokakarya Pengoptimalan Nirserver (Kinerja dan Biaya)
- Menskalakan arsitektur hemat biaya

PERF01-BP04 Mengevaluasi bagaimana kompromi berdampak pada pelanggan dan efisiensi arsitektur

Saat mengevaluasi peningkatan terkait kinerja, tentukan pilihan mana yang berdampak pada efisiensi beban kerja dan pelanggan Anda. Misalnya, jika menggunakan penyimpanan data nilai-kunci dapat meningkatkan kinerja sistem, penting untuk mengevaluasi bagaimana dampak sifat eventual consistency-nya nanti terhadap pelanggan.

Antipola umum:

- Anda berasumsi bahwa semua kinerja yang dimiliki harus diimplementasikan, meskipun ada kompromi untuk implementasi.

- Anda hanya mengevaluasi perubahan beban kerja ketika masalah kinerja telah mencapai titik kritis.

Manfaat menjalankan praktik terbaik ini: Ketika Anda mengevaluasi potensi peningkatan terkait performa, Anda harus menentukan apakah kompromi untuk perubahan dapat diterima dengan persyaratan beban kerja. Dalam beberapa kasus, Anda mungkin harus mengimplementasikan beberapa kontrol tambahan untuk mengimbangi kompensasi.

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

Panduan implementasi

Identifikasi area kritis dalam arsitektur Anda dalam hal dampak terhadap kinerja dan pelanggan. Tentukan cara Anda mewujudkan peningkatan, kompromi seperti apa yang ditimbulkan peningkatan, serta bagaimana pengaruhnya terhadap sistem dan pengalaman pengguna. Misalnya, mengimplementasikan pembuatan cache data dapat membantu meningkatkan kinerja secara signifikan tetapi memerlukan strategi yang jelas terkait cara dan waktu untuk memperbarui atau menonaktifkan data yang di-cache guna mencegah perilaku sistem yang tidak sesuai.

Langkah implementasi

- Pahami persyaratan beban kerja dan SLA Anda.
- Tentukan faktor evaluasi secara jelas. Faktor-faktor mungkin berhubungan dengan biaya, keandalan, keamanan, dan kinerja beban kerja Anda.
- Pilih arsitektur dan layanan yang dapat memenuhi kebutuhan Anda.
- Lakukan eksperimen dan bukti konsep (POC) untuk mengevaluasi faktor kompromi dan dampak terhadap pelanggan dan efisiensi arsitektur. Biasanya, beban kerja dengan ketersediaan tinggi, berkinerja tinggi, dan aman mengonsumsi lebih banyak sumber daya cloud sekaligus memberikan pengalaman pelanggan yang lebih baik. Pahami kompromi antara kompleksitas, kinerja, dan biaya beban kerja Anda. Umumnya, ketika dua faktor diprioritaskan, faktor ketiga akan dikorbankan.

Sumber daya

Dokumen terkait:

- Amazon Builders' Library
- KPI Amazon QuickSight
- Amazon CloudWatch RUM

- Dokumentasi X-Ray
- Memahami pola ketahanan dan kompromi untuk merancang secara efisien di cloud

Video terkait:

- Optimalkan aplikasi dengan Amazon CloudWatch RUM
- AWS re:Invent 2023 - Kapasitas, ketersediaan, efisiensi biaya: Pilih tiga
- AWS re:Invent 2023 - Pola integrasi tingkat lanjut & kompromi untuk sistem yang digabungkan secara longgar

Contoh terkait:

- Ukur waktu pemuatan halaman dengan Amazon CloudWatch Synthetics
- Klien Web Amazon CloudWatch RUM

PERF01-BP05 Menggunakan kebijakan dan arsitektur referensi

Gunakan kebijakan internal dan arsitektur referensi yang ada saat memilih layanan dan konfigurasi agar lebih efisien saat merancang dan mengimplementasikan beban kerja Anda.

Antipola umum:

- Anda mengizinkan berbagai macam teknologi yang berdampak pada biaya manajemen biaya perusahaan.

Manfaat menjalankan praktik terbaik ini: Dengan menetapkan kebijakan untuk pilihan arsitektur, teknologi, dan vendor, keputusan dapat diambil dengan lebih cepat.

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

Panduan implementasi

Adanya kebijakan internal dalam memilih sumber daya dan arsitektur memberikan standar dan pedoman untuk diikuti ketika membuat pilihan arsitektur. Pedoman tersebut merampingkan proses pengambilan keputusan saat memilih layanan cloud yang tepat dan dapat membantu meningkatkan efisiensi kinerja. Lakukan deployment beban kerja Anda menggunakan arsitektur referensi atau

kebijakan. Integrasikan layanan ke dalam deployment cloud, lalu gunakan pengujian kinerja untuk memastikan bahwa Anda dapat terus memenuhi persyaratan kinerja.

Langkah implementasi

- Pahami dengan jelas persyaratan beban kerja cloud Anda.
- Tinjau kebijakan internal dan eksternal untuk mengidentifikasi kebijakan yang paling relevan.
- Gunakan arsitektur referensi yang sesuai yang disediakan oleh AWS atau praktik terbaik industri Anda.
- Buat rangkaian yang terdiri dari kebijakan, standar, arsitektur referensi, dan pedoman preskriptif untuk situasi umum. Tindakan tersebut memungkinkan tim Anda bergerak lebih cepat. Sesuaikan aset untuk bidang Anda jika perlu.
- Validasi kebijakan dan arsitektur referensi ini untuk beban kerja Anda di lingkungan sandbox.
- Terus ikuti perkembangan standar industri dan pembaruan AWS untuk memastikan kebijakan dan arsitektur referensi Anda membantu mengoptimalkan beban kerja cloud Anda.

Sumber daya

Dokumen terkait:

- Pusat Arsitektur AWS
- AWS Partner Network
- Pustaka Solusi AWS
- Pusat Pengetahuan AWS
- Blog Arsitektur AWS

Video terkait:

- Ini Arsitektur saya
- AWS re:Invent 2022 - Percepat nilai bisnis Anda dengan SAP & arsitektur referensi AWS

Contoh terkait:

- Sampel AWS
- Contoh SDK AWS

PERF01-BP06 Menggunakan tolok ukur untuk mendorong keputusan arsitektur

Lakukan tolok ukur pada kinerja beban kerja yang ada untuk memahami kinerjanya di cloud dan mendorong keputusan arsitektur berdasarkan data tersebut.

Antipola umum:

- Anda mengandalkan tolok ukur umum yang tidak mewakili karakteristik beban kerja Anda.
- Anda bergantung pada persepsi dan tanggapan pelanggan sebagai satu-satunya tolok ukur.

Manfaat menerapkan praktik terbaik ini: Melalui tolok ukur implementasi saat ini, Anda dapat mengukur peningkatan kinerja Anda.

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

Panduan implementasi

Gunakan benchmarking dengan pengujian sintetis untuk menilai kinerja komponen beban kerja Anda. Benchmarking umumnya dapat disiapkan dengan lebih cepat daripada pengujian beban dan digunakan untuk mengevaluasi teknologi untuk komponen tertentu. Benchmarking sering digunakan pada awal proyek baru, saat Anda tidak memiliki solusi lengkap untuk memuat pengujian.

Anda dapat merancang pengujian tolok ukur kustom atau menggunakan pengujian standar industri, misalnya TPC-DS, untuk menolok ukur beban kerja Anda. Tolok ukur industri sangat membantu saat memperbandingkan lingkungan. Tolok ukur kustom bermanfaat untuk menargetkan jenis operasi tertentu yang ingin dibuat dalam arsitektur.

Saat melakukan tolok ukur, penting untuk menyiapkan lingkungan terlebih dahulu untuk memastikan hasil yang valid. Jalankan tolok ukur yang sama beberapa kali untuk memastikan Anda memperoleh variasi apa pun dari waktu ke waktu.

Karena tolok ukur umumnya lebih cepat untuk menjalankan pengujian daripada memuatnya, maka tolok ukur dapat digunakan terlebih dahulu dalam deployment pipeline dan memberikan umpan balik pada deviasi kinerja. Saat Anda mengevaluasi perubahan yang signifikan dalam komponen atau layanan, tolok ukur dapat menjadi cara cepat guna menentukan apakah perubahan memang perlu dibuat. Menggunakan benchmarking bersama dengan pengujian beban begitu penting karena pengujian beban memberi tahu Anda tentang bagaimana kinerja beban kerja Anda dalam produksi.

Langkah implementasi

- Rencanakan dan tentukan:
  - Tentukan tujuan, acuan dasar, skenario pengujian, metrik (seperti pemanfaatan CPU, latensi, atau throughput), dan KPI untuk tolok ukur Anda.
  - Fokus pada persyaratan pengguna dalam hal pengalaman pengguna dan faktor-faktor seperti waktu respons dan aksesibilitas.
  - Identifikasi alat tolok ukur yang sesuai dengan beban kerja Anda. Anda dapat menggunakan layanan AWS seperti Amazon CloudWatch atau alat pihak ketiga yang kompatibel dengan beban kerja Anda.
- Konfigurasi dan persiapkan:
  - Siapkan lingkungan Anda dan konfigurasikan sumber daya Anda.
  - Implementasikan pemantauan dan pembuatan log untuk merekam hasil pengujian.
- Lakukan tolok ukur dan pemantauan:
  - Lakukan pengujian tolok ukur Anda dan pantau metrik selama pengujian.
- Analisis dan dokumentasikan:
  - Dokumentasikan proses dan temuan tolok ukur Anda.
  - Analisis hasil untuk mengidentifikasi hambatan, tren, dan area perbaikan.
  - Gunakan hasil pengujian untuk mengambil keputusan arsitektur dan menyesuaikan beban kerja Anda. Termasuk di dalamnya mungkin adalah mengubah layanan atau mengadopsi fitur baru.
- Optimalkan dan ulangi:
  - Sesuaikan konfigurasi dan alokasi sumber daya berdasarkan tolok ukur Anda.
  - Uji ulang beban kerja Anda setelah penyesuaian untuk memvalidasi perbaikan Anda.
  - Dokumentasikan pembelajaran Anda, dan ulangi proses untuk mengidentifikasi area perbaikan lainnya.

Sumber daya

Dokumen terkait:

- Pusat Arsitektur AWS
- AWS Partner Network
- Pustaka Solusi AWS

- Pusat Pengetahuan AWS
- RUM Amazon CloudWatch
- Amazon CloudWatch Synthetics
- Alur kerja genomik, Bagian 5: pembuatan tolok ukur otomatis
- Lakukan tolok ukur dan optimalkan deployment titik akhir di Amazon SageMaker JumpStart

Video terkait:

- AWS re:Invent 2023 - Pembuatan tolok ukur cold start AWS Lambda
- Pembuatan tolok ukur layanan stateful di cloud
- Ini Arsitektur saya
- Optimalkan aplikasi dengan RUM Amazon CloudWatch
- Demo Amazon CloudWatch Synthetics

Contoh terkait:

- Sampel AWS
- Contoh SDK AWS
- Pengujian Beban Terdistribusi
- Ukur waktu pemuatan halaman dengan Amazon CloudWatch Synthetics
- Klien Web RUM Amazon CloudWatch

PERF01-BP07 Menggunakan pendekatan berbasis data untuk pilihan arsitektur

Tentukan pendekatan yang jelas dan berbasis data untuk pilihan arsitektur guna memastikan layanan dan konfigurasi cloud yang tepat digunakan untuk memenuhi kebutuhan bisnis spesifik Anda.

Antipola umum:

- Anda berasumsi bahwa arsitektur Anda saat ini statis dan tidak perlu diperbarui dari waktu ke waktu.
- Pilihan arsitektur Anda didasarkan pada tebakan dan asumsi.
- Anda memperkenalkan perubahan arsitektur seiring waktu tanpa justifikasi.

Manfaat menjalankan praktik terbaik ini: Dengan memiliki pendekatan yang terdefinisi dengan baik dalam membuat pilihan arsitektur, Anda menggunakan data untuk memengaruhi desain beban kerja Anda dan mengambil keputusan berdasarkan informasi dari waktu ke waktu.

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

Panduan implementasi

Gunakan pengalaman internal dan pengetahuan tentang cloud, atau sumber daya eksternal seperti kasus penggunaan yang dipublikasi atau laporan resmi untuk memilih sumber daya dan layanan di arsitektur Anda. Anda harus memiliki proses yang terdefinisi dengan baik yang mendorong eksperimen dan tolok ukur dengan layanan yang bisa digunakan pada beban kerja Anda.

Backlog untuk beban kerja kritis tidak boleh hanya terdiri dari cerita pengguna yang memberikan fungsionalitas yang relevan dengan bisnis dan pengguna, melainkan juga harus berisi cerita teknis yang membentuk landasan arsitektur untuk beban kerja. Landasan ini didasarkan pada kemajuan teknologi baru serta layanan baru dan mengadopsinya berdasarkan data dan pembenaran yang tepat. Hal ini memastikan bahwa arsitektur tetap relevan di masa depan dan tidak jalan di tempat.

Langkah implementasi

- Lakukan interaksi dengan pemangku kepentingan utama untuk menentukan persyaratan beban kerja, termasuk kinerja, ketersediaan, dan pertimbangan biaya. Pertimbangkan faktor-faktor seperti jumlah pengguna dan pola penggunaan untuk beban kerja Anda.
- Ciptakan landasan arsitektur atau backlog teknologi yang diprioritaskan bersamaan dengan backlog fungsional.
- Evaluasi dan nilai berbagai layanan cloud (untuk detail selengkapnya, lihat PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia).
- Jelajahi pola-pola arsitektur yang berbeda, seperti layanan mikro atau nirserver, yang memenuhi persyaratan kinerja Anda (untuk detail selengkapnya, lihat PERF01-BP02 Menggunakan panduan dari penyedia cloud Anda atau mitra yang tepat untuk mempelajari pola arsitektur dan praktik terbaik).

- Belajar dari tim lain, diagram arsitektur, dan sumber daya seperti Arsitek Solusi AWS, Pusat Arsitektur AWS, dan AWS Partner Network, untuk membantu Anda memilih arsitektur yang tepat untuk beban kerja Anda.

- Tentukan metrik kinerja seperti throughput dan waktu respons yang dapat membantu Anda mengevaluasi kinerja beban kerja Anda.
- Lakukan eksperimen dan gunakan metrik yang ditentukan untuk memvalidasi kinerja arsitektur yang dipilih.
- Teruslah memantau dan melakukan penyesuaian sesuai kebutuhan untuk mempertahankan kinerja optimal arsitektur Anda.
- Dokumentasikan arsitektur dan keputusan pilihan Anda sebagai referensi untuk pembaruan dan pembelajaran di masa mendatang.
- Teruslah meninjau dan memperbarui pendekatan pemilihan arsitektur berdasarkan pembelajaran, teknologi baru, dan metrik yang menunjukkan kebutuhan perubahan atau masalah dalam pendekatan saat ini.

Sumber daya

Dokumen terkait:

- Pustaka Solusi AWS
- Pusat Pengetahuan AWS
- Pola Arsitektur untuk Membangun Aplikasi yang Didorong Data Menyeluruh di AWS

Video terkait:

- Ini Arsitektur saya
- AWS re:Invent 2021 - Korporasi yang didorong data: Bergerak dari visi menuju nilai
- AWS re:Invent 2022 - Menghadirkan arsitektur berkelanjutan dan berkinerja tinggi
- AWS re:Invent 2023 - Mengoptimalkan biaya dan kinerja serta melacak kemajuan menuju mitigasi
- AWS re:Invent 2022 - Optimisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung

Contoh terkait:

- Sampel AWS
- Contoh SDK AWS

Komputasi dan perangkat keras

Pilihan komputasi yang optimal untuk beban kerja tertentu bervariasi berdasarkan desain aplikasi, pola penggunaan, dan pengaturan konfigurasi. Arsitektur dapat menggunakan pilihan komputasi yang berbeda untuk berbagai komponen, dan memungkinkan fitur yang berbeda untuk meningkatkan kinerja. Memilih pilihan komputasi yang salah untuk arsitektur dapat menyebabkan efisiensi kinerja menjadi lebih rendah.

Area fokus ini membagikan panduan dan praktik terbaik tentang cara mengidentifikasi dan mengoptimalkan opsi komputasi untuk efisiensi kinerja di cloud.

Praktik terbaik

- PERF02-BP01 Memilih opsi komputasi terbaik untuk beban kerja Anda
- PERF02-BP02 Memahami konfigurasi dan fitur komputasi yang tersedia
- PERF02-BP03 Mengumpulkan komputasi metrik terkait
- PERF02-BP04 Mengonfigurasi dan menyesuaikan ukuran sumber daya komputasi
- PERF02-BP05 Menskalakan sumber daya komputasi Anda secara dinamis
- PERF02-BP06 Menggunakan akselerator komputasi berbasis perangkat keras yang dioptimalkan

PERF02-BP01 Memilih opsi komputasi terbaik untuk beban kerja Anda

Dengan memilih opsi komputasi yang paling tepat untuk beban kerja, Anda dapat meningkatkan kinerja, mengurangi biaya infrastruktur yang tidak perlu, dan menurunkan upaya operasional yang diperlukan untuk memelihara beban kerja Anda.

Antipola umum:

- Anda menggunakan opsi komputasi yang sama yang digunakan secara on-premise.
- Anda tidak mengetahui opsi, fitur, dan solusi komputasi cloud, dan bagaimana solusi tersebut dapat meningkatkan kinerja komputasi Anda.
- Anda melakukan pengadaan opsi komputasi yang berlebihan untuk memenuhi persyaratan penskalaan atau kinerja ketika ada opsi komputasi lain yang lebih sesuai dengan karakteristik beban kerja Anda.

Manfaat menerapkan praktik terbaik ini: Dengan mengidentifikasi persyaratan komputasi dan mengevaluasi opsi-opsi yang tersedia, Anda dapat membuat beban kerja Anda lebih hemat sumber daya.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Untuk mengoptimalkan beban kerja cloud Anda demi efisiensi kinerja, penting untuk memilih opsi komputasi yang paling sesuai dengan kasus penggunaan dan persyaratan kinerja Anda. AWS menyediakan berbagai opsi komputasi yang sesuai untuk berbagai beban kerja di cloud. Misalnya, Anda dapat menggunakan Amazon EC2 untuk meluncurkan dan mengelola server virtual, AWS Lambda untuk menjalankan kode tanpa harus menyediakan atau mengelola server, Amazon ECS atau Amazon EKS untuk menjalankan dan mengelola kontainer, atau AWS Batch untuk memproses volume data yang besar secara paralel. Berdasarkan skala dan kebutuhan komputasi Anda, Anda harus memilih dan mengonfigurasi solusi komputasi yang optimal untuk situasi Anda. Anda juga dapat mempertimbangkan untuk menggunakan beberapa jenis solusi komputasi dalam satu beban kerja, karena masing-masing memiliki kelebihan dan kekurangannya sendiri.

Langkah-langkah berikut ini memandu Anda dalam memilih opsi komputasi yang tepat agar sesuai dengan karakteristik beban kerja dan persyaratan kinerja Anda.

Langkah implementasi

- Pahami persyaratan komputasi beban kerja Anda. Persyaratan utama yang harus dipertimbangkan antara lain kebutuhan pemrosesan, pola lalu lintas, pola akses data, kebutuhan penskalaan, dan persyaratan latensi.
- Pelajari berbagai opsi komputasi yang tersedia untuk beban kerja Anda di AWS (seperti yang diuraikan dalam PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia. Berikut adalah beberapa opsi komputasi AWS utama, karakteristiknya, dan kasus penggunaan umumnya:

AWS service	Key characteristics	Common use cases
Amazon Elastic Compute Cloud (Amazon EC2)	Has dedicated option for hardware, license requirements, large selection of different instance families, processor	Lift and shift migrations, monolithic application, hybrid environments, enterprise applications

AWS service	Key characteristics	Common use cases
	types and compute accelerators	
Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS)	Easy deployment, consistent environments, scalable	Microservices, hybrid environments
AWS Lambda	Komputasi nirserver service that runs code in response to events and automatically manages the underlying compute resources.	Microservices, event-driven applications
AWS Batch	Efficiently and dynamically provisions and scales Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), and AWS Fargate compute resources, with an option to use On-Demand or Spot Instances based on your job requirements	HPC, train ML models
Amazon Lightsail	Preconfigured Linux and Windows application for running small workloads	Simple web applications, custom website

- Lakukan evaluasi biaya (seperti biaya per jam atau transfer data) dan overhead manajemen (seperti patching dan penskalaan) yang terkait dengan setiap opsi komputasi.
- Lakukan uji coba dan uji tolok ukur di lingkungan nonproduksi untuk mengidentifikasi opsi komputasi mana yang paling sesuai dengan kebutuhan beban kerja Anda.

- Setelah menguji coba dan mengidentifikasi solusi komputasi baru Anda, rencanakan migrasi dan validasikan metrik kinerja Anda.
- Gunakan alat pemantauan AWS seperti Amazon CloudWatch dan layanan optimisasi seperti AWS Compute Optimizer untuk terus mengoptimalkan sumber daya komputasi Anda berdasarkan pola penggunaan dunia nyata.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Tipe Instans Amazon EC2
- Kontainer Amazon EKS: Simpul Pekerja Amazon EKS
- Kontainer Amazon ECS: Instans Kontainer Amazon ECS
- Fungsi: Konfigurasi Fungsi Lambda
- Panduan Preskriptif untuk Kontainer
- Panduan Preskriptif untuk Nirserver

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 AMS
- AWS re:Invent 2023 - Apa yang baru dengan Amazon Elastic Compute Cloud
- AWS re:Invent 2023 - Penghematan cerdas: Strategi optimisasi biaya Amazon Elastic Compute Cloud
- AWS re:Invent 2021 - Mendukung Amazon Elastic Compute Cloud generasi berikutnya: Mendalami Sistem Nitro
- AWS re:Invent 2019 - Mengoptimalkan performa dan biaya untuk komputasi AWS Anda
- AWS re:Invent 2019 - Fondasi Amazon Elastic Compute Cloud
- AWS re:Invent 2022 - Melakukan deployment model ML untuk inferensi dengan performa tinggi dan biaya rendah
- AWS re:Invent 2019 - Mengoptimalkan performa dan biaya untuk komputasi AWS Anda
- Fondasi Amazon EC2

- Melakukan deployment model ML untuk inferensi dengan performa tinggi dan biaya rendah

Contoh terkait:

- Memigrasikan Aplikasi web ke kontainer
- Jalankan Hello World Nirserver
- Lokakarya Amazon EKS
- Lokakarya Amazon EC2
- Beban Kerja yang Efisien dan Tangguh dengan Penskalaan Otomatis Amazon Elastic Compute Cloud
- Bermigrasi ke AWS Graviton dengan Layanan Kontainer

PERF02-BP02 Memahami konfigurasi dan fitur komputasi yang tersedia

Pahami opsi dan fitur konfigurasi yang tersedia bagi layanan komputasi Anda untuk membantu Anda menyediakan jumlah sumber daya yang tepat dan meningkatkan efisiensi kinerja.

Antipola umum:

- Anda tidak mengevaluasi opsi komputasi atau family instans yang tersedia berdasarkan karakteristik beban kerja.
- Anda menyediakan sumber daya komputasi secara berlebihan untuk memenuhi persyaratan permintaan puncak.

Manfaat menjalankan praktik terbaik ini: Pahami fitur dan konfigurasi komputasi AWS sehingga Anda dapat menggunakan solusi komputasi yang dioptimalkan untuk memenuhi karakteristik dan kebutuhan beban kerja Anda.

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

Panduan implementasi

Setiap solusi komputasi memiliki konfigurasi dan fitur unik yang tersedia untuk mendukung berbagai karakteristik dan persyaratan beban kerja. Pelajari bagaimana opsi-opsi tersebut melengkapi beban kerja Anda, dan tentukan opsi konfigurasi yang terbaik untuk aplikasi Anda. Contoh dari opsi

tersebut meliputi family instans, ukuran, fitur (GPU, I/O), lonjakan, waktu habis, ukuran fungsi, instans kontainer, dan konkurensi. Jika beban kerja Anda telah menggunakan opsi komputasi yang sama selama lebih dari empat pekan dan Anda mengantisipasi bahwa karakteristiknya akan tetap sama di masa depan, Anda dapat menggunakan AWS Compute Optimizer untuk mengetahui apakah opsi komputasi Anda saat ini cocok untuk beban kerja dari perspektif CPU dan memori.

Langkah implementasi

1. Pahami persyaratan beban kerja (seperti kebutuhan CPU, memori, dan latensi).
2. Tinjau dokumentasi dan praktik terbaik AWS untuk mempelajari rekomendasi opsi konfigurasi yang dapat membantu meningkatkan kinerja komputasi. Berikut adalah beberapa opsi konfigurasi utama yang perlu dipertimbangkan:

Opsi konfigurasi	Contoh
Jenis instans	• Instans komputasi yang dioptimalkan ideal untuk beban kerja yang membutuhkan rasio vCPU terhadap memori yang lebih tinggi. • Instans memori yang dioptimalkan mengirimkan sejumlah besar memori untuk mendukung beban kerja intensif memori. • Instans penyimpanan yang dioptimalkan didesain untuk beban kerja yang memerlukan akses baca dan tulis sekuensial (IOPS) yang tinggi ke penyimpanan lokal.
Model harga	• Instans Sesuai Permintaan memungkinkan Anda menggunakan kapasitas komputasi per jam atau per detik tanpa komitmen jangka panjang. Instans ini bagus untuk lonjakan di atas kebutuhan dasar kinerja. • Savings Plans menawarkan penghematan yang signifikan atas Instans Sesuai Permintaan dengan komitmen untuk

Opsi konfigurasi	Contoh
	menggunakan daya komputasi dalam jumlah tertentu selama jangka waktu satu atau tiga tahun. • Instans Spot memungkinkan Anda memanfaatkan kapasitas instans yang tidak terpakai untuk beban kerja stateless dan toleran terhadap kesalahan.
Auto Scaling	Gunakan konfigurasi Auto Scaling untuk mencocokkan sumber daya komputasi dengan pola lalu lintas.
Penyesuaian ukuran	• Gunakan Compute Optimizer untuk mendapatkan rekomendasi berbasis machine learning mengenai konfigura si komputasi yang paling cocok dengan karakteristik komputasi Anda. • Gunakan AWS Lambda Power Tuning untuk memilih konfigurasi terbaik untuk fungsi Lambda Anda.
Akselerator komputasi berbasis perangkat keras	• Instans komputasi terakselerasi menjalank an fungsi seperti pemrosesan grafis atau pencocokan pola data secara lebih efisien daripada alternatif berbasis CPU. • Untuk beban kerja machine learning, manfaatkan perangkat keras yang dibuat khusus untuk beban kerja Anda, seperti AWS Trainium, AWS Inferentia, dan Amazon EC2 DL1

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Tipe Instans Amazon EC2
- Kontrol Status Prosesor untuk Instans Amazon EC2 Anda
- Kontainer Amazon EKS: Simpul Pekerja Amazon EKS
- Kontainer Amazon ECS: Instans Kontainer Amazon ECS
- Fungsi: Konfigurasi Fungsi Lambda

Video terkait:

- AWS re:Invent 2023 – AWS Graviton: Performa harga terbaik untuk beban kerja AWS Anda
- AWS re:Invent 2023 – Kemampuan AI generatif Amazon EC2 baru di AWS Management Console
- AWS re:Invent 2023 – Apa yang baru dengan Amazon EC2
- AWS re:Invent 2023 – Penghematan cerdas: Strategi optimisasi biaya Amazon EC2
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Mendalami Sistem Nitro
- AWS re:Invent 2019 – Landasan Amazon EC2
- AWS re:Invent 2022 – https://www.youtube.com/watch?v=5B4-s_ivn1o

Contoh terkait:

- Kode demo Compute Optimizer
- Lokakarya instans spot Amazon EC2
- Beban Kerja yang Efisien dan Tangguh dengan Amazon EC2 AWS Auto Scaling
- Lokakarya pengembang Graviton
- Hari imersi beban kerja AWS for Microsoft
- Hari imersi beban kerja AWS for Linux
- Kode AWS Compute Optimizer Demo
- Lokakarya Amazon EKS

PERF02-BP03 Mengumpulkan komputasi metrik terkait

Rekam dan lacak metrik terkait komputasi untuk lebih memahami kinerja sumber daya komputasi Anda dan meningkatkan kinerja serta pemanfaatannya.

Antipola umum:

- Anda hanya menggunakan pencarian file log manual untuk metrik.
- Anda hanya menggunakan metrik default yang dicatat oleh perangkat lunak pemantauan Anda.
- Anda hanya meninjau metrik ketika terdapat masalah.

Manfaat menjalankan praktik terbaik ini: Mengumpulkan metrik terkait kinerja akan membantu Anda menyelaraskan kinerja aplikasi dengan persyaratan bisnis untuk memastikan Anda memenuhi kebutuhan beban kerja Anda. Ini juga dapat membantu Anda terus meningkatkan kinerja dan pemanfaatan sumber daya dalam beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Beban kerja dapat menghasilkan data dalam jumlah besar seperti metrik, log, dan event. Di AWS Cloud, mengumpulkan metrik adalah langkah penting untuk meningkatkan keamanan, efisiensi biaya, kinerja, dan keberlanjutan. AWS menyediakan berbagai metrik terkait kinerja menggunakan layanan pemantauan seperti Amazon CloudWatch untuk memberi Anda wawasan yang berharga. Metrik seperti penggunaan CPU, penggunaan memori, I/O disk, serta lalu lintas masuk dan keluar jaringan dapat memberikan wawasan tentang hambatan kinerja atau tingkat penggunaan. Gunakan metrik ini sebagai bagian dari pendekatan berdasarkan data yang digunakan untuk mengatur dan mengoptimalkan sumber daya beban kerja Anda. Dalam kasus yang ideal, Anda harus mengumpulkan semua metrik yang terkait dengan sumber daya komputasi Anda dalam satu platform dengan kebijakan retensi yang diterapkan untuk mendukung sasaran biaya dan operasional.

Langkah implementasi

1. Identifikasi metrik terkait kinerja apa saja yang relevan dengan beban kerja Anda. Anda harus mengumpulkan metrik seputar pemanfaatan sumber daya dan cara cloud Anda beroperasi (seperti waktu respons dan throughput).
   a. Metrik default Amazon EC2
   b. Metrik default Amazon ECS
   c. Metrik default Amazon EKS
   d. Metrik default Lambda
   e. Metrik disk dan memori Amazon EC2

2. Pilih dan siapkan solusi pembuatan log dan pemantauan yang tepat untuk beban kerja Anda.
    a. Observabilitas native AWS
    b. AWS Distro for OpenTelemetry
    c. Amazon Managed Service for Prometheus
3. Tentukan filter dan agregasi yang diperlukan untuk metrik berdasarkan persyaratan beban kerja Anda.
    a. Mengukur metrik aplikasi kustom dengan Amazon CloudWatch Logs dan filter metrik
    b. Mengumpulkan metrik kustom dengan pembuatan tag strategis Amazon CloudWatch
4. Konfigurasikan kebijakan retensi data untuk metrik Anda agar sesuai dengan sasaran keamanan dan operasional Anda.
    a. Retensi data default untuk metrik CloudWatch
    b. Retensi data default untuk CloudWatch Logs
5. Jika diperlukan, buat alarm dan notifikasi untuk metrik Anda agar membantu Anda merespons masalah terkait kinerja secara proaktif.
    a. Membuat alarm untuk metrik kustom menggunakan deteksi anomali Amazon CloudWatch
    b. Membuat metrik dan alarm untuk halaman web tertentu dengan Amazon CloudWatch RUM
6. Gunakan otomatisasi untuk melakukan deployment agen agregasi log dan metrik Anda.
    a. Otomatisasi AWS Systems Manager
    b. OpenTelemetry Collector

Sumber daya

Dokumen terkait:

- Pemantauan dan observabilitas
- Praktik terbaik: mengimplementasikan observabilitas dengan AWS
- Dokumentasi Amazon CloudWatch
- Kumpulkan metrik dan log dari instans Amazon EC2 serta server on-premise dengan Agen CloudWatch
- Mengakses Amazon CloudWatch Logs untuk AWS Lambda
- Menggunakan CloudWatch Logs dengan instans kontainer
- Publikasikan metrik kustom

- Jawaban AWS: Pencatatan Log Terpusat
- Layanan AWS yang Memublikasikan Metrik CloudWatch
- Memantau Amazon EKS pada AWS Fargate

Video terkait:

- AWS re:Invent 2023 – [PELUNCURAN] Pemantauan aplikasi untuk beban kerja modern
- AWS re:Invent 2023 – Mengimplementasikan observabilitas aplikasi
- AWS re:Invent 2023 – Membangun strategi observabilitas yang efektif
- AWS re:Invent 2023 – Observabilitas mulus dengan AWS Distro for OpenTelemetry
- Manajemen Kinerja Aplikasi di AWS

Contoh terkait:

- Hari Imersi Beban Kerja AWS untuk Linux - Amazon CloudWatch
- Memantau klaster dan kontainer Amazon ECS
- Pemantauan dengan dasbor Amazon CloudWatch
- Lokakarya Amazon EKS

PERF02-BP04 Mengonfigurasi dan menyesuaikan ukuran sumber daya komputasi

Konfigurasikan dan tentukan ukuran yang tepat untuk sumber daya agar sesuai dengan persyaratan kinerja beban kerja Anda dan hindari sumber daya dengan pemanfaatan yang terlalu rendah atau terlalu tinggi.

Antipola umum:

- Anda mengabaikan persyaratan kinerja beban kerja yang menghasilkan sumber daya komputasi dengan pemanfaatan yang terlalu rendah atau terlalu tinggi.
- Anda hanya memilih instans terbesar atau terkecil untuk semua beban kerja.
- Anda hanya menggunakan satu family instans untuk kemudahan manajemen.
- Anda mengabaikan rekomendasi dari AWS Cost Explorer atau Compute Optimizer untuk penentuan ukuran yang tepat.

- Anda tidak mengevaluasi ulang beban kerja untuk kesesuaian tipe instans baru.
- Anda hanya mengesahkan sejumlah kecil konfigurasi instans untuk organisasi Anda.

Manfaat menjalankan praktik terbaik ini: Penyesuaian ukuran yang tepat untuk sumber daya komputasi memastikan pengoperasian yang optimal di cloud dengan menghindari penyediaan sumber daya yang terlalu banyak dan terlalu sedikit. Penyesuaian ukuran sumber daya komputasi secara tepat biasanya menghasilkan kinerja yang lebih baik dan pengalaman pelanggan yang ditingkatkan, sekaligus menurunkan biaya.

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

Panduan implementasi

Penentuan ukuran yang tepat memungkinkan organisasi untuk mengoperasikan infrastruktur cloud mereka dengan cara yang efisien dan hemat biaya sambil menangani kebutuhan bisnis mereka. Penyediaan sumber daya cloud yang berlebihan dapat menyebabkan biaya tambahan, sementara penyediaan yang kurang dapat mengakibatkan kinerja yang buruk dan pengalaman pelanggan yang negatif. AWS menyediakan alat seperti AWS Compute Optimizer dan AWS Trusted Advisor yang menggunakan data historis untuk memberikan rekomendasi untuk menyesuaikan ukuran sumber daya komputasi yang tepat.

Langkah implementasi

- Pilih tipe instans yang paling sesuai dengan kebutuhan Anda:
  - Bagaimana cara memilih jenis instans Amazon EC2 yang tepat untuk beban kerja saya?
  - Pemilihan jenis instans berdasarkan atribut untuk Armada Amazon EC2
  - Membuat grup Auto Scaling menggunakan pemilihan jenis instans berdasarkan atribut
  - Mengoptimalkan biaya komputasi Kubernetes Anda dengan konsolidasi Karpenter
- Analisa berbagai karakteristik kinerja beban kerja Anda serta kaitannya dengan penggunaan memori, jaringan, dan CPU. Gunakan data ini untuk memilih sumber daya yang paling sesuai dengan profil beban kerja dan sasaran kinerja Anda.
- Pantau penggunaan sumber daya Anda menggunakan alat pemantauan AWS sepertiAmazon CloudWatch.
- Pilih konfigurasi yang tepat untuk sumber daya komputasi.

  - Untuk beban kerja sementara, evaluasi metrik Amazon CloudWatch instans seperti `CPUUtilization` untuk mengidentifikasi apakah pemanfaatan instans terlalu rendah atau terlalu tinggi.
  - Untuk beban kerja stabil, periksa alat penyesuaian ukuran AWS seperti AWS Compute Optimizer dan AWS Trusted Advisor secara berkala untuk mengidentifikasi peluang untuk mengoptimalkan dan menyesuaikan ukuran sumber daya komputasi dengan tepat.
- Uji perubahan konfigurasi di lingkungan nonproduksi sebelum diimplementasikan di lingkungan langsung.
- Terus evaluasi ulang penawaran komputasi baru dan bandingkan berdasarkan kebutuhan beban kerja Anda.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Tipe Instans Amazon EC2
- Kontainer Amazon ECS: Instans Kontainer Amazon ECS
- Kontainer Amazon EKS: Simpul Pekerja Amazon EKS
- Fungsi: Konfigurasi Fungsi Lambda
- Kontrol Status Prosesor untuk Instans Amazon EC2 Anda

Video terkait:

- Fondasi Amazon EC2
- AWS re:Invent 2023 – AWS Graviton: Performa harga terbaik untuk beban kerja AWS Anda
- AWS re:Invent 2023 – Kemampuan AI generatif Amazon EC2 baru di AWS Management Console
- AWS re:Invent 2023 – Apa yang baru dengan Amazon EC2
- AWS re:Invent 2023 – Penghematan cerdas: Strategi optimisasi biaya Amazon EC2
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Mendalami Sistem Nitro
- AWS re:Invent 2019 – Landasan Amazon EC2

Contoh terkait:

- Kode AWS Compute Optimizer Demo
- Lokakarya Amazon EKS
- Rekomendasi penyesuaian ukuran

PERF02-BP05 Menskalakan sumber daya komputasi Anda secara dinamis

Gunakan elastisitas cloud untuk menaikkan atau menurunkan skala sumber daya komputasi Anda secara dinamis agar sesuai dengan kebutuhan Anda dan hindari kapasitas penyediaan yang berlebihan atau terlalu sedikit untuk beban kerja Anda.

Antipola umum:

- Anda bereaksi pada alarm dengan meningkatkan kapasitas secara manual.
- Anda menggunakan pedoman penyesuaian ukuran yang sama (umumnya infrastruktur statis) seperti di on-premise.
- Anda membiarkan peningkatan kapasitas setelah peristiwa penskalaan, bukannya menurunkan kembali skala.

Manfaat menjalankan praktik terbaik ini: Mengonfigurasi dan menguji elastisitas sumber daya komputasi dapat membantu Anda menghemat dana, mempertahankan tolok ukur kinerja, dan meningkatkan keandalan saat lalu lintas berubah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

AWS memberikan fleksibilitas untuk menaikkan atau menurunkan skala sumber daya Anda secara dinamis melalui berbagai mekanisme penskalaan untuk memenuhi perubahan permintaan. Digabungkan dengan metrik yang terkait dengan komputasi, penskalaan dinamis memungkinkan beban kerja untuk merespons perubahan secara otomatis dan menggunakan rangkaian optimal sumber daya komputasi untuk mencapai tujuannya.

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 penskalaan 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 otomatis menskalakan sesuai rancangan.

Anda harus memastikan bahwa deployment beban kerja dapat menangani peristiwa kenaikan dan penurunan skala.

Langkah implementasi

- Kontainer, fungsi, dan instans komputasi menyediakan mekanisme bagi elastisitas melalui kombinasi dengan penskalaan otomatis atau sebagai fitur layanan. Berikut beberapa contoh mekanisme penskalaan otomatis:

Mekanisme Penskalaan Otomatis	Di mana harus menggunakan
Amazon EC2 Auto Scaling	Untuk memastikan Anda memiliki jumlah yang tepat untuk instans Amazon EC2 yang tersedia guna menangani beban pengguna untuk aplikasi Anda.
Application Auto Scaling	Untuk secara otomatis menskalakan sumber daya bagi layanan AWS individu di luar Amazon EC2, seperti fungsi AWS Lambda atau layanan Amazon Elastic Container Service (Amazon ECS) .
Kubernetes Cluster Autoscaler/Karpenter	Untuk secara otomatis menskalakan klaster Kubernetes.

- Penskalaan sering dibahas terkait dengan layanan komputasi seperti Instans Amazon EC2 atau fungsi AWS Lambda. Pastikan juga untuk mempertimbangkan konfigurasi layanan nonkomputasi seperti AWS Glue untuk mengimbangi permintaan.

- Pastikan metrik untuk penskalaan cocok dengan karakteristik beban kerja yang sedang digunakan. Jika Anda men-deploy aplikasi transkode video, 100% pemanfaatan CPU adalah hal normal dan tidak boleh menjadi metrik primer Anda. Gunakan kedalaman antrean tugas transkode sebagai gantinya. Anda dapat menggunakan metrik yang disesuaikan untuk kebijakan penskalaan Anda jika diperlukan. Untuk memilih metrik yang tepat, pertimbangkan panduan berikut untuk Amazon EC2:
  - Metrik harus merupakan metrik pemanfaatan yang valid dan mendeskripsikan tingkat kesibukan suatu instans.
  - Nilai metrik harus meningkat atau menurun secara proporsional dengan jumlah instans dalam grup Auto Scaling.
- Pastikan Anda menggunakan penskalaan dinamis, bukan penskalaan manual untuk grup Auto Scaling Anda. Sebaiknya gunakan juga kebijakan penskalaan pelacakan target dalam penskalaan dinamis.
- Pastikan deployment beban kerja dapat menangani event penskalaan (naik dan turun). Sebagai contoh, Anda dapat menggunakan Riwayat aktivitas guna memastikan aktivitas penskalaan untuk grup Auto Scaling.
- Evaluasi beban kerja Anda untuk pola terprediksi dan secara proaktif skalakan saat Anda mengantisipasi perubahan terencana dan terprediksi dalam permintaan. Dengan penskalaan prediktif, Anda dapat meniadakan kebutuhan untuk menyediakan kapasitas secara berlebih. untuk detail selengkapnya, lihat Penskalaan Prediktif dengan Amazon EC2 Auto Scaling.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Tipe Instans Amazon EC2
- Kontainer Amazon ECS: Instans Kontainer Amazon ECS
- Kontainer Amazon EKS: Simpul Pekerja Amazon EKS
- Fungsi: Konfigurasi Fungsi Lambda
- Kontrol Status Prosesor untuk Instans Amazon EC2 Anda
- Pendalaman tentang Amazon ECS Cluster Auto Scaling
- Memperkenalkan Karpenter - Kubernetes Cluster Autoscaler Sumber Terbuka Berkinerja Tinggi

Video terkait:

- AWS re:Invent 2023 – AWS Graviton: Performa harga terbaik untuk beban kerja AWS Anda
- AWS re:Invent 2023 – Kemampuan AI generatif Amazon EC2 baru di Konsol Manajemen AWS
- AWS re:Invent 2023 – Apa yang baru dengan Amazon EC2
- AWS re:Invent 2023 – Penghematan cerdas: Strategi optimisasi biaya Amazon EC2
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Mendalami Sistem Nitro
- AWS re:Invent 2019 – Landasan Amazon EC2

Contoh terkait:

- Contoh Grup Amazon EC2 Auto Scaling
- Lokakarya Amazon EKS
- Menskalakan beban kerja Amazon EKS Anda dengan menjalankan di IPv6

PERF02-BP06 Menggunakan akselerator komputasi berbasis perangkat keras yang dioptimalkan

Gunakan akselerator perangkat keras untuk melakukan fungsi tertentu secara lebih efisien daripada alternatif berbasis CPU.

Antipola umum:

- Dalam beban kerja Anda, Anda belum melakukan uji tolok ukur instans tujuan umum dengan instans yang dibuat khusus yang dapat memberikan kinerja lebih tinggi dan biaya lebih rendah.
- Anda menggunakan akselerator komputasi berbasis perangkat keras untuk tugas yang bisa lebih efisien jika menggunakan alternatif berbasis CPU.
- Anda tidak memantau penggunaan GPU.

Manfaat menerapkan praktik terbaik ini: Dengan menggunakan akselerator berbasis perangkat keras, seperti unit pemrosesan grafis (GPU) dan field programmable gate array (FPGA), Anda dapat melakukan fungsi pemrosesan tertentu dengan lebih efisien.

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

Panduan implementasi

Instans komputasi terakselerasi menyediakan akses ke akselerator komputasi berbasis perangkat keras seperti GPU dan FPGA. Akselerator perangkat keras ini menjalankan fungsi-fungsi tertentu seperti pemrosesan grafis atau pencocokan pola data secara lebih efisien daripada alternatif berbasis CPU. Banyak beban kerja yang terakselerasi, seperti perenderan, transkode, dan machine learning, memiliki variabel tinggi sehubungan dengan penggunaan sumber daya. Jalankan perangkat keras ini hanya ketika diperlukan, dan nonaktifkan instans GPU secara otomatis saat tidak diperlukan untuk meningkatkan keseluruhan efisiensi kinerja.

Langkah implementasi

- Identifikasi instans komputasi terakselerasi mana yang dapat menangani persyaratan 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. Instans AWS 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. Sebagai contoh, Anda dapat menggunakan agen CloudWatch untuk mengumpulkan metrik-metrik seperti `utilization_gpu` dan `utilization_memory` untuk GPU Anda sebagaimana 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.
  - Optimalkan 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:

- Bekerja dengan GPU di Amazon Elastic Container Service

- Instans GPU
- Instans dengan AWS Trainium
- Instans dengan AWS Inferentia
- Mari Merancang! Merancang dengan chip dan akselerator kustom

- Komputasi Terakselerasi
- Instans Amazon EC2 VT1
- Bagaimana cara memilih tipe instans Amazon EC2 yang tepat untuk beban kerja saya?
- 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 Elastic Compute Cloud untuk deep learning
- AWS re:Invent 2022 - [PELUNCURAN BARU!] Memperkenalkan instans Amazon EC2 Inf2 berbasis AWS Inferentia2
- AWS re:Invent 2022 - Percepat deep learning dan berinovasi lebih cepat dengan AWS Trainium
- AWS re:Invent 2022 - Deep learning di AWS dengan NVIDIA: Dari pelatihan hingga deployment

Contoh terkait:

- Amazon SageMaker dan NVIDIA GPU Cloud (NGC)
- Gunakan SageMaker dengan Trainium dan Inferentia untuk beban kerja pelatihan dan inferensi deep learning yang dioptimalkan
- Mengoptimalkan model NLP dengan instans Amazon Elastic Compute Cloud Inf1 di Amazon SageMaker

Manajemen Data

Solusi manajemen data yang optimal untuk sistem tertentu bervariasi berdasarkan jenis data (blok, file, atau objek), pola akses (acak atau berurutan), throughput yang diperlukan, frekuensi akses (online, offline, arsip), frekuensi pembaruan (WORM, dinamis), dan ketersediaan serta batasan daya tahan. Beban kerja Well-Architected menggunakan penyimpanan data yang dibuat khusus yang memungkinkan berbagai fitur untuk meningkatkan kinerja.

Area fokus ini berbagi panduan dan praktik terbaik untuk mengoptimalkan penyimpanan data, pergerakan dan pola akses, serta efisiensi kinerja penyimpanan data.

Praktik terbaik

- PERF03-BP01 Menggunakan penyimpanan data yang dibuat khusus yang paling mendukung persyaratan akses dan penyimpanan data Anda
- PERF03-BP02 Evaluasi opsi konfigurasi yang tersedia untuk penyimpanan data
- PERF03-BP03 Mengumpulkan dan merekam metrik kinerja penyimpanan data
- PERF03-BP04 Menerapkan strategi untuk meningkatkan kinerja kueri di penyimpanan data
- PERF03-BP05 Mengimplementasikan pola akses data yang memanfaatkan caching

PERF03-BP01 Menggunakan penyimpanan data yang dibuat khusus yang paling mendukung persyaratan akses dan penyimpanan data Anda

Pahami karakteristik data (seperti dapat dibagikan, ukuran, ukuran cache, pola akses, latensi, throughput, dan persistensi data) untuk memilih penyimpanan data khusus (penyimpanan atau basis data) yang tepat untuk beban kerja Anda.

Antipola umum:

- Anda bertahan dengan satu solusi basis data disebabkan pengetahuan dan pengalaman internal tentang satu jenis solusi basis data tertentu.
- Anda berasumsi bahwa semua beban kerja memiliki persyaratan penyimpanan dan akses data yang serupa.
- Anda belum mengimplementasikan katalog data untuk menginventarisasi aset data Anda.

Manfaat menjalankan praktik terbaik ini: Dengan memahami karakteristik dan persyaratan data, Anda dapat menentukan teknologi penyimpanan yang paling efisien dan berkinerja paling tinggi sesuai dengan kebutuhan beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Saat memilih dan menerapkan penyimpanan data, pastikan karakteristik penyimpanan, kueri, dan penskalaan mendukung persyaratan data beban kerja. AWS menyediakan banyak teknologi penyimpanan data dan basis data termasuk penyimpanan blok, penyimpanan objek, penyimpanan streaming, sistem file, relasional, nilai kunci, dokumen, penyimpanan dalam memori, grafik, deret waktu, dan basis data buku besar. Setiap solusi manajemen data memiliki opsi dan konfigurasi yang tersedia bagi Anda untuk mendukung kasus penggunaan dan model data Anda. Dengan memahami karakteristik dan persyaratan data, Anda dapat melepaskan diri dari teknologi penyimpanan monolitik dan pendekatan satu-untuk-semua yang terbatas guna memfokuskan diri pada manajemen data yang tepat.

Langkah implementasi

- Lakukan inventaris berbagai jenis data yang ada dalam beban kerja Anda.
- Pahami dan dokumentasikan karakteristik serta persyaratan data, termasuk:
  - Tipe data (tidak terstruktur, semi-terstruktur, relasional)
  - Volume dan pertumbuhan data
  - Ketahanan data: persisten, sementara, transien
  - Persyaratan ACID (atomisitas, konsistensi, isolasi, durabilitas)
  - Pola akses data (intensif baca atau intensif tulis)
  - Latensi
  - Throughput
  - IOPS (operasi input/output per detik)
  - Periode retensi data
- Pelajari berbagai tempat penyimpanan data (layanan penyimpanan dan basis data) yang tersedia untuk beban kerja Anda di AWS yang dapat memenuhi karakteristik data Anda (sebagaimana diuraikan dalam PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia. Berikut adalah beberapa contoh teknologi penyimpanan serta karakteristik utamanya:

Type (Jenis)	Layanan AWS	Karakteristik utama
Object storage	Amazon S3	Unlimited scalability, high availability, and multiple options for accessibility. Transferring and accessing objects in and out of Amazon S3 can use a service, such as Transfer Acceleration or Access Points, to support your location, security needs, and access patterns.
Archiving storage	Amazon S3 Glacier	Built for data archiving.
Streaming storage	Amazon Kinesis Amazon Managed Streaming for Apache Kafka (Amazon MSK)	Efficient ingestion and storage of streaming data.
Shared file system	Amazon Elastic File System (Amazon EFS)	Sistem file yang dapat dipasang yang dapat diakses oleh berbagai jenis solusi komputasi.
Shared file system	Amazon FSx	Built on the latest AWS compute solutions to support four commonly used file systems: NetApp ONTAP, OpenZFS, Windows File Server, and Lustre. Amazon FSx latensi, throughput, dan IOPS vary per file system and should be considered when selecting the right file system for your workload needs.

Type (Jenis)	Layanan AWS	Karakteristik utama
Block storage	Amazon Elastic Block Store (Amazon EBS)	Scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS includes SSD-backed storage for transactional, IOPS-intensive workloads and HDD-backed storage for throughput-intensive workloads.
Relational database	Amazon Aurora, Amazon RDS, Amazon Redshift.	Designed to support ACID (atomicity, consistency, isolation, durability) transactions, and maintain referential integrity and strong data consistency. Many traditional applications, enterprise resource planning (ERP), customer relationship management (CRM), and ecommerce use relational databases to store their data.
Key-value database	Amazon DynamoDB	Optimized for common access patterns, typically to store and retrieve large volumes of data. High-traffic web apps, ecommerce systems, and gaming applications are typical use-cases for key-value databases.

Type (Jenis)	Layanan AWS	Karakteristik utama
Document database	Amazon DocumentDB	Designed to store semi-structured data as JSON-like documents. These databases help developers build and update applications such as content management, catalogs, and user profiles quickly.
In-memory database	Amazon ElastiCache , Amazon MemoryDB for Redis	Used for applications that require real-time access to data, lowest latency and highest throughput. You may use in-memory databases for application caching, session management, gaming leaderboards, low latency ML feature store, microservices messaging system, and a high-throughput streaming mechanism
Graph database	Amazon Neptune	Used for applications that must navigate and query millions of relationships between highly connected graph datasets with millisecond latency at large scale. Many companies use graph databases for fraud detection , social networking, and recommendation engines.

Type (Jenis)	Layanan AWS	Karakteristik utama
Time Series database	Amazon Timestream	Used to efficiently collect, synthesize, and derive insights from data that changes over time. IoT applications, DevOps, and industrial telemetry can utilize time-series databases.
Wide column	Amazon Keyspaces (untuk Apache Cassandra)	Uses tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table. You typically see a wide column store in high scale industrial apps for equipment maintenance, fleet management, and route optimization.
Ledger	Amazon Quantum Ledger Database (Amazon QLDB)	Provides a centralized and trusted authority to maintain a scalable, immutable, and cryptographically verifiabl e record of transactions for every application. We see ledger databases used for systems of record, supply chain, registrations, and even banking transactions.

- Jika Anda sedang membangun platform data, manfaatkan arsitektur data modern di AWS untuk mengintegrasikan danau data, gudang data, dan penyimpanan data yang dibuat khusus.

- Pertanyaan kunci yang perlu Anda pertimbangkan saat memilih penyimpanan data untuk beban kerja Anda adalah sebagai berikut:

Question	Things to consider
How is the data structured?	• Jika data tidak terstruktur, pertimbangkan penyimpanan objek seperti Amazon S3 atau basis data NoSQL seperti Amazon DocumentDB • Untuk data nilai kunci, pertimbangkan DynamoDB, Amazon ElastiCache for Redis atau Amazon MemoryDB for Redis
What level of referential integrity is required?	• Untuk kendala utama asing, basis data relasional seperti Amazon RDS dan Aurora dapat memberikan tingkat integritas ini. • Biasanya, dalam model data NoSQL, Anda akan melakukan denormalisasi data menjadi satu dokumen atau kumpulan dokumen untuk diambil dalam satu permintaan dan bukannya digabungkan dalam berbagai dokumen atau tabel.
Is ACID (atomicity, consistency, isolation, durability) compliance required?	• Jika diperlukan sifat ACID yang terkait dengan basis data relasional, pertimban gkan basis data relasional seperti Amazon RDS dan Aurora. • Jika diperlukan konsistensi tinggi untuk basis data NoSQL, Anda dapat menggunak an bacaan sangat konsisten dengan DynamoDB.

Question	Things to consider
How will the storage requirements change over time? How does this impact scalability?	• Basis data nirserver seperti DynamoDB dan Amazon Quantum Ledger Database (Amazon QLDB) akan menskalakan secara dinamis. • Basis data relasional memiliki batas atas terkait penyimpanan yang tersedia, dan sering kali harus dipartisi secara horizontal menggunakan mekanisme seperti serpihan setelah penyimpanan tersebut mencapai batas ini.
What is the proportion of read queries in relation to write queries? Would caching be likely to improve performance?	• Beban kerja intensif baca dapat diuntungkan dari lapisan caching, seperti ElastiCache atau DAX jika basis datanya adalah DynamoDB. • Bacaan juga dapat dilimpahkan ke replika baca dengan basis data relasional seperti Amazon RDS.
Does storage and modification (OLTP - Online Transaction Processing) or retrieval and reporting (OLAP - Online Analytical Processing) have a higher priority?	• Untuk pemrosesan transaksional baca apa adanya throughput tinggi, pertimbangkan basis data NoSQL seperti DynamoDB. • Untuk throughput tinggi dan pola baca yang kompleks (seperti join) dengan konsistensi, gunakan Amazon RDS. • Untuk kueri analitis, pertimbangkan basis data kolom seperti Amazon Redshift atau mengekspor data ke Amazon S3 dan melakukan analitik menggunakan Athena atau Amazon QuickSight.

Question	Things to consider
What level of durability does the data require?	• Aurora secara otomatis mereplikasi data Anda di tiga Zona Ketersediaan dalam satu Wilayah, yang artinya data Anda sangat tahan lama dengan lebih sedikit kemungkin an hilangnya data. • DynamoDB secara otomatis direplikasi di beberapa Zona Ketersediaan, memberikan durabilitas data dan ketersediaan tinggi. • Amazon S3 memberikan 11 sembilan durabilitas. Banyak layanan basis data seperti Amazon RDS dan DynamoDB mendukung ekspor data ke Amazon S3 untuk pengarsipan dan retensi jangka panjang.
Is there a desire to move away from commercial database engines or licensing costs?	• Pertimbangkan mesin sumber terbuka seperti PostgreSQL dan MySQL di Amazon RDS atau Aurora. • Manfaatkan AWS Database Migration Service dan AWS Schema Conversion Tool untuk melakukan migrasi dari mesin basis data komersial ke sumber terbuka
What is the operational expectation for the database? Is moving to managed services a primary concern?	• Pemanfaatan Amazon RDS sebagai ganti Amazon EC2, dan DynamoDB atau Amazon DocumentDB sebagai ganti hosting mandiri basis data NoSQL dapat mengurangi biaya tambahan operasional.

Question	Things to consider
How is the database currently accessed? Is it only application access, or are there business intelligence (BI) users and other connected off-the-shelf applications?	• Jika Anda memiliki ketergantungan pada alat eksternal maka Anda mungkin harus mempertahankan kompatibilitas dengan basis data yang didukungnya. Amazon RDS sepenuhnya kompatibel dengan berbagai versi mesin yang didukungnya, termasuk Microsoft SQL Server, Oracle, MySQL, dan PostgreSQL.

- Lakukan uji coba dan uji tolok ukur di lingkungan nonproduksi untuk mengidentifikasi penyimpanan data mana yang paling sesuai dengan kebutuhan beban kerja Anda.

Sumber daya

Dokumen terkait:

- Tipe Volume Amazon EBS
- Penyimpanan Amazon EC2
- Amazon EFS: Kinerja Amazon EFS
- Kinerja Amazon FSx for Lustre
- Kinerja Amazon FSx for Windows File Server
- Amazon S3 Glacier: Dokumentasi S3 Glacier
- Amazon S3: Pertimbangan Tingkat Permintaan dan Kinerja
- Penyimpanan Cloud dengan AWS
- Karakteristik I/O Amazon EBS
- Basis Data Cloud dengan AWS
- Caching Basis Data AWS
- DynamoDB Accelerator
- Praktik Terbaik Amazon Aurora
- Kinerja Amazon Redshift
- 10 kiat kinerja terbaik Amazon Athena

- Praktik terbaik Amazon Redshift Spectrum
- Praktik terbaik Amazon DynamoDB
- Pilih antara Amazon EC2 dan Amazon RDS
- Praktik Terbaik untuk Mengimplementasikan Amazon ElastiCache

Video terkait:

- AWS re:Invent 2023: Tingkatkan efisiensi Amazon Elastic Block Store dan menjadi lebih hemat biaya
- AWS re:Invent 2023: Mengoptimalkan harga dan kinerja penyimpanan dengan Amazon Simple Storage Service
- AWS re:Invent 2023: Membangun dan mengoptimalkan danau data di Amazon Simple Storage Service
- AWS re:Invent 2022: Membangun arsitektur data modern di AWS
- 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 lanjutan dengan Amazon DynamoDB
- AWS re:Invent 2022: Lakukan modernisasi aplikasi dengan basis data yang dibuat khusus
- Pendalaman Amazon DynamoDB: Pola desain lanjutan

Contoh terkait:

- Lokakarya Basis Data yang Dirancang Khusus AWS
- Basis Data untuk Developer
- Hari Imersi Arsitektur Data Modern AWS
- Bangun Jala Data di AWS
- Contoh Amazon S3
- Mengoptimalkan Pola Data menggunakan Pembagian Data Amazon Redshift
- Migrasi Basis Data
- MS SQL Server - Demo Replikasi AWS Database Migration Service (AWS DMS)
- Lokakarya Praktik Langsung Modernisasi Basis Data
- Sampel Amazon Neptune

PERF03-BP02 Evaluasi opsi konfigurasi yang tersedia untuk penyimpanan data

Pahami dan evaluasi berbagai fitur dan opsi konfigurasi yang tersedia untuk penyimpanan data Anda guna mengoptimalkan ruang penyimpanan dan kinerja untuk beban kerja Anda.

Antipola umum:

- Anda hanya menggunakan satu jenis penyimpanan, seperti Amazon EBS, untuk semua beban kerja.
- Anda menggunakan IOPS yang tersedia untuk semua beban kerja tanpa pengujian dunia nyata terhadap semua tingkat penyimpanan.
- Anda tidak memahami opsi konfigurasi pada solusi manajemen data yang Anda pilih.
- Anda hanya mengandalkan peningkatan ukuran instans tanpa mempertimbangkan opsi konfigurasi lain yang tersedia.
- Anda tidak menguji karakteristik penskalaan penyimpanan data Anda.

Manfaat menerapkan praktik terbaik ini: Dengan menjelajahi dan mencoba berbagai konfigurasi penyimpanan data, Anda mungkin dapat mengurangi biaya infrastruktur, meningkatkan kinerja, serta mengurangi upaya pengelolaan beban kerja.

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

Panduan implementasi

Beban kerja dapat memiliki satu atau beberapa penyimpanan data yang digunakan berdasarkan persyaratan penyimpanan dan akses data. Untuk mengoptimalkan biaya dan efisiensi kinerja, Anda harus mengevaluasi pola akses data guna menentukan konfigurasi penyimpanan data yang sesuai. Saat mencoba opsi-opsi penyimpanan data, pertimbangkan beberapa aspek seperti opsi penyimpanan, memori, komputasi, replika baca, persyaratan konsistensi, pooling koneksi, dan opsi cache. Coba beberapa opsi konfigurasi ini untuk meningkatkan metrik efisiensi kinerja.

Langkah implementasi

- Pahami konfigurasi saat ini (seperti tipe instans, ukuran penyimpanan, atau versi mesin basis data) penyimpanan data Anda.

- Tinjau dokumentasi dan praktik terbaik AWS untuk mempelajari rekomendasi opsi konfigurasi yang dapat membantu meningkatkan kinerja penyimpanan data Anda. Berikut ini adalah opsi-opsi penyimpanan data utama yang perlu dipertimbangkan:

Configuration option	Examples
Offloading reads (like read replicas and caching)	• Untuk tabel DynamoDB, Anda dapat meringankan beban baca menggunakan DAX untuk caching. • Anda dapat membuat klaster Amazon ElastiCache for Redis dan mengonfig urasikan aplikasi Anda untuk membaca dari cache terlebih dahulu, kembali ke basis data jika item yang diminta tidak tersedia. • Basis data relasional seperti Amazon RDS dan Aurora, serta basis data NoSQL yang tersedia seperti Neptune dan Amazon DocumentDB semua mendukung penambahan replika baca untuk mengurang i porsi baca beban kerja. • Basis data nirserver seperti DynamoDB akan menskalakan secara otomatis. Pastikan unit kapasitas baca (RSU) yang tersedia cukup untuk mengatasi beban kerja.

Configuration option	Examples
Scaling writes (like partition key sharding or introducing a queue)	• Untuk basis data relasional, Anda dapat memperbesar ukuran instans untuk mengakomodasi tambahan beban kerja atau menambah IOPS yang tersedia untuk memfasilitasi kenaikan throughput pada penyimpanan yang mendasari. • Anda juga dapat membuat antrean di depan basis data, bukan menulis secara langsung ke basis data. Dengan pola ini, Anda dapat memisahkan penyerapan dari basis data dan mengontrol tingkat aliran, sehingga basis data tidak kewalahan. • Mengganti pembuatan transaksi berdurasi pendek dengan pembuatan batch permintaa n penulisan dapat membantu meningkat kan throughput dalam basis data relasional dengan volume penulisan tinggi. • Basis data nirserver seperti DynamoDB dapat menskalakan throughput tulis secara otomatis atau dengan menyesuaikan unit kapasitas tulis (WCU) yang tersedia, bergantung pada mode kapasitasnya. • Anda tetap dapat menjumpai masalah dengan partisi panas ketika Anda mencapai batas throughput pada kunci partisi tertentu. Hal ini dapat dikurangi dengan memilih distribusi kunci partisi yang lebih merata atau dengan memisah penulisan kunci partisi.

Configuration option	Examples
Policies to manage the lifecycle of your datasets	• Anda dapat menggunakan Siklus Hidup Amazon S3 untuk mengelola objek-objek Anda di sepanjang siklus hidupnya. Jika pola akses Anda tidak diketahui, berubah-ubah, atau tidak dapat diprediksi, Anda dapat menggunakan Amazon S3 Intelligent-Tiering, yang memantau pola akses dan secara otomatis memindahkan objek yang belum diakses ke tingkat akses dengan biaya lebih rendah. Anda dapat memanfaatkan metrik Lensa Penyimpanan Amazon S3 untuk mengidentifikasi peluang dan celah pengoptimalan dalam manajemen siklus hidup. • Manajemen siklus hidup Amazon EFS secara otomatis mengelola penyimpanan file untuk sistem file Anda.
Connection management and pooling	• Proksi Amazon RDS dapat digunakan dengan Amazon RDS dan Aurora untuk mengelola koneksi ke basis data. • Basis data nirserver seperti DynamoDB tidak terkait dengan koneksi apa pun, tetapi pertimbangkan kapasitas yang tersedia atau kebijakan penskalaan otomatis untuk mengatasi lonjakan beban.

- Lakukan uji coba dan uji tolok ukur di lingkungan nonproduksi untuk mengidentifikasi opsi konfigurasi mana yang dapat memenuhi persyaratan beban kerja Anda.
- Setelah bereksperimen, rencanakan migrasi dan validasikan metrik kinerja Anda.
- Gunakan alat pemantauan AWS (seperti Amazon CloudWatch) dan optimisasi (seperti Lensa Penyimpanan Amazon S3) untuk terus mengoptimalkan penyimpanan data Anda menggunakan pola penggunaan dunia nyata.

Sumber daya

Dokumen terkait:

- Penyimpanan Cloud dengan AWS
- Tipe Volume Amazon EBS
- Penyimpanan Amazon EC2
- Amazon EFS: Kinerja Amazon EFS
- Kinerja Amazon FSx for Lustre
- Kinerja Amazon FSx for Windows File Server
- Amazon S3 Glacier: Dokumentasi S3 Glacier
- Amazon S3: Pertimbangan Tingkat Permintaan dan Kinerja
- Karakteristik I/O Amazon EBS
- Basis Data Cloud dengan AWS
- Caching Basis Data AWS
- DynamoDB Accelerator
- Praktik Terbaik Amazon Aurora
- Kinerja Amazon Redshift
- 10 kiat kinerja terbaik Amazon Athena
- Praktik terbaik Amazon Redshift Spectrum
- Praktik terbaik Amazon DynamoDB

Video terkait:

- AWS re:Invent 2023: Tingkatkan efisiensi Amazon Elastic Block Store dan menjadi lebih hemat biaya
- AWS re:Invent 2023: Optimalkan harga dan kinerja penyimpanan dengan Amazon Simple Storage Service
- AWS re:Invent 2023: Membangun dan mengoptimalkan danau data di Amazon Simple Storage Service
- AWS re:Invent 2023: Apa yang baru dengan penyimpanan file AWS
- AWS re:Invent 2023: Memahami Amazon DynamoDB

Contoh terkait:

- Lokakarya Basis Data yang Dirancang Khusus AWS
- Basis Data untuk Developer
- Hari Imersi Arsitektur Data Modern AWS
- Penskalaan Otomatis Amazon EBS
- Contoh Amazon S3
- Contoh Amazon DynamoDB
- Sampel migrasi Basis Data AWS
- Lokakarya Modernisasi Basis Data
- Menggunakan parameter di Amazon RDS for Postgress DB

PERF03-BP03 Mengumpulkan dan merekam metrik kinerja penyimpanan data

Lacak dan rekam metrik kinerja yang relevan untuk penyimpanan data Anda guna memahami kinerja solusi manajemen data Anda. Metrik-metrik ini dapat membantu Anda mengoptimalkan penyimpanan data Anda, memastikan terpenuhinya persyaratan beban kerja Anda, dan memberikan gambaran umum yang jelas tentang kinerja beban kerja.

Antipola umum:

- Anda hanya menggunakan pencarian file log manual untuk metrik.
- Anda hanya memublikasikan metrik ke alat-alat internal yang digunakan tim Anda dan tidak memiliki gambaran yang komprehensif tentang beban kerja Anda.
- Anda hanya menggunakan metrik default yang dicatat oleh perangkat lunak pemantauan Anda yang dipilih.
- Anda hanya meninjau metrik ketika terdapat masalah.
- Anda hanya memantau metrik tingkat sistem dan tidak merekam metrik akses atau penggunaan data.

Manfaat menjalankan praktik terbaik ini: Memiliki dasar acuan kinerja membantu Anda memahami perilaku normal dan persyaratan beban kerja. Pola abnormal dapat diidentifikasi dan diperbaiki lebih cepat sehingga meningkatkan kinerja dan keandalan penyimpanan data.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Untuk memantau kinerja penyimpanan data, Anda harus merekam beberapa metrik kinerja secara berkala. Dengan begitu Anda dapat mendeteksi anomali serta mengukur kinerja berdasarkan metrik bisnis untuk memastikan kebutuhan beban kerja Anda terpenuhi.

Metrik harus menyertakan metrik sistem dasar yang mendukung penyimpanan data serta metrik basis data. Metrik sistem dasar dapat meliputi metrik pemanfaatan CPU, memori, penyimpanan disk yang tersedia, I/O disk, rasio cache hit, dan jaringan masuk serta keluar, sedangkan metrik basis data dapat meliputi transaksi per detik, kueri teratas, rata-rata laju kueri, waktu respons, penggunaan indeks, penguncian tabel, batas waktu kueri, dan jumlah koneksi yang terbuka. Data ini sangat penting untuk mengetahui kinerja beban kerja dan bagaimana solusi manajemen data digunakan. Gunakan metrik ini sebagai bagian dari pendekatan berbasis data yang digunakan untuk mengatur dan mengoptimalkan sumber daya beban kerja Anda.

Gunakan alat, pustaka, dan sistem yang merekam pengukuran kinerja terkait kinerja basis data.

Langkah implementasi

1. Identifikasi metrik kinerja utama yang perlu dilacak oleh penyimpanan data Anda.
    a. Metrik dan dimensi Amazon S3
    b. Memantau metrik untuk instans Amazon RDS
    c. Memantau beban DB dengan Wawasan Performa di Amazon RDS
    d. Ikhtisar Pemantauan yang Ditingkatkan
    e. Metrik dan dimensi DynamoDB
    f. Memantau DynamoDB Accelerator
    g. Memantau Amazon MemoryDB for Redis dengan Amazon CloudWatch
    h. Metrik Mana yang Harus Saya Pantau?
    i. Memantau kinerja klaster Amazon Redshift
    j. Metrik dan dimensi Timestream
    k. Metrik Amazon CloudWatch untuk Amazon Aurora
    l. Pencatatan log dan pemantauan di Amazon Keyspaces (for Apache Cassandra)
    m. Memantau Sumber Daya Amazon Neptune

2. Gunakan solusi pencatatan log dan pemantauan yang disetujui untuk mengumpulkan metrik ini. Amazon CloudWatch dapat mengumpulkan metrik di seluruh sumber daya dalam arsitektur Anda. Anda juga dapat mengumpulkan dan memublikasikan metrik kustom untuk memunculkan metrik turunan (derived metric) atau bisnis. Gunakan CloudWatch atau solusi pihak ketiga untuk menetapkan alarm yang memberikan indikasi saat ambang batas terlampaui.
3. Periksa apakah pemantauan penyimpanan data dapat terbantu dengan solusi machine learning yang mendeteksi anomali kinerja.
   a. Amazon DevOps Guru untuk Amazon RDS menyediakan visibilitas masalah kinerja dan memberikan saran tindakan perbaikan.
4. Konfigurasikan retensi data dalam solusi pemantauan dan pencatatan log Anda agar sesuai dengan tujuan keamanan dan operasional Anda.
   a. Retensi data default untuk metrik CloudWatch
   b. Retensi data default untuk CloudWatch Logs

Sumber daya

Dokumen terkait:

- Caching Basis Data AWS
- 10 kiat performa terbaik Amazon Athena
- Praktik terbaik Amazon Aurora
- DynamoDB Accelerator
- Praktik terbaik Amazon DynamoDB
- Praktik terbaik Amazon Redshift Spectrum
- Performa Amazon Redshift
- Basis Data Cloud dengan AWS
- Wawasan Kinerja Amazon RDS

Video terkait:

- AWS re:Invent 2022 - Pemantauan kinerja dengan Amazon RDS dan Aurora, bersama Autodesk
- Pemantauan dan Penyetelan Kinerja Basis Data dengan Amazon DevOps Guru for Amazon RDS
- AWS re:Invent 2023 - Apa yang baru dengan penyimpanan file AWS

- AWS re:Invent 2023 - Memahami Amazon DynamoDB
- AWS re:Invent 2023 - Membangun dan mengoptimalkan danau data di Amazon S3
- AWS re:Invent 2023 - Apa yang baru dengan penyimpanan file AWS
- AWS re:Invent 2023 - Memahami Amazon DynamoDB
- Praktik Terbaik untuk Memantau Beban Kerja Redis di Amazon ElastiCache

Contoh terkait:

- Kerangka Kerja Pengumpulan Metrik Penyerapan Set Data AWS
- Lokakarya Pemantauan Amazon RDS
- Lokakarya Basis Data yang Dirancang Khusus AWS

PERF03-BP04 Menerapkan strategi untuk meningkatkan kinerja kueri di penyimpanan data

Terapkan strategi untuk mengoptimalkan data dan meningkatkan kueri data untuk memungkinkan skalabilitas yang lebih besar dan kinerja yang efisien untuk beban kerja Anda.

Antipola umum:

- Anda tidak mempartisi data di penyimpanan data Anda.
- Anda menyimpan data hanya dalam satu format file di penyimpanan data Anda.
- Anda tidak menggunakan indeks di penyimpanan data Anda.

Manfaat menjalankan praktik terbaik ini: Optimisasi data dan performa kueri menghasilkan efisiensi yang lebih tinggi, biaya lebih rendah, dan pengalaman pengguna yang lebih baik.

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

Panduan implementasi

Pengoptimalan data dan penyetelan kueri adalah aspek penting efisiensi kinerja di penyimpanan data, karena berdampak pada kinerja dan responsivitas seluruh beban kerja cloud. Kueri yang tidak dioptimalkan dapat menghasilkan penggunaan dan kemacetan sumber daya yang lebih besar, yang mengurangi keseluruhan efisiensi sebuah penyimpanan data.

Pengoptimalan data mencakup beberapa teknik untuk memastikan penyimpanan dan akses data yang efisien. Hal ini juga membantu meningkatkan performa kueri di penyimpanan data. Strategi utama mencakup partisi data, kompresi data, dan denormalisasi data, yang membantu data dioptimalkan untuk penyimpanan dan akses.

Langkah implementasi

- Pahami dan analisis kueri data penting yang dilakukan di penyimpanan data Anda.
- Identifikasi kueri lambat di penyimpanan data Anda dan gunakan rencana kueri untuk memahami statusnya saat ini.
  - Menganalisis rencana kueri di Amazon Redshift
  - Menggunakan EXPLAIN dan EXPLAIN ANALYZE di Athena
- Terapkan strategi untuk meningkatkan kinerja kueri. Beberapa strategi utamanya meliputi:
  - Menggunakan format file kolom (seperti Parquet atau ORC).
  - Mengompresi data di penyimpanan data untuk mengurangi ruang penyimpanan dan operasi I/O.
  - Partisi data untuk membagi data menjadi bagian-bagian yang lebih kecil dan mengurangi waktu pemindaian data.
    - Mempartisi data di Athena
    - Partisi dan distribusi data
  - Pengindeksan data pada kolom umum dalam kueri.
  - Gunakan tampilan terwujud untuk kueri yang sering.
    - Memahami tampilan terwujud
    - Membuat tampilan terwujud di Amazon Redshift
  - Pilih operasi gabungan yang tepat untuk kueri. Saat Anda menggabungkan dua tabel, tentukan tabel yang lebih besar di sisi kiri gabungan dan tabel yang lebih kecil di sisi kanan gabungan.
  - Solusi caching terdistribusi untuk meningkatkan latensi dan mengurangi jumlah operasi I/O basis data.
  - Pemeliharaan rutin seperti mengeksekusi statistik.
- Lakukan eksperimen dan uji strategi di lingkungan nonproduksi.

Sumber daya

Dokumen terkait:

- Praktik terbaik Amazon Aurora
- Performa Amazon Redshift
- 10 kiat performa terbaik Amazon Athena
- Caching Basis Data AWS
- Praktik Terbaik untuk Mengimplementasikan Amazon ElastiCache
- Mempartisi data di Athena

Video terkait:

- AWS re:Invent 2023 - Praktik terbaik optimisasi biaya penyimpanan AWS
- AWS re:Invent 2022 - Pemantauan kinerja dengan Amazon RDS dan Aurora, bersama Autodesk
- Mengoptimalkan Kueri Amazon Athena dengan Alat Analisis Kueri Baru

Contoh terkait:

- Amazon S3 Select - Mengkueri data tanpa server atau basis data
- Lokakarya Basis Data yang Dirancang Khusus AWS

PERF03-BP05 Mengimplementasikan pola akses data yang memanfaatkan caching

Implementasikan pola akses yang dapat memanfaatkan caching data untuk pengambilan cepat data yang sering diakses.

Antipola umum:

- Anda menyimpan cache data yang sering berubah.
- Anda mengandalkan data dalam cache seolah-olah data tersebut disimpan dengan durabilitas tinggi dan selalu tersedia.
- Anda tidak mempertimbangkan konsistensi data cache Anda.
- Anda tidak memantau efisiensi implementasi caching Anda.

Manfaat menjalankan praktik terbaik ini: Menyimpan data dalam cache dapat meningkatkan latensi baca, throughput baca, pengalaman pengguna, dan efisiensi secara keseluruhan, serta mengurangi biaya.

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

Panduan implementasi

Cache adalah komponen perangkat lunak atau perangkat keras yang dimaksudkan untuk menyimpan data sehingga permintaan di masa mendatang untuk data yang sama dapat dilayani lebih cepat atau lebih efisien. Data yang disimpan dalam cache dapat direkonstruksi jika hilang dengan mengulangi perhitungan sebelumnya atau mengambilnya dari tempat penyimpanan data lain.

Caching data dapat menjadi salah satu strategi paling efektif untuk meningkatkan performa aplikasi Anda secara keseluruhan dan mengurangi beban pada sumber data primer yang mendasarinya. Data dapat di-cache di berbagai tingkatan dalam aplikasi, seperti di dalam aplikasi yang membuat panggilan jarak jauh, yang dikenal sebagai caching sisi klien, atau dengan menggunakan layanan sekunder cepat untuk menyimpan data, yang dikenal sebagai caching jarak jauh.

Caching sisi klien

Dengan caching sisi klien, setiap klien (aplikasi atau layanan yang mengkueri penyimpanan data backend) dapat menyimpan hasil kueri unik mereka secara lokal selama jangka waktu tertentu. Hal ini dapat mengurangi jumlah permintaan di seluruh jaringan ke sebuah penyimpanan data dengan memeriksa cache klien lokal terlebih dahulu. Jika hasilnya tidak ada, aplikasi kemudian dapat mengkueri penyimpanan data tersebut dan menyimpan hasilnya secara lokal. Dengan pola ini, setiap klien dapat menyimpan data di lokasi terdekat (klien itu sendiri), sehingga menghasilkan latensi serendah mungkin. Klien juga dapat terus melayani beberapa kueri ketika penyimpanan data backend tidak tersedia, sehingga meningkatkan ketersediaan sistem secara keseluruhan.

Salah satu kelemahan pendekatan ini adalah ketika ada beberapa klien yang dilibatkan, semuanya dapat menyimpan data cache yang sama secara lokal. Hal ini mengakibatkan penggunaan penyimpanan duplikat dan inkonsistensi data antara klien-klien tersebut. Salah satu klien mungkin melakukan caching hasil kueri, dan satu menit kemudian klien lainnya dapat menjalankan kueri yang sama dan mendapatkan hasil yang berbeda.

Caching jarak jauh

Untuk mengatasi masalah duplikat data antarklien, layanan eksternal yang cepat, atau cache jarak jauh, dapat digunakan untuk menyimpan data yang dikueri. Alih-alih memeriksa penyimpanan

data lokal, setiap klien akan memeriksa cache jarak jauh sebelum mengkueri penyimpanan data backend. Strategi ini memungkinkan respons yang lebih konsisten antarklien, efisiensi yang lebih baik pada data yang disimpan, dan volume data cache yang lebih tinggi karena ruang penyimpanannya diskalakan tanpa terikat klien.

Kelemahan cache jarak jauh adalah keseluruhan sistem mungkin mengalami latensi yang lebih tinggi karena diperlukan lompatan jaringan tambahan untuk memeriksa cache jarak jauh. Caching sisi klien dapat digunakan bersama caching jarak jauh untuk caching multitingkat sehingga dapat memperbaiki latensi.

Langkah implementasi

1. Identifikasikan basis data, API, dan layanan jaringan yang dapat memanfaatkan caching. Layanan yang memiliki beban kerja baca yang berat, memiliki rasio baca-tulis yang tinggi, atau mahal untuk diskalakan dapat memanfaatkan caching.
   - Caching Basis Data
   - Mengaktifkan caching API untuk meningkatkan responsivitas
2. Identifikasikan jenis strategi caching yang tepat yang paling sesuai dengan pola akses Anda.
   - Strategi caching
   - Solusi Caching AWS
3. Ikuti Praktik Terbaik Caching untuk penyimpanan data Anda.
4. Konfigurasikan strategi pembatalan cache, seperti time-to-live (TTL), untuk semua data yang menyeimbangkan kesegaran data dan mengurangi tekanan pada penyimpanan data backend.
5. Aktifkan fitur seperti percobaan ulang koneksi otomatis, mundur eksponensial, batas waktu sisi klien, dan pooling koneksi di dalam klien, jika tersedia, karena fitur-fitur tersebut dapat meningkatkan performa dan keandalan.
   - Praktik terbaik: Klien Redis dan Amazon ElastiCache for Redis
6. Pantau laju hit cache dengan target 80% atau lebih tinggi. Nilai yang lebih rendah mungkin menunjukkan ukuran cache yang tidak mencukupi atau pola akses yang tidak diuntungkan dengan caching.
   - Metrik mana yang harus saya pantau?
   - Praktik terbaik untuk memantau beban kerja Redis di Amazon ElastiCache
   - Praktik terbaik pemantauan dengan Amazon ElastiCache for Redis menggunakan Amazon CloudWatch

7. Implementasikan replikasi data untuk melimpahkan baca ke beberapa instans dan meningkatkan performa dan ketersediaan pembacaan data.

Sumber daya

Dokumen terkait:

- Menggunakan Amazon ElastiCache Well-Architected Lens
- Praktik terbaik pemantauan dengan Amazon ElastiCache for Redis menggunakan Amazon CloudWatch
- Metrik Mana yang Harus Saya Pantau?
- Laporan resmi Performa pada Skala Besar dengan Amazon ElastiCache
- Tantangan dan strategi caching

Video terkait:

- Jalur Pembelajaran Amazon ElastiCache
- Desain untuk keberhasilan dengan praktik terbaik Amazon ElastiCache
- AWS re:Invent 2020 - Desain untuk keberhasilan dengan praktik terbaik Amazon ElastiCache
- AWS re:Invent 2023 - [PELUNCURAN] Memperkenalkan Amazon ElastiCache Nirserver
- AWS re:Invent 2022 - 5 cara hebat untuk mengonsep ulang lapisan data Anda dengan Redis
- AWS re:Invent 2021 - Memahami Amazon ElastiCache for Redis

Contoh terkait:

- Meningkatkan performa basis data MySQL dengan strategi pemberian tag Amazon ElastiCache for Redis

Jaringan dan pengiriman konten

Solusi jaringan optimal untuk beban kerja bervariasi berdasarkan latensi, persyaratan throughput, jitter, dan bandwith. Batas fisik, seperti sumber daya on-premise atau pengguna, menentukan opsi lokasi. Batas-batas ini dapat diimbangi dengan penempatan sumber daya atau lokasi edge.

Di AWS, jaringan dibuat menjadi virtual dan tersedia dalam berbagai jenis dan konfigurasi yang berbeda-beda. Hal ini membuatnya lebih mudah untuk disesuaikan dengan kebutuhan jaringan Anda. AWS menawarkan fitur produk (misalnya, Enhanced Networking, instans yang dioptimalkan Amazon EC2, akselerasi transfer Amazon S3, dan Amazon CloudFront yang dinamis) untuk mengoptimalkan lalu lintas jaringan. AWS juga menawarkan fitur jaringan (misalnya perutean latensi Amazon Route 53, titik akhir Amazon VPC, AWS Direct Connect, dan AWS Global Accelerator) untuk mengurangi jarak jaringan atau jitter.

Area fokus ini berbagi panduan dan praktik terbaik untuk mendesain, mengonfigurasi, dan mengoperasikan solusi jaringan dan pengiriman konten yang efisien di cloud.

Praktik terbaik

- PERF04-BP01 Memahami bagaimana jaringan memengaruhi performa
- PERF04-BP02 Mengevaluasi fitur jaringan yang tersedia
- PERF04-BP03 Memilih konektivitas khusus atau VPN yang tepat untuk beban kerja Anda
- PERF04-BP04 Menggunakan penyeimbangan beban untuk mendistribusikan lalu lintas di berbagai sumber daya
- PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa
- PERF04-BP06 Memilih lokasi beban kerja Anda berdasarkan kebutuhan jaringan
- PERF04-BP07 Mengoptimalkan konfigurasi jaringan berdasarkan metrik

PERF04-BP01 Memahami bagaimana jaringan memengaruhi performa

Analisis dan pahami bagaimana keputusan terkait jaringan memengaruhi beban kerja Anda untuk memberikan performa yang efisien dan pengalaman pengguna yang lebih baik.

Antipola umum:

- Semua lalu lintas mengalir melalui pusat data Anda.
- Anda merutekan semua lalu lintas melalui firewall pusat, bukan menggunakan alat keamanan jaringan cloud-native.
- Anda menyediakan koneksi AWS Direct Connect tanpa memahami persyaratan penggunaan aktual.
- Anda tidak mempertimbangkan karakteristik beban kerja dan biaya overhead enkripsi ketika menentukan solusi jaringan Anda.
- Anda menggunakan konsep dan strategi on-premise untuk solusi jaringan di cloud.

Manfaat menjalankan praktik terbaik ini: Memahami bagaimana jaringan memengaruhi kinerja beban kerja membantu Anda mengidentifikasi potensi hambatan, meningkatkan pengalaman pengguna, meningkatkan keandalan, dan menurunkan pemeliharaan operasional saat beban kerja berubah.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Jaringan bertanggung jawab atas konektivitas antara komponen aplikasi, layanan cloud, jaringan edge, dan data on-premise, oleh karena itu, jaringan dapat sangat memengaruhi performa beban kerja. Selain performa beban kerja, pengalaman pengguna juga dapat terpengaruh oleh latensi jaringan, bandwidth, protokol, lokasi, kemacetan jaringan, jitter, throughput, dan aturan perutean.

Miliki daftar terdokumentasi kebutuhan jaringan dari beban kerja termasuk latensi, ukuran paket, aturan perutean, protokol, dan pola lalu lintas pendukung. Tinjau solusi jaringan yang tersedia dan identifikasi layanan mana yang memenuhi karakteristik jaringan beban kerja Anda. Jaringan berbasis cloud dapat dengan cepat dibangun kembali, sehingga diperlukan peningkatan arsitektur jaringan Anda seiring berjalannya waktu untuk meningkatkan efisiensi kinerja.

Langkah Implementasi:

1. Tentukan dan dokumentasikan persyaratan performa jaringan, termasuk metrik seperti latensi jaringan, bandwidth, protokol, lokasi, pola lalu lintas (lonjakan dan frekuensi), throughput, enkripsi, inspeksi, dan aturan perutean.
2. Pelajari tentang layanan jaringan utama AWS seperti VPC, AWS Direct Connect, Elastic Load Balancing (ELB), dan Amazon Route 53.
3. Rekam karakteristik jaringan utama berikut:

Karakteristik	Alat dan metrik
Karakteristik jaringan dasar	• VPC Flow Logs • AWS Transit Gateway Flow Logs • Metrik AWS Transit Gateway • Metrik AWS PrivateLink
Karakteristik jaringan aplikasi	• Elastic Fabric Adapter • Metrik AWS App Mesh • Metrik Amazon API Gateway
Karakteristik jaringan edge	• Metrik Amazon CloudFront • Metrik Amazon Route 53 • Metrik AWS Global Accelerator
Karakteristik jaringan hybrid	• Metrik AWS Direct Connect • Metrik AWS Site-to-Site VPN • Metrik AWS Client VPN • Metrik AWS Cloud WAN
Karakteristik jaringan keamanan	• Metrik AWS Shield, AWS WAF, dan AWS Network Firewall
Karakteristik penelusuran	• AWS X-Ray • Reachability Analyzer VPC • Network Access Analyzer • Amazon Inspector • Amazon CloudWatch RUM

4. Buat tolok ukur dan uji kinerja jaringan:
   a. Buat tolok ukur throughput jaringan, karena beberapa faktor yang dapat memengaruhi kinerja jaringan Amazon EC2 saat instans berada di VPC yang sama. Ukur bandwidth jaringan antar instans Linux Amazon EC2 di VPC yang sama.
   b. Jalankan pengujian beban untuk bereksperimen dengan solusi dan opsi jaringan

Sumber daya

Dokumen terkait:

- Application Load Balancer
- Peningkatan Jaringan EC2 di Linux
- Jaringan yang Ditingkatkan EC2 di Windows
- Grup Penempatan EC2
- Memungkinkan Jaringan yang Ditingkatkan dengan Elastic Network Adapter (ENA) di Instans Linux
- Network Load Balancer
- Produk Jaringan dengan AWS
- Transit Gateway
- Beralih ke perutean berbasis latensi di Amazon Route 53
- Titik akhir VPC

Video terkait:

- AWS re:Invent 2023 - Fondasi jaringan AWS
- AWS re:Invent 2023 - Apa manfaat jaringan untuk aplikasi Anda?
- AWS re:Invent 2023 - Desain VPC tingkat lanjut dan kemampuan baru
- AWS re:Invent 2023 - Panduan jaringan cloud bagi pengembang
- AWS re:Invent 2019 - Konektivitas ke AWS dan arsitektur jaringan AWS hibrid
- AWS re:Invent 2019 - Mengoptimalkan Kinerja Jaringan untuk Instans Amazon EC2
- AWS Summit Online - Meningkatkan Kinerja Jaringan Global untuk Aplikasi
- AWS re:Invent 2020 - Praktik terbaik dan tips jaringan dengan Kerangka Kerja Well-Architected
- AWS re:Invent 2020 - Praktik terbaik jaringan AWS dalam migrasi skala besar

Contoh terkait:

- AWS Transit Gateway dan Solusi Keamanan yang Dapat Diskalakan
- Lokakarya Jaringan AWS
- Lokakarya Firewall Jaringan Langsung
- Mengamati dan Mendiagnosis Jaringan Anda di AWS

- Menemukan dan mengatasi Kesalahan Konfigurasi Jaringan pada AWS

PERF04-BP02 Mengevaluasi fitur jaringan yang tersedia

Evaluasi fitur jaringan di cloud yang dapat meningkatkan kinerja. Ukur dampak fitur-fitur ini melalui pengujian, metrik, dan analisis. Misalnya, manfaatkan fitur tingkat jaringan yang tersedia untuk mengurangi latensi, jarak jaringan, atau masalah kecepatan (jitter).

Antipola umum:

- Anda hanya menggunakan satu Wilayah karena di sanalah lokasi fisik kantor pusat Anda.
- Anda menggunakan firewall, bukan grup keamanan, untuk memfilter lalu lintas.
- Anda lebih memilih melanggar TLS untuk pemeriksaan lalu lintas daripada mengandalkan grup keamanan, kebijakan titik akhir, dan fungsionalitas cloud-native lainnya.
- Anda hanya menggunakan segmentasi berbasis subnet, bukan grup keamanan.

Manfaat menjalankan praktik terbaik ini: Mengevaluasi semua fitur dan opsi layanan dapat meningkatkan performa beban kerja Anda, menurunkan biaya infrastruktur, mengurangi upaya yang diperlukan untuk memelihara beban kerja Anda, dan meningkatkan postur keamanan Anda secara keseluruhan. Anda dapat menggunakan backbone AWS global memberikan pengalaman jaringan yang optimal bagi pelanggan Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

AWS menawarkan layanan seperti AWS Global Accelerator dan Amazon CloudFront yang dapat membantu meningkatkan performa jaringan, sementara sebagian besar layanan AWS memiliki fitur produk (seperti fitur Amazon S3 Transfer Acceleration) untuk mengoptimalkan lalu lintas jaringan.

Tinjau opsi konfigurasi terkait jaringan mana yang tersedia untuk Anda serta bagaimana dampaknya terhadap beban kerja Anda. Optimalisasi performa bergantung pada pemahaman tentang bagaimana opsi-opsi ini berinteraksi dengan arsitektur Anda serta dampaknya terhadap performa terukur dan pengalaman pengguna.

Langkah implementasi

- Buat daftar komponen beban kerja.

  - Pertimbangkan untuk menggunakan AWS Cloud WAN untuk membangun, mengelola, dan memantau jaringan organisasi Anda saat membangun jaringan global terpadu.
  - Pantau jaringan global dan inti Anda dengan metrik Amazon CloudWatch Logs. Manfaatkan Amazon CloudWatch RUM, yang memberikan wawasan untuk membantu mengidentifikasi, memahami, dan menyempurnakan pengalaman digital pengguna.
  - Lihat latensi jaringan agregat antara Wilayah AWS dan Zona Ketersediaan, serta di dalam setiap Zona Ketersediaan, menggunakan AWS Network Manager untuk mendapatkan wawasan tentang bagaimana performa aplikasi Anda berkaitan dengan performa jaringan AWS yang mendasarinya.
  - Gunakan alat basis data manajemen konfigurasi (CMDB) yang ada atau layanan seperti AWS Config untuk membuat inventaris beban kerja Anda dan cara mengonfigurasinya.
- Jika ini adalah beban kerja yang ada, identifikasi dan dokumentasikan tolok ukur untuk metrik performa Anda, yang fokus pada hambatan dan area yang perlu ditingkatkan. Metrik jaringan terkait performa akan berbeda per beban kerja berdasarkan persyaratan bisnis dan karakteristik beban kerja. Sebagai permulaan, metrik ini mungkin penting untuk ditinjau untuk beban kerja Anda: bandwidth, latensi, kehilangan paket, jitter, dan transmisi ulang.
- Jika ini adalah beban kerja baru, lakukan pengujian beban untuk mengidentifikasi hambatan performa.
- Untuk hambatan performa yang Anda identifikasi, tinjau opsi konfigurasi untuk solusi Anda guna mengidentifikasi peluang peningkatan performa. Lihat opsi dan fitur jaringan utama berikut:

Peluang peningkatan	Solusi
Jalur atau rute jaringan	Gunakan Network Access Analyzer untuk mengidentifikasi jalur atau rute.
Protokol jaringan	Lihat PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa
Topologi jaringan	Evaluasi tarik ulur operasional dan performa Anda antara Peering VPC dan AWS Transit Gateway saat menghubungkan beberapa akun. AWS Transit Gateway menyederh anakan cara Anda menyambungkan silang semua VPC Anda, yang bisa saja berada di ribuan Akun AWS dan ke dalam jaringan on-

Peluang peningkatan	Solusi
	premise. Bagikan AWS Transit Gateway Anda di antara beberapa akun menggunakan AWS Resource Access Manager. Lihat PERF04-BP03 Memilih konektivitas khusus atau VPN yang tepat untuk beban kerja Anda
Layanan jaringan	AWS Global Accelerator adalah layanan jaringan yang meningkatkan performa lalu lintas pengguna Anda hingga 60% menggunakan infrastruktur jaringan global AWS. Amazon CloudFront dapat meningkatkan performa pengiriman konten beban kerja dan memperbaiki latensi Anda secara global. Gunakan Lambda@Edge untuk menjalank an fungsi yang menyesuaikan konten yang dikirimkan CloudFront lebih dekat ke pengguna, mengurangi latensi, dan meningkat kan performa. Amazon Route 53 menawarkan perutean berbasis latensi, perutean geolokasi, perutean geoproksimitas, dan perutean berbasis IP opsi untuk membantu Anda meningkatkan performa beban kerja Anda bagi audiens global. Identifikasikan opsi perutean mana yang akan mengoptimalkan performa beban kerja Anda dengan meninjau lalu lintas beban kerja dan lokasi pengguna Anda saat beban kerja Anda terdistribusi secara global.

Peluang peningkatan	Solusi
Fitur sumber daya penyimpanan	Amazon S3 Transfer Acceleration adalah fitur yang memungkinkan pengguna eksternal mendapatkan manfaat pengoptimalan jaringan dari CloudFront untuk mengungga h data ke Amazon S3. Hal ini meningkatkan kemampuan transfer data dalam jumlah besar dari lokasi jarak jauh yang tidak memiliki koneksi khusus ke AWS Cloud. Titik Akses Multi-Wilayah Amazon S3 mereplikasi konten ke beberapa Wilayah dan menyederhanakan beban kerja dengan menyediakan satu titik akses. Saat Titik Akses Multi-Wilayah digunakan, Anda dapat meminta atau menulis data ke Amazon S3 dengan layanan yang mengidentifikasi bucket latensi terendah.

Peluang peningkatan	Solusi
Fitur sumber daya komputasi	Antarmuka Jaringan Elastis (ENA) yang digunakan oleh instans Amazon EC2, kontainer, dan fungsi Lambda dibatasi berdasarkan per alur. Tinjau grup penempatan Anda untuk mengoptimalkan throughput jaringan EC2. Untuk menghindari kemacetan pada basis per alur, rancang aplikasi Anda sedemikian rupa hingga menggunakan beberapa alur. Untuk memantau dan mendapatkan visibilitas tentang metrik jaringan terkait komputasi Anda, gunakan CloudWatch Metrics dan ethtool. Perintah `ethtool` disertakan dalam driver ENA dan mengekspos metrik terkait jaringan tambahan yang dapat dipublikasikan sebagai metrik kustom ke CloudWatch. Amazon Elastic Network Adapters (ENA) memberikan pengoptimalan lebih lanjut dengan memberikan throughput yang lebih baik untuk instans Anda dalam grup penempatan klaster. Elastic Fabric Adapter (EFA) adalah antarmuka jaringan untuk instans Amazon EC2 yang memungkinkan Anda menjalankan beban kerja yang memerlukan komunikasi antarsimpul tingkat tinggi dalam skala besar di AWS. Instans yang dioptimalkan Amazon EBS menggunakan tumpukan konfigurasi yang dioptimalkan dan menyediakan kapasitas khusus tambahan untuk meningkatkan I/O Amazon EBS.

Sumber daya

Dokumen terkait:

- Application Load Balancer
- Peningkatan Jaringan EC2 di Linux
- Jaringan yang Ditingkatkan EC2 di Windows
- Grup Penempatan EC2
- Memungkinkan Jaringan yang Ditingkatkan dengan Elastic Network Adapter (ENA) di Instans Linux
- Network Load Balancer
- Produk Jaringan dengan AWS
- Beralih ke Perutean Berbasis Latensi di Amazon Route 53
- Titik Akhir VPC
- VPC Flow Logs

Video terkait:

- AWS re:Invent 2023 – Siap dengan yang berikutnya? Merancang jaringan untuk pertumbuhan dan fleksibilitas
- AWS re:Invent 2023 – Desain VPC tingkat lanjut dan kemampuan baru
- AWS re:Invent 2023 – Panduan jaringan cloud bagi pengembang
- AWS re:Invent 2022 – Memahami infrastruktur jaringan AWS
- AWS re:Invent 2019 – Konektivitas ke AWS dan arsitektur jaringan AWS hibrid
- AWS re:Invent 2018 – Mengoptimalkan Kinerja Jaringan untuk Instans Amazon EC2
- AWS Global Accelerator

Contoh terkait:

- AWS Transit Gateway dan Solusi Keamanan yang Dapat Diskalakan
- Lokakarya Jaringan AWS
- Mengamati dan mendiagnosis jaringan Anda
- Menemukan dan mengatasi kesalahan konfigurasi jaringan pada AWS

PERF04-BP03 Memilih konektivitas khusus atau VPN yang tepat untuk beban kerja Anda

Ketika diperlukan konektivitas hybrid untuk menghubungkan sumber daya on-premise dan cloud, sediakan bandwidth yang memadai untuk memenuhi persyaratan performa Anda. Perkirakan persyaratan bandwidth dan latensi untuk beban kerja hybrid Anda. Angka-angka ini akan mendorong persyaratan penyesuaian ukuran Anda.

Antipola umum:

- Anda hanya mengevaluasi solusi VPN untuk persyaratan enkripsi jaringan Anda.
- Anda tidak mengevaluasi opsi cadangan atau konektivitas redundan.
- Anda tidak mengidentifikasi semua persyaratan beban kerja (kebutuhan enkripsi, protokol, bandwidth, dan lalu lintas).

Manfaat menjalankan praktik terbaik ini: Memilih dan mengonfigurasi solusi konektivitas yang tepat akan meningkatkan keandalan beban kerja dan memaksimalkan performa. Dengan mengidentifikasi persyaratan beban kerja, membuat perencanaan ke depan, dan mengevaluasi solusi hybrid, Anda dapat meminimalkan perubahan jaringan fisik yang mahal dan biaya operasional sekaligus meningkatkan kecepatan perolehan nilai.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Kembangkan arsitektur jaringan hybrid berdasarkan persyaratan bandwidth Anda. AWS Direct Connect memungkinkan Anda untuk menghubungkan jaringan on-premise Anda secara privat dengan AWS. Layanan ini ideal ketika Anda memerlukan bandwidth tinggi dan latensi rendah sambil mencapai performa yang konsisten. Sambungan VPN membangun sambungan yang aman lewat internet. VPN digunakan ketika yang diperlukan hanyalah sambungan sementara, ketika biaya menjadi pertimbangan, atau sebagai kontingensi sambil menunggu terbentuknya konektivitas jaringan fisik yang kuat saat menggunakan AWS Direct Connect.

Jika persyaratan bandwidth Anda tinggi, Anda dapat mempertimbangkan beberapa layanan AWS Direct Connect atau VPN. Lalu lintas dapat diberikan penyeimbangan beban di seluruh layanan, meskipun kami tidak merekomendasikan penyeimbangan beban antara AWS Direct Connect dan VPN dikarenakan perbedaan latensi dan bandwidth.

Langkah implementasi

1. Perkirakan persyaratan bandwidth dan latensi aplikasi yang sudah Anda miliki.
   a. Untuk beban kerja lama yang akan beralih ke AWS, manfaatkan data dari sistem pemantauan jaringan internal Anda.
   b. Untuk beban kerja baru atau lama yang data pemantauannya tidak Anda miliki, hubungi pemilik produk untuk menentukan metrik performa yang memadai dan memberikan pengalaman pengguna yang baik.
2. Pilih sambungan khusus atau VPN sebagai opsi konektivitas Anda. Berdasarkan semua persyaratan beban kerja Anda (kebutuhan enkripsi, bandwidth, dan lalu lintas), Anda dapat memilih AWS Direct Connect atau AWS VPN (atau keduanya). Diagram berikut dapat membantu Anda memilih jenis sambungan yang tepat.
   a. AWS Direct Connect menyediakan konektivitas khusus ke lingkungan AWS, mulai dari 50 Mbps hingga 100 Gbps, menggunakan sambungan khusus atau sambungan yang di-host. Layanan ini memberi Anda latensi yang terkelola dan terkontrol serta bandwidth yang tersedia agar beban kerja Anda dapat terhubung ke lingkungan lain secara efisien. Dengan menggunakan partner AWS Direct Connect, Anda dapat memiliki konektivitas menyeluruh dari beberapa lingkungan, yang memberikan jaringan lebih luas dengan performa konsisten. AWS menawarkan bandwidth sambungan langsung dengan penskalaan menggunakan 100 Gbps native, link aggregation group (LAG), atau BGP equal-cost multipath (ECMP).
   b. AWS Site-to-Site VPN memberikan layanan VPN terkelola yang mendukung keamanan protokol internet (IPsec). Ketika sambungan VPN dibuat, setiap sambungan VPN mencakup dua terowongan untuk ketersediaan tinggi.
3. Ikuti dokumentasi AWS untuk memilih opsi konektivitas yang tepat:
   a. Jika Anda memutuskan untuk menggunakanAWS Direct Connect, pilih bandwidth yang sesuai untuk konektivitas Anda.
   b. Jika Anda menggunakan AWS Site-to-Site VPN di beberapa lokasi untuk terhubung ke Wilayah AWS, gunakan sambungan Site-to-Site VPN yang dipercepat untuk mendapatkan peluang peningkatan performa jaringan.
   c. Jika desain jaringan Anda terdiri dari koneksi VPN IPsec lewat AWS Direct Connect, pertimbangkan menggunakan VPN IP Privat untuk meningkatkan keamanan dan mencapai segmentasi. VPN IP Privat Site-to-Site AWS di-deploy di atas antarmuka virtual transit (VIF).
   d. AWS Direct Connect SiteLink memungkinkan pembuatan koneksi latensi rendah dan redundan antara semua pusat data Anda di seluruh dunia dengan mengirimkan data melalui jalur tercepat antara lokasi-lokasi AWS Direct Connect, dengan melewati Wilayah AWS.

4. Lakukan validasi penyiapan konektivitas Anda sebelum deployment ke produksi. Lakukan pengujian keamanan dan performa untuk memastikan persyaratan bandwidth, keandalan, latensi, dan kepatuhan Anda terpenuhi.
5. Pantau performa dan penggunaan konektivitas Anda secara teratur dan optimalkan jika diperlukan.

Bagan alur performa penentu.

Sumber daya

Dokumen terkait:

- Produk Jaringan dengan AWS
- AWS Transit Gateway
- Titik akhir VPC
- Membangun Infrastruktur Jaringan AWS Multi-VPC yang Aman dan Dapat Diskalakan
- VPN Klien

Video terkait:

- AWS re:Invent 2023 – Membangun konektivitas jaringan hibrida dengan AWS
- AWS re: Invent 2023 – Mengamankan konektivitas jarak jauh ke AWS
- AWS re: Invent 2022 – Mengoptimalkan kinerja dengan Amazon CloudFront
- AWS re:Invent 2019 – Konektivitas ke AWS dan arsitektur jaringan AWS hibrid
- AWS re:Invent 2020 – AWS Transit Gateway Connect

Contoh terkait:

- AWS Transit Gateway dan Solusi Keamanan yang Dapat Diskalakan
- Lokakarya Jaringan AWS

PERF04-BP04 Menggunakan penyeimbangan beban untuk mendistribusikan lalu lintas di berbagai sumber daya

Distribusikan lalu lintas di berbagai sumber daya atau layanan untuk memanfaatkan elastisitas yang ada di cloud untuk beban kerja Anda. Anda juga dapat menggunakan penyeimbang beban untuk memindahkan beban penghentian enkripsi guna meningkatkan performa dan keandalan serta untuk mengelola dan merutekan lalu lintas secara efektif.

Antipola umum:

- Anda tidak mempertimbangkan persyaratan beban kerja Anda ketika memilih jenis penyeimbang beban.
- Anda tidak memanfaatkan fitur penyeimbang beban untuk mengoptimalkan performa.
- Beban kerja terpapar langsung ke internet tanpa penyeimbang beban.
- Anda merutekan semua lalu lintas internet melalui penyeimbang beban yang ada.
- Anda menggunakan penyeimbangan beban TCP umum dan membuat setiap simpul komputasi menangani enkripsi SSL.

Manfaat menjalankan praktik terbaik ini: Penyeimbang beban menangani berbagai beban lalu lintas aplikasi Anda dalam satu atau beberapa Zona Ketersediaan dan menghadirkan ketersediaan yang tinggi, penskalaan otomatis, dan pemanfaatan yang lebih baik untuk beban kerja Anda.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Penyeimbang beban berfungsi sebagai titik masuk untuk beban kerja Anda, yakni titik asal penyeimbang beban mendistribusikan lalu lintas ke target backend Anda, seperti kontainer atau instans komputasi, untuk meningkatkan pemanfaatan.

Memilih jenis penyeimbang beban yang tepat adalah langkah pertama untuk mengoptimalkan arsitektur Anda. Mulai dengan mencantumkan karakteristik beban kerja Anda, seperti protokol (misalnya, TCP, HTTP, TLS, atau WebSockets), jenis target (seperti instans, kontainer, atau nirserver), persyaratan aplikasi (seperti sambungan berdurasi lama, autentikasi pengguna, atau keeratan), dan penempatan (seperti Wilayah, Zona Lokal, Outpost, atau isolasi zona).

AWS menyediakan beberapa model untuk aplikasi Anda untuk menggunakan penyeimbangan beban. Application Load Balancer sangat ideal untuk menyeimbangkan beban lalu lintas HTTP dan

HTTPS dan menyediakan perutean permintaan lanjutan yang ditargetkan pada pengiriman arsitektur aplikasi modern, termasuk layanan mikro dan kontainer.

Network Load Balancer sangat ideal untuk menyeimbangkan beban lalu lintas TCP yang memerlukan kinerja ekstrem. Penyeimbang beban ini mampu menangani jutaan permintaan per detik sekaligus membuat latensi tetap rendah, serta dioptimalkan untuk menangani pola lalu lintas yang tidak stabil dan mendadak.

Elastic Load Balancing menyediakan manajemen sertifikat terintegrasi dan dekripsi SSL/TLS, memberikan fleksibilitas kepada Anda untuk mengelola pengaturan SSL penyeimbang beban secara terpusat serta memindahkan beban yang banyak menggunakan CPU dari beban kerja Anda.

Setelah memilih penyeimbang beban yang tepat, Anda dapat mulai memanfaatkan fitur-fiturnya untuk mengurangi jumlah upaya yang harus dilakukan backend guna melayani lalu lintas.

Contohnya, dengan menggunakan Application Load Balancer (ALB) dan Network Load Balancer (NLB), Anda dapat melakukan pemindahan beban enkripsi SSL/TLS, yang merupakan peluang untuk menghindari handshake TLS yang sarat CPU diselesaikan oleh target Anda dan juga untuk meningkatkan manajemen sertifikat.

Ketika Anda mengonfigurasi pemindahan beban SSL/TLS di penyeimbang beban, penyeimbang beban menjadi bertanggung jawab atas enkripsi lalu lintas dari dan ke klien sekaligus memberikan lalu lintas tidak terenkripsi ke backend Anda, sehingga membebaskan sumber daya backend Anda dan meningkatkan waktu respons untuk klien.

Application Load Balancer juga dapat melayani lalu lintas HTTP/2 tanpa harus mendukungnya di target Anda. Keputusan sederhana ini dapat meningkatkan waktu respons aplikasi Anda, karena HTTP/2 menggunakan sambungan TCP dengan lebih efisien.

Persyaratan latensi beban kerja Anda harus dipertimbangkan ketika menentukan arsitektur. Sebagai contoh, jika Anda memiliki aplikasi yang sensitif latensi, Anda dapat memutuskan untuk menggunakan Network Load Balancer, yang menawarkan latensi yang sangat rendah. Alternatifnya, Anda dapat memutuskan untuk membawa beban kerja lebih dekat ke pelanggan dengan memanfaatkan Application Load Balancer di Zona Lokal AWS atau bahkan AWS Outposts.

Pertimbangan lain untuk beban kerja yang sensitif latensi adalah penyeimbangan beban lintas zona. Dengan penyeimbangan beban lintas zona, setiap simpul penyeimbang beban mendistribusikan lalu lintas ke target terdaftar di semua Zona Ketersediaan yang diaktifkan.

Gunakan Auto Scaling yang terintegrasi dengan penyeimbang beban Anda. Salah satu aspek penting dari sistem dengan performa yang efisien berkaitan dengan penyesuaian ukuran sumber daya

backend Anda. Untuk melakukannya, Anda dapat memanfaatkan integrasi penyeimbang beban untuk sumber daya target backend. Dengan menggunakan integrasi penyeimbang beban dengan grup Auto Scaling, target akan ditambahkan atau disingkirkan dari penyeimbang beban sebagaimana diperlukan untuk merespons lalu lintas masuk. Penyeimbang beban juga dapat diintegrasikan dengan Amazon ECS dan Amazon EKS untuk beban kerja dalam kontainer.

- Amazon ECS - Penyeimbangan beban layanan
- Penyeimbangan beban aplikasi di Amazon EKS
- Penyeimbangan beban jaringan di Amazon EKS

Langkah implementasi

- Tentukan persyaratan penyeimbangan beban Anda termasuk volume lalu lintas, ketersediaan, dan skalabilitas aplikasi.
- Pilih jenis penyeimbang beban yang tepat untuk aplikasi Anda.
  - Gunakan Application Load Balancer untuk beban kerja HTTP/HTTPS.
  - Gunakan Network Load Balancer untuk beban kerja non-HTTP yang dijalankan di TCP atau UDP.
  - Gunakan kombinasi keduanya (ALB sebagai target NLB) jika Anda ingin memanfaatkan fitur kedua produk. Contohnya, Anda dapat melakukan hal ini jika Anda ingin menggunakan IP statis NLB bersama dengan perutean berbasis header HTTP dari ALB, atau jika Anda ingin memaparkan beban kerja HTTP Anda ke AWS PrivateLink
  - Untuk perbandingan lengkap penyeimbang beban, lihat Perbandingan produk ELB.
- Gunakan pemindahan beban SSL/TLS jika memungkinkan.
  - Konfigurasikan pendengar HTTPS/TLS dengan Application Load Balancer dan Network Load Balancer yang terintegrasi dengan AWS Certificate Manager.
  - Perhatikan, beberapa beban kerja mungkin memerlukan enkripsi menyeluruh karena alasan kepatuhan. Jika demikian, enkripsi wajib diaktifkan di target.
  - Untuk praktik terbaik keamanan, lihat SEC09-BP02 Menerapkan enkripsi data bergerak.
- Pilih algoritma perutean yang tepat (khusus ALB).
  - Algoritma perutean dapat membuat perbedaan tentang seberapa baik target backend Anda digunakan, oleh karena itu juga membuat perbedaan dalam dampaknya pada performa. Contohnya, ALB memberikan dua opsi untuk algoritma perutean:

  - Permintaan paling sedikit belum selesai: Gunakan untuk mendapatkan distribusi beban yang lebih baik ke target backend Anda untuk kasus ketika permintaan untuk aplikasi Anda bervariasi dalam tingkat kompleksitas atau target Anda bervariasi dalam kemampuan pemrosesannya.
  - Round robin: Gunakan ketika permintaan dan target serupa, atau jika Anda harus mendistribusikan permintaan sama rata di antara target.
- Pertimbangkan isolasi zona atau lintas zona.
  - Gunakan penonaktifan lintas zona (isolasi zona) untuk peningkatan latensi dan domain kegagalan zona. Ini dinonaktifkan menurut default di NLB dan di ALB Anda dapat menonaktifkannya per grup target.
  - Gunakan pengaktifan lintas zona untuk peningkatan ketersediaan dan fleksibilitas. Menurut default, lintas zona diaktifkan untuk ALB dan di NLB Anda dapat mengaktifkannya per grup target.
- Aktifkan keep-alive HTTP untuk beban kerja HTTP Anda (khusus ALB). Dengan fitur ini, penyeimbang beban dapat menggunakan ulang sambungan backend sampai waktu tetap aktif habis, sehingga meningkatkan waktu respons dan permintaan HTTP Anda serta mengurangi pemanfaatan sumber daya di target backend Anda. Untuk informasi mendetail tentang cara melakukan ini untuk Apache dan Nginx, lihat Apa saja pengaturan yang optimal untuk menggunakan Apache atau NGINX sebagai server backend untuk ELB?
- Aktifkan pemantauan untuk penyeimbang beban Anda.
  - Aktifkan log akses untuk Application Load Balancer dan Network Load Balancer.
  - Bidang utama yang perlu dipertimbangkan untuk ALB adalah `request_processing_time`, `request_processing_time`, dan `response_processing_time`.
  - Bidang utama yang harus dipertimbangkan untuk NLB adalah `connection_time` dan `tls_handshake_time`.
  - Bersiaplah untuk melakukan kueri log ketika Anda memerlukannya. Anda dapat menggunakan Amazon Athena untuk melakukan kueri log ALB dan log NLB.
  - Buat alarm untuk metrik yang terkait dengan performa seperti `TargetResponseTime` untuk ALB.

Sumber daya

Dokumen terkait:

- Perbandingan produk ELB
- Infrastruktur Global AWS
- Meningkatkan Performa dan Mengurangi Biaya Menggunakan Afinitas Zona Ketersediaan
- Langkah demi langkah untuk Analisis Log dengan Amazon Athena
- Mengueri log Application Load Balancer
- Memantau Application Load Balancers Anda
- Memantau Network Load Balancer Anda
- Menggunakan Elastic Load Balancing untuk mendistribusikan lalu lintas ke seluruh instans di grup Auto Scaling Anda

Video terkait:

- AWS re:Invent 2023: Apa manfaat jaringan untuk aplikasi Anda?
- AWS re:Inforce 20: Cara menggunakan Elastic Load Balancing untuk meningkatkan postur keamanan Anda dalam skala besar
- AWS re:Invent 2018: Elastic Load Balancing: Pembahasan Mendalam dan Praktik Terbaik
- AWS re:Invent 2021 - Cara memilih penyeimbang beban yang tepat untuk beban kerja AWS Anda
- AWS re:Invent 2019: Mendapatkan hasil maksimal dari Elastic Load Balancing untuk berbagai beban kerja

Contoh terkait:

- Gateway Load Balancer
- Sampel CDK dan AWS CloudFormation untuk Analisis Log dengan Amazon Athena

PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa

Buat keputusan terkait protokol untuk komunikasi antara sistem dan jaringan berdasarkan dampaknya terhadap kinerja beban kerja.

Ada hubungan antara latensi dan bandwidth untuk mencapai throughput. Jika transfer file Anda menggunakan Transmission Control Protocol (TCP), latensi yang lebih tinggi kemungkinan besar

akan mengurangi throughput secara keseluruhan. Ada pendekatan untuk memperbaiki hal ini dengan penyesuaian TCP dan pengoptimalan protokol transfer, tetapi salah satu solusinya adalah menggunakan User Datagram Protocol (UDP)).

Antipola umum:

- Anda menggunakan TCP untuk semua beban kerja tanpa memperhatikan persyaratan kinerjanya.

Manfaat menjalankan praktik terbaik ini: Memverifikasi bahwa protokol yang tepat telah digunakan untuk komunikasi antara pengguna dan komponen beban kerja akan membantu meningkatkan pengalaman pengguna secara keseluruhan untuk aplikasi Anda. Misalnya, UDP tanpa koneksi memungkinkan kecepatan tinggi, tetapi tidak menawarkan transmisi ulang atau keandalan tinggi. TCP adalah protokol berfitur lengkap, tetapi memerlukan biaya tambahan yang lebih besar untuk memproses paket.

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

Panduan implementasi

Jika Anda memiliki kemampuan untuk memilih protokol yang berbeda-beda untuk aplikasi Anda dan Anda memiliki keahlian di bidang ini, optimalkan aplikasi dan pengalaman pengguna akhir Anda dengan menggunakan protokol yang berbeda. Perhatikan bahwa pendekatan ini memiliki tingkat kesulitan yang tinggi dan hanya boleh dicoba jika Anda telah mengoptimalkan aplikasi Anda dengan cara lain terlebih dahulu.

Pertimbangan utama dalam meningkatkan performa beban kerja Anda yakni pemahaman persyaratan latensi dan throughput, lalu pemilihan protokol jaringan yang mengoptimalkan performa.

Kapan penggunaan TCP harus dipertimbangkan

TCP memberikan pengiriman data yang andal, dan dapat digunakan untuk komunikasi antara komponen beban kerja di mana keandalan dan jaminan pengiriman data merupakan hal yang penting. Banyak aplikasi berbasis web mengandalkan protokol berbasis TCP, seperti HTTP dan HTTPS, untuk membuka soket TCP untuk komunikasi antara komponen-komponen aplikasi. Email dan transfer data file adalah penerapan umum yang juga menggunakan TCP, karena ini adalah mekanisme transfer yang sederhana dan andal antara komponen-komponen aplikasi. Menggunakan TLS dengan TCP dapat menambahkan beberapa overhead ke komunikasi, yang dapat mengakibatkan peningkatan latensi dan pengurangan throughput, tetapi memiliki keunggulan dari segi keamanan. Overhead ini terutama berasal dari penambahan overhead proses handshake,

yang dapat memerlukan beberapa perjalanan pulang pergi agar selesai. Setelah handshake selesai, overhead enkripsi dan dekripsi data relatif kecil.

Kapan penggunaan UDP harus dipertimbangkan

UDP adalah protokol dengan orientasi nirkoneksi, oleh karena itu, cocok untuk aplikasi yang membutuhkan transmisi cepat dan efisien, seperti data VoIP, pemantauan, dan log. Selain itu, pertimbangkan untuk menggunakan UDP jika Anda memiliki komponen beban kerja yang merespons kueri kecil dari banyak klien untuk memastikan performa beban kerja yang optimal. Keamanan Lapisan Pengangkutan Datagram (DTLS) merupakan ekuivalen UDP untuk Keamanan Lapisan Pengangkutan (TLS). Ketika menggunakan DTLS dengan UDP, overhead berasal dari enkripsi dan dekripsi data, karena proses handshake disederhanakan. DTLS juga menambahkan sejumlah kecil overhead ke paket UDP, karena mencakup bidang tambahan untuk menunjukkan parameter keamanan dan untuk mendeteksi gangguan.

Kapan penggunaan SRD harus dipertimbangkan

Scalable reliable datagram (SRD) adalah protokol transpor jaringan yang dioptimalkan untuk beban kerja throughput tinggi karena kemampuannya untuk menjalankan lalu lintas penyeimbang beban melintasi beberapa jalur dan pulih dengan cepat dari penurunan paket atau kegagalan tautan. Oleh karena itu, SRD paling sesuai digunakan untuk beban kerja komputasi performa tinggi (HPC) yang memerlukan komunikasi latensi rendah dan throughput tinggi antara simpul komputasi. Hal ini dapat mencakup tugas pemrosesan paralel seperti simulasi, pemodelan, dan analisis data yang melibatkan banyak transfer data antara simpul.

Langkah implementasi

1. Gunakan AWS Global Accelerator dan AWS Transfer Family untuk memperbaiki throughput aplikasi transfer file online Anda. Layanan AWS Global Accelerator membantu Anda mendapatkan latensi lebih rendah antara perangkat klien dan beban kerja Anda di AWS. Dengan AWS Transfer Family, Anda dapat menggunakan protokol berbasis TCP seperti Secure Shell File Transfer Protocol (SFTP) dan File Transfer Protocol over SSL (FTPS) untuk menskalakan dengan aman dan mengelola transfer file ke layanan penyimpanan AWS.
2. Gunakan latensi jaringan untuk menentukan apakah TCP sesuai untuk komunikasi antara komponen beban kerja. Jika latensi jaringan antara server dan aplikasi klien Anda tinggi, maka handshake tiga arah TCP dapat memerlukan beberapa waktu, sehingga memengaruhi responsivitas aplikasi Anda. Metrik seperti time to first byte (TTFB) dan round-trip time (RTT) dapat digunakan untuk mengukur latensi jaringan. Jika beban kerja Anda menyajikan konten

dinamis kepada pengguna, pertimbangkan untuk menggunakan Amazon CloudFrontyang membuat sambungan persisten ke masing-masing asal konten dinamis untuk menyingkirkan waktu penyiapan sambungan yang akan memperlambat setiap permintaan klien.

3. Menggunakan TLS dengan TCP atau UDP dapat mengakibatkan peningkatan latensi dan pengurangan throughput untuk beban kerja Anda karena dampak enkripsi dan dekripsi. Untuk beban kerja tersebut, pertimbangkan pemindahan beban SSL/TLS di Elastic Load Balancing untuk meningkatkan performa beban kerja dengan mengizinkan penyeimbang beban menangani proses enkripsi dan dekripsi SSL/TLS, bukan menggunakan instans backend. Hal ini dapat membantu mengurangi pemanfaatan CPU di instans backend, yang dapat meningkatkan performa dan kapasitas.
4. Gunakan Network Load Balancer (NLB) untuk melakukan deployment layanan yang mengandalkan protokol UDP, seperti autentikasi dan otorisasi, logging, DNS, IoT, dan media streaming, untuk meningkatkan performa dan keandalan beban kerja Anda. NLB mendistribusikan lalu lintas UDP masuk di beberapa target, sehingga Anda dapat menskalakan beban kerja secara horizontal, meningkatkan kapasitas, dan mengurangi overhead satu target.
5. Untuk beban kerja Komputasi Performa Tinggi (HPC) Anda, pertimbangkan untuk menggunakan fungsi Adaptor Jaringan Elastis (ENA) Ekspres yang menggunakan protokol SRD untuk meningkatkan performa jaringan dengan memberikan bandwidth satu aliran yang lebih tinggi (25 Gbps) dan latensi ekor lebih rendah (99,9 persentil) untuk lalu lintas jaringan antara instans EC2.
6. Gunakan Application Load Balancer (ALB) untuk mengarahkan dan menyeimbangkan beban lalu lintas gRPC (Remote Procedure Calls) Anda antara komponen beban kerja atau antara layanan dan klien gRPC. gRPC menggunakan protokol HTTP/2 berbasis TCP untuk transpor dan gRPC memberikan manfaat terkait performa, seperti jejak jaringan lebih ringan, kompresi, serialisasi biner yang efisien, dukungan untuk berbagai bahasa, dan streaming dua arah.

Sumber daya

Dokumen terkait:

- Cara merutekan lalu lintas UDP ke Kubernetes
- Application Load Balancer
- Peningkatan Jaringan EC2 di Linux
- Jaringan yang Ditingkatkan EC2 di Windows
- Grup Penempatan EC2
- Memungkinkan Jaringan yang Ditingkatkan dengan Elastic Network Adapter (ENA) di Instans Linux

- Network Load Balancer
- Produk Jaringan dengan AWS
- Beralih ke Perutean Berbasis Latensi di Amazon Route 53
- Titik akhir VPC

Video terkait:

- AWS re:Invent 2022 – Menskalakan kinerja jaringan pada instans Amazon Elastic Compute Cloud generasi berikutnya
- AWS re:Invent 2022 – Fondasi jaringan aplikasi

Contoh terkait:

- AWS Transit Gateway dan Solusi Keamanan yang Dapat Diskalakan
- Lokakarya Jaringan AWS

PERF04-BP06 Memilih lokasi beban kerja Anda berdasarkan kebutuhan jaringan

Evaluasi opsi untuk penempatan sumber daya guna mengurangi latensi jaringan dan meningkatkan throughput, yang memberikan pengalaman pengguna optimal dengan mengurangi beban halaman dan waktu transfer data.

Antipola umum:

- Anda menggabungkan semua sumber daya beban kerja ke dalam satu lokasi geografis.
- Anda memilih Wilayah terdekat dengan lokasi Anda tetapi tidak dekat dengan pengguna akhir beban kerja.

Manfaat menjalankan praktik terbaik ini: Pengalaman pengguna sangat dipengaruhi oleh latensi antara pengguna dan aplikasi Anda. Dengan menggunakan Wilayah AWS yang sesuai dan jaringan global privat AWS, Anda dapat mengurangi latensi dan memberikan pengalaman yang lebih baik kepada pengguna jarak jauh.

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

Panduan implementasi

Sumber daya, seperti instans Amazon EC2, ditempatkan di Zona Ketersediaan dalam zona Wilayah AWS, Zona Lokal AWS, AWS Outposts, atau AWS Wavelength. Pemilihan lokasi ini memengaruhi latensi jaringan dan throughput dari lokasi pengguna tertentu. Layanan edge seperti Amazon CloudFront dan AWS Global Accelerator juga dapat digunakan untuk meningkatkan performa jaringan dengan caching konten di lokasi edge atau memberikan kepada pengguna jalur optimal ke beban kerja melalui jaringan global AWS.

Amazon EC2 menyediakan grup penempatan untuk jaringan. Grup penempatan adalah pengelompokan logis instans untuk mengurangi latensi. Menggunakan grup penempatan dengan jenis instans yang didukung dan Elastic Network Adapter (ENA) memungkinkan beban kerja dapat berpartisipasi di jaringan 25 Gbps berlatensi rendah dan lebih sedikit jitter. Grup penempatan direkomendasikan untuk beban kerja yang memanfaatkan latensi jaringan yang rendah, throughput jaringan yang tinggi, atau keduanya.

Layanan yang sensitif terhadap latensi dikirimkan di lokasi edge menggunakan jaringan global AWS, seperti Amazon CloudFront. Lokasi edge ini biasanya menyediakan layanan seperti jaringan pengiriman konten (CDN) dan sistem nama domain (DNS). Dengan memiliki layanan ini di edge, beban kerja dapat merespons dengan latensi yang rendah untuk meminta konten atau resolusi DNS. Layanan-layanan ini juga menyediakan layanan geografis seperti penargetan geografis konten (menyediakan konten yang berbeda berdasarkan lokasi pengguna akhir) atau perutean berbasis latensi untuk mengarahkan pengguna akhir ke Wilayah terdekat (latensi minimum).

Gunakan layanan edge untuk mengurangi latensi dan memungkinkan caching konten. Konfigurasikan kontrol cache dengan benar untuk DNS dan HTTP/HTTPS untuk mendapat manfaat maksimal dari pendekatan ini.

Langkah implementasi

- Tangkap informasi tentang lalu lintas IP ke dan dari antarmuka jaringan.
  - Pencatatan log lalu lintas IP menggunakan VPC Flow Logs
  - Cara alamat IP klien dipertahankan di AWS Global Accelerator
- Analisis 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.

- Pilih Wilayah untuk deployment beban kerja Anda berdasarkan elemen utama berikut:
  - Lokasi data: Untuk aplikasi dengan banyak data (seperti big data dan machine learning), kode aplikasi harus dijalankan sedekat mungkin dengan data.
  - Lokasi pengguna: Untuk aplikasi yang berinteraksi dengan pengguna, pilih Wilayah (satu atau lebih) yang dekat dengan pengguna beban kerja Anda.
  - Penghalang lainnya: Pertimbangkan penghalang seperti biaya dan kepatuhan sebagaimana dijelaskan di Hal-Hal yang Perlu Dipertimbangkan Saat Memilih Wilayah untuk Beban Kerja.
- Gunakan Zona Lokal AWS untuk menjalankan beban kerja seperti rendering video. Zona Lokal memungkinkan Anda mendapatkan semua manfaat dari komputasi dan sumber daya penyimpanan yang lebih dekat dengan pengguna akhir.
- Gunakan AWS Outposts untuk beban kerja yang harus tetap berada on-premise dan di tempat Anda ingin beban kerja tersebut berfungsi dengan lancar bersama beban kerja Anda yang lain di AWS.
- Aplikasi seperti streaming video live dengan resolusi tinggi, audio dengan fidelity tinggi, dan realitas tertambah atau realitas virtual (AR/VR) memerlukan latensi yang sangat rendah untuk perangkat 5G. Untuk aplikasi tersebut, pertimbangkan AWS Wavelength. AWS Wavelength menyematkan layanan komputasi dan penyimpanan AWS dalam jaringan 5G, sehingga dapat menyediakan infrastruktur komputasi edge seluler untuk mengembangkan, melakukan deployment, dan menskalakan aplikasi berlatensi sangat rendah.
- Gunakan caching lokal atau Solusi Caching AWS untuk sumber daya yang sering digunakan guna meningkatkan performa, mengurangi pergerakan data, dan menurunkan dampak lingkungan.

Service	When to use
Amazon CloudFront	Gunakan untuk meng-cache konten statis seperti gambar, skrip, dan video, serta 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 yang dapat membantu Anda menjalankan kode lebih dekat dengan pengguna beban kerja Anda seperti berikut:

Service	When to use
Lambda@edge	Gunakan untuk operasi dengan banyak komputasi yang dimulai saat objek tidak ada dalam cache.
Fungsi Amazon CloudFront	Gunakan untuk kasus penggunaan sederhana seperti permintaan HTTP atau manipulasi respons yang dapat dimulai oleh fungsi dengan masa pakai singkat.
AWS IoT Greengrass	Gunakan untuk menjalankan komputasi lokal, olahpesan, dan caching data untuk perangkat yang terhubung.

- Beberapa aplikasi memerlukan titik masuk tetap atau performa lebih tinggi dengan mengurangi jitter dan latensi bita pertama, dan meningkatkan throughput. Aplikasi ini bisa mendapatkan manfaat dari layanan jaringan yang memberikan alamat IP anycast statis dan penghentian TCP di lokasi edge. AWS Global Accelerator dapat meningkatkan performa untuk aplikasi Anda hingga sebesar 60% dan memberikan failover cepat untuk arsitektur multi-wilayah. AWS Global Accelerator memberikan kepada Anda alamat IP anycast statis yang berfungsi sebagai titik masuk tetap untuk aplikasi Anda yang di-hosting di satu atau lebih Wilayah AWS. Alamat IP ini mengizinkan lalu lintas masuk ke jaringan global AWS sedekat mungkin ke pengguna Anda. AWS Global Accelerator mengurangi waktu penyiapan sambungan awal dengan membuat sambungan TCP antara klien dan lokasi edge AWS yang terdekat ke klien. Tinjau penggunaan AWS Global Accelerator untuk meningkatkan performa beban kerja TCP/UDP Anda dan memberikan failover cepat untuk arsitektur multi-Wilayah.

Sumber daya

Praktik Terbaik Terkait:

- COST07-BP02 Mengimplementasikan Wilayah berdasarkan biaya
- COST08-BP03 Mengimplementasikan layanan untuk mengurangi biaya transfer data
- REL10-BP01 Melakukan deployment beban kerja ke beberapa lokasi
- REL10-BP02 Memilih lokasi yang sesuai untuk deployment multilokasi

- SUS01-BP01 Memilih Wilayah berdasarkan persyaratan bisnis dan tujuan keberlanjutan
- SUS02-BP04 Mengoptimalkan penempatan geografis beban kerja berdasarkan persyaratan jaringannya
- SUS04-BP07 Meminimalkan perpindahan data di jaringan

Dokumen terkait:

- 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
- AWS Wavelength
- Amazon CloudFront
- AWS Global Accelerator
- AWS Direct Connect
- AWS Site-to-Site VPN
- Amazon Route 53

Video terkait:

- Video Penjelas Zona Lokal AWS
- AWS Outposts: Gambaran Umum 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 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 menggunakan AWS
- AWS re:Invent 2020: Manajemen lalu lintas global dengan Amazon Route 53

Contoh terkait:

- Lokakarya Perutean Kustom AWS Global Accelerator
- Menangani Penulisan Ulang dan Pengarahan Ulang menggunakan Fungsi Edge

PERF04-BP07 Mengoptimalkan konfigurasi jaringan berdasarkan metrik

Gunakan data yang telah terkumpul dan dianalisis untuk mengambil keputusan yang tepat terkait pengoptimalan konfigurasi jaringan Anda.

Antipola umum:

- Anda beranggapan bahwa semua masalah kinerja disebabkan oleh aplikasi.
- Anda hanya menguji performa jaringan dari lokasi yang dekat dari tempat deployment beban kerja.
- Anda menggunakan konfigurasi default untuk semua layanan jaringan.
- Anda menyediakan terlalu banyak sumber daya jaringan untuk memberikan kapasitas yang memadai.

Manfaat menjalankan praktik terbaik ini: Dengan mengumpulkan metrik jaringan AWS yang diperlukan dan mengimplementasikan alat pemantauan jaringan, Anda dapat memahami performa jaringan dan mengoptimalkan konfigurasi jaringan.

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

Panduan implementasi

Memantau lalu lintas ke dan dari VPC, subnet, atau antarmuka jaringan sangat penting untuk memahami cara memanfaatkan sumber daya jaringan AWS dan mengoptimalkan konfigurasi jaringan. Dengan menggunakan alat jaringan AWS berikut ini, Anda dapat lebih jauh memeriksa informasi tentang penggunaan lalu lintas, akses jaringan, dan log.

Langkah implementasi

- Identifikasikan metrik kinerja utama seperti latensi atau kehilangan paket untuk dikumpulkan. AWS menyediakan beberapa alat yang dapat membantu Anda mengumpulkan metrik-metrik ini. Dengan

menggunakan alat-alat berikut, Anda dapat lebih lanjut memeriksa informasi tentang penggunaan lalu lintas, akses jaringan, dan log:

Alat AWS	Di mana harus menggunakan
Amazon VPC IP Address Manager.	Gunakan IPAM untuk merencanakan, melacak, dan memantau alamat IP untuk AWS dan beban kerja on-premise Anda. Ini adalah praktik terbaik untuk mengoptimalkan alokasi dan penggunaan alamat IP.
VPC Flow Logs	Gunakan VPC Flow Log untuk menangkap informasi mendetail tentang lalu lintas ke dan dari antarmuka jaringan di VPC Anda. Dengan VPC Flow Log, Anda dapat mendiagnosis aturan grup keamanan yang terlalu ketat atau longgar dan menentukan arah lalu lintas ke dan dari antarmuka jaringan.
AWS Transit Gateway Flow Logs	Gunakan AWS Transit Gateway Flow Logs untuk menangkap informasi tentang lalu lintas IP yang masuk dan keluar gateway transit Anda.
Logging kueri DNS	Informasi log tentang kueri DNS publik atau privat yang diterima Route 53. Dengan log DNS, Anda dapat mengoptimalkan konfigura si DNS dengan memahami domain atau sub-domain yang diminta atau lokasi EDGE Route 53 yang merespons kueri DNS.

Alat AWS	Di mana harus menggunakan
Reachability Analyzer	Reachability Analyzer untuk menganalisis dan melakukan debug keterjangkauan jaringan. Reachability Analyzer adalah alat analisis konfigurasi yang memungkinkan Anda melakukan pengujian konektivitas antara sumber daya sumber dan sumber daya destinasi di VPC Anda. Alat ini membantu Anda memverifikasi bahwa konfigurasi jaringan Anda sesuai dengan konektivitas yang ditarget.
Network Access Analyzer	Network Access Analyzer membantu Anda memahami akses jaringan ke sumber daya Anda. Anda dapat menggunakan Network Access Analyzer untuk menentukan persyaratan akses jaringan Anda serta mengidentifikasi jalur jaringan yang berpotensi tidak memenuhi persyaratan yang Anda tentukan. Dengan mengoptimalkan konfigurasi jaringan Anda yang bersangkutan, Anda dapat memahami dan memverifikasi status jaringan Anda dan menunjukkan apakah jaringan Anda di AWS memenuhi persyaratan kepatuhan Anda.

Alat AWS	Di mana harus menggunakan
Amazon CloudWatch	Gunakan konfigurasi Amazon CloudWatch dan aktifkan metrik yang sesuai untuk opsi jaringan. Pastikan Anda memilih metrik jaringan yang tepat untuk beban kerja Anda. Contohnya, Anda dapat mengaktifkan metrik untuk Penggunaan Alamat Jaringan VPC, Gateway NAT VPC, AWS Transit Gateway, terowongan VPN, AWS Network Firewall, Elastic Load Balancing, dan AWS Direct Connect. Terus-menerus memantau metrik merupakan praktik yang bagus untuk mengamati dan memahami penggunaan dan status jaringan Anda, yang membantu Anda mengoptimalkan konfigurasi jaringan berdasarkan pengamatan Anda.
AWS Network Manager	Dengan menggunakan AWS Network Manager, Anda dapat memantau kinerja waktu nyata dan historis dari Jaringan Global AWS untuk tujuan operasional dan perencanaan. Network Managermenyediakan latensi jaringan agregat antara Wilayah AWS dan Zona Ketersediaan dan dalam setiap Zona Ketersediaan, sehingga Anda dapat lebih memahami bagaimana performa aplikasi Anda terkait dengan performa jaringan AWS yang mendasarinya.
Amazon CloudWatch RUM	Gunakan Amazon CloudWatch RUM untuk mengumpulkan metrik yang memberi Anda wawasan yang membantu Anda mengidentifikasi, memahami, dan meningkatkan pengalaman pengguna.

- Identifikasikan sumber data terbesar dan pola lalu lintas aplikasi menggunakan VPC dan AWS Transit Gateway Flow Logs.
- Nilai dan optimalkan arsitektur jaringan Anda saat ini termasuk VPC, subnet, dan perutean. Sebagai contoh, Anda dapat mengevaluasi bagaimana AWS Transit Gateway atau peering VPC yang berbeda dapat membantu Anda meningkatkan jaringan dalam arsitektur Anda.
- Nilai jalur perutean di jaringan Anda untuk memastikan digunakannya jalur terpendek antartujuan. Network Access Analyzer dapat membantu Anda melakukannya.

Sumber daya

Dokumen terkait:

- Logging kueri DNS publik
- Apa itu IPAM?
- Apa itu Reachability Analyzer?
- Apa itu Network Access Analyzer?
- Metrik CloudWatch untuk VPC Anda
- Mengoptimalkan performa dan mengurangi biaya untuk analitik jaringan dengan VPC Flow Logs dalam format Apache Parquet
- Memantau jaringan global dan inti Anda dengan metrik Amazon CloudWatch
- Memantau sumber daya dan lalu lintas jaringan terus-menerus

Video terkait:

- AWS re:Invent 2023 – Panduan jaringan cloud bagi pengembang
- AWS re:Invent 2023 – Siap dengan yang berikutnya? Merancang jaringan untuk pertumbuhan dan fleksibilitas
- AWS re:Invent 2023 – Desain VPC tingkat lanjut dan kemampuan baru
- AWS re:Invent 2022 – Memahami infrastruktur jaringan AWS
- AWS re:Invent 2020 – Praktik terbaik dan kiat jaringan dengan Kerangka Kerja AWS Well-Architected
- AWS re:Invent 2020 – Memantau dan memecahkan masalah lalu lintas jaringan

Contoh terkait:

- Lokakarya Jaringan AWS
- Pemantauan Jaringan AWS
- Mengamati dan mendiagnosis jaringan Anda di AWS
- Menemukan dan mengatasi kesalahan konfigurasi jaringan pada AWS

Proses dan budaya

Saat merancang beban kerja, ada prinsip dan praktik yang dapat Anda adopsi untuk membantu Anda menjalankan beban kerja cloud berkinerja tinggi yang efisien dengan lebih baik. Area fokus ini menawarkan praktik terbaik untuk membantu mengadopsi budaya yang mendorong efisiensi kinerja beban kerja cloud.

Pertimbangkan prinsip-prinsip utama berikut untuk membangun budaya ini:

- Infrastruktur sebagai kode: Tetapkan infrastruktur Anda sebagai kode menggunakan pendekatan seperti templat AWS CloudFormation. Penggunaan templat memungkinkan Anda untuk menempatkan infrastruktur di kontrol sumber bersama dengan konfigurasi dan kode aplikasi Anda. Ini memungkinkan Anda untuk menerapkan praktik yang sama yang Anda gunakan untuk mengembangkan perangkat lunak di infrastruktur Anda sehingga Anda dapat mengulang dengan cepat.
- Alur deployment: Gunakan alur integrasi berkelanjutan/deployment berkelanjutan (CI/CD) (misalnya, tempat penyimpanan kode sumber, sistem pembangunan, deployment, dan otomatisasi pengujian) untuk melakukan deployment infrastruktur Anda. Ini memungkinkan Anda untuk melakukan deployment dengan cara yang dapat diulang, konsisten, dan murah saat Anda melakukan pengulangan.
- Metrik yang ditetapkan dengan baik: Atur dan pantau metrik untuk mencatat indikator performa utama (KPI). Kami menyarankan Anda menggunakan metrik teknis dan metrik bisnis. Untuk situs web atau aplikasi seluler, metrik utama menangkap waktu ke bita pertama atau rendering. Metrik lain yang umumnya berlaku antara lain, hitungan thread, laju pengumpulan sampah, dan keadaan tunggu. Metrik bisnis, seperti biaya kumulatif agregat per permintaan, dapat memberikan peringatan kepada Anda tentang berbagai cara untuk menghemat biaya. Pertimbangkan dengan hati-hati bagaimana Anda akan menafsirkan metrik. Misalnya, Anda dapat memilih nilai maksimum atau persentil 99 dan bukannya nilai rata-rata.
- Lakukan uji performa secara otomatis: Sebagai bagian dari proses deployment Anda, otomatis mulai uji performa setelah lulus pengujian yang lebih cepat. Otomatisasi harus menciptakan lingkungan baru, menyiapkan kondisi awal seperti data uji, kemudian jalankan serangkaian uji beban dan tolok ukur. Hasil dari pengujian-pengujian ini harus dikaitkan kembali dengan pembangunan sehingga Anda dapat melacak perubahan performa seiring waktu. Untuk pengujian yang lama, Anda dapat membuat ini sebagai bagian dari alur yang asinkron dari sisa pembangunan. Atau, Anda dapat menjalankan uji performa semalaman menggunakan Instans Spot Amazon EC2.

- Pembuatan beban: Anda harus membuat serangkaian skrip pengujian yang mereplikasi perjalanan pengguna sintetis atau tercatat sebelumnya. Skrip ini harus idempoten dan tidak digabungkan, dan Anda mungkin perlu menyertakan pra-pemanasan skrip untuk menghasilkan hasil yang valid. Sejauh dapat dilakukan, skrip pengujian Anda harus mereplikasi perilaku penggunaan dalam produksi. Anda dapat menggunakan solusi perangkat lunak sebagai layanan (SaaS) atau perangkat lunak untuk membuat beban. Pertimbangkan untuk menggunakan solusi AWS Marketplace dan Instans Spot — hal-hal tersebut bisa menjadi cara yang hemat biaya untuk menghasilkan beban.
- Visibilitas performa: Metrik utama harus dapat dilihat oleh tim Anda, khususnya metrik untuk setiap versi pembangunan. Ini memungkinkan Anda untuk melihat setiap tren positif atau negatif yang signifikan seiring waktu. Anda juga harus menampilkan metrik atas jumlah kesalahan atau pengecualian untuk memastikan Anda menguji sistem yang berfungsi.
- Visualisasi: Gunakan teknik visualisasi yang membuat jelas di mana terjadi masalah performa, hotspot, keadaan tunggu, atau penggunaan rendah. Lapisi diagram arsitektur dengan metrik performa — kode atau grafik panggilan dapat membantu mengidentifikasi masalah dengan cepat.
- Proses peninjauan rutin: Arsitektur dengan performa buruk biasanya merupakan akibat dari tidak adanya proses peninjauan performa, atau proses peninjauan performa yang bermasalah. Jika arsitektur Anda memiliki performa buruk, implementasi proses peninjauan performa memungkinkan Anda untuk mendorong peningkatan berulang.
- Optimisasi berkelanjutan: Adopsi budaya untuk terus mengoptimalkan efisiensi kinerja beban kerja cloud Anda.

Praktik terbaik

- PERF05-BP01 Membuat indikator kinerja utama (KPI) untuk mengukur kesehatan dan kinerja beban kerja
- PERF05-BP02 Menggunakan solusi pemantauan untuk memahami area dengan kinerja paling penting
- PERF05-BP03 Menetapkan proses untuk meningkatkan kinerja beban kerja
- PERF05-BP04 Menguji beban untuk beban kerja Anda
- PERF05-BP05 Menggunakan otomatisasi untuk secara proaktif memulihkan masalah terkait kinerja
- PERF05-BP06 Menjaga kemutakhiran beban kerja dan layanan Anda
- PERF05-BP07 Meninjau metrik dalam interval yang selaras

PERF05-BP01 Membuat indikator kinerja utama (KPI) untuk mengukur kesehatan dan kinerja beban kerja

Identifikasi KPI yang secara kuantitatif dan kualitatif mengukur kinerja beban kerja. KPI membantu Anda mengukur kesehatan dan kinerja beban kerja yang terkait dengan tujuan bisnis.

Antipola umum:

- Anda hanya memantau metrik tingkat sistem untuk memperoleh wawasan tentang beban kerja Anda dan tidak memahami dampak bisnis pada metrik-metrik tersebut.
- Anda berasumsi bahwa KPI Anda sudah dipublikasikan dan dibagikan sebagai data metrik standar.
- Anda tidak menetapkan KPI kuantitatif yang dapat diukur.
- Anda tidak menyelaraskan KPI dengan tujuan atau strategi bisnis.

Manfaat menerapkan praktik terbaik ini: Mengidentifikasi KPI tertentu yang mewakili kesehatan dan performa beban kerja dapat membantu menyelaraskan tim pada prioritas mereka dan menentukan hasil bisnis yang sukses. Ketika metrik-metrik tersebut kepada semua departemen, akan ada visibilitas dan kesepakatan tentang ambang batas, harapan, dan dampak bisnis.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

KPI memungkinkan tim bisnis dan rekayasa untuk menyepakati pengukuran tujuan dan strategi serta bagaimana faktor-faktor tersebut bekerja bersama untuk menciptakan hasil bisnis. Misalnya, beban kerja situs web mungkin menggunakan waktu muat halaman sebagai indikasi kinerja secara keseluruhan. Metrik ini adalah salah satu dari beberapa poin data yang mengukur pengalaman pengguna. Selain mengidentifikasi ambang batas waktu muat halaman, Anda harus mendokumentasikan hasil yang diharapkan atau risiko bisnis yang diperkirakan jika kinerja ideal tidak dipenuhi. Waktu muat halaman yang lama memengaruhi pengguna akhir Anda secara langsung, mengurangi tingkat pengalaman pengguna mereka, dan dapat menyebabkan hilangnya pelanggan. Saat Anda menetapkan ambang batas KPI Anda, gabungkan ambang batas industri serta harapan pengguna akhir Anda. Misalnya, jika ambang batas industri saat ini adalah halaman web dimuat dalam waktu dua detik, tetapi pengguna akhir Anda mengharapkan halaman web dimuat dalam waktu satu detik, maka Anda harus mempertimbangkan kedua poin data ini ketika menetapkan KPI.

Tim Anda harus mengevaluasi KPI beban kerja Anda menggunakan data granular waktu nyata dan data historis sebagai rujukan dan membuat dasbor yang menjalankan penghitungan metrik pada data KPI Anda untuk menghasilkan wawasan operasi dan pemanfaatan. KPI harus didokumentasikan dan mencakup ambang batas yang disepakati yang mendukung tujuan, dan harus dipetakan ke metrik-metrik yang dipantau. KPI harus ditinjau ulang ketika tujuan bisnis, strategi, dan kebutuhan pengguna akhir berubah.

Langkah implementasi

- Identifikasi pemangku kepentingan: Identifikasi dan dokumentasikan pemangku kepentingan bisnis utama, termasuk tim pengembangan dan operasi.
- Tentukan sasaran: Bekerjalah dengan para pemangku kepentingan ini untuk menentukan dan mendokumentasikan sasaran beban kerja Anda. Pertimbangkan aspek-aspek kinerja penting beban kerja Anda, seperti throughput, waktu respons, dan biaya, serta tujuan bisnis, seperti kepuasan pengguna.
- Tinjau praktik terbaik industri: Tinjau praktik terbaik industri untuk mengidentifikasi KPI relevan yang diselaraskan dengan sasaran beban kerja Anda.
- Identifikasi metrik: Identifikasi metrik yang selaras dengan sasaran beban kerja Anda dan dapat membantu Anda mengukur kinerja dan tujuan bisnis. Tetapkan KPI berdasarkan metrik-metrik tersebut. Contoh metrik adalah pengukuran seperti waktu respons rata-rata atau jumlah pengguna serentak.
- Tentukan dan dokumentasikan KPI: Gunakan praktik terbaik industri dan sasaran beban kerja Anda untuk menetapkan target KPI beban kerja Anda. Gunakan informasi ini untuk mengatur ambang batas KPI untuk tingkat keparahan atau alarm. Identifikasi dan dokumentasikan risiko dan dampak jika suatu KPI tidak terpenuhi.
- Implementasikan pemantauan: Gunakan alat pemantauan seperti Amazon CloudWatch atau AWS Config untuk mengumpulkan metrik dan mengukur KPI.
- Komunikasikan KPI secara visual: Gunakan alat dasbor seperti Amazon QuickSight untuk memvisualisasikan dan mengomunikasikan KPI dengan pemangku kepentingan.
- Analisis dan optimalkan: Tinjau dan analisis KPI secara rutin untuk mengidentifikasi area beban kerja Anda yang perlu diperbaiki. Bekerjalah dengan para pemangku kepentingan untuk mengimplementasikan perbaikan tersebut.
- Tinjau ulang dan sempurnakan: Tinjau metrik dan KPI secara rutin untuk menilai efektivitasnya, terutama ketika tujuan bisnis atau kinerja beban kerja berubah.

Sumber daya

Dokumen terkait:

- Dokumentasi CloudWatch
- AWS Partner Pemantauan, Pencatatan Log, dan Kinerja
- Alat observabilitas AWS
- Pentingnya Indikator Kinerja Utama (KPI) untuk Migrasi Cloud Berskala Besar
- Cara melacak KPI optimisasi biaya Anda dengan Dasbor KPI
- Dokumentasi X-Ray
- Menggunakan dasbor Amazon CloudWatch
- KPI Amazon QuickSight

Video terkait:

- AWS re:Invent 2023 - Mengoptimalkan biaya dan kinerja serta melacak kemajuan menuju mitigasi
- AWS re:Invent 2023 - Kelola peristiwa siklus hidup sumber daya dalam skala besar dengan AWS Health
- AWS re:Invent 2023 - Kinerja & efisiensi di Pinterest: Mengoptimalkan instans terbaru
- AWS re:Invent 2022 - Optimisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung
- AWS re:Invent 2023 - Membangun strategi observabilitas yang efektif
- AWS Summit SF 2022 - Pemantauan aplikasi dan observabilitas tumpukan penuh dengan AWS
- AWS re:Invent 2023 - Penskalaan di AWS untuk 10 juta pengguna pertama
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- Creating an Effective Metrics Strategy for Your Business | AWS Events

Contoh terkait:

- Membuat dasbor dengan Amazon QuickSight

PERF05-BP02 Menggunakan solusi pemantauan untuk memahami area dengan kinerja paling penting

Pahami dan identifikasi area di mana peningkatan kinerja beban kerja akan memiliki dampak positif pada efisiensi atau pengalaman pelanggan. Contohnya, situs web yang memiliki banyak interaksi pelanggan dapat memperoleh manfaat dari penggunaan layanan edge untuk memindahkan penyampaian konten lebih dekat ke pelanggan.

Antipola umum:

- Anda berasumsi bahwa metrik komputasi standar seperti penggunaan CPU atau tekanan memori sudah cukup untuk menemukan masalah kinerja.
- Anda hanya menggunakan metrik default yang dicatat oleh perangkat lunak pemantauan Anda yang dipilih.
- Anda hanya meninjau metrik ketika terdapat masalah.

Manfaat menerapkan praktik terbaik ini: Pemahaman tentang area yang memerlukan kinerja tinggi membantu para pemilik beban kerja dalam memantau KPI dan memprioritaskan peningkatan berdampak tinggi.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak dijalankan: Tinggi

Panduan implementasi

Siapkan pelacakan menyeluruh untuk mengidentifikasi pola lalu lintas, latensi, dan area kinerja kritis. Pantau pola akses data Anda untuk kueri yang lambat atau data dengan fragmentasi dan partisi yang buruk. Identifikasi area beban kerja terbatas menggunakan pengujian atau pemantauan beban.

Tingkatkan efisiensi kinerja dengan memahami arsitektur, pola lalu lintas, dan pola akses data Anda, serta identifikasi latensi dan waktu pemrosesan Anda. Identifikasi potensi hambatan yang bisa memengaruhi pengalaman pelanggan selama beban kerja berkembang. Setelah menginvestigasi area-area tersebut, lihat solusi mana yang dapat Anda deploy untuk menghilangkan masalah kinerja tersebut.

Langkah implementasi

- Siapkan pemantauan menyeluruh untuk mengetahui semua komponen dan metrik beban kerja. Berikut adalah contoh solusi pemantauan di AWS.

Service	Where to use
Pemantauan Pengguna Nyata (RUM) Amazon CloudWatch	To capture application performance metrics from real user client-side and frontend sessions.
AWS X-Ray	To trace traffic through the application layers and identify latency between components and dependencies. Use X-Ray service maps to see relationships and latency between workload components.
Wawasan Kinerja Amazon Relational Database Service	To view database performance metrics and identify performance improvements.
Pemantauan yang Ditingkatkan Amazon RDS	To view database OS performance metrics.
Amazon DevOps Guru	To detect abnormal operating patterns so you can identify operational issues before they impact your customers.

- Lakukan pengujian untuk membuat metrik, mengidentifikasi pola lalu lintas, hambatan, dan area kinerja kritis. Berikut adalah beberapa contoh cara melakukan pengujian:
  - Siapkan CloudWatch Synthetic Canaries untuk meniru aktivitas pengguna berbasis browser secara terprogram menggunakan ekspresi tingkat dan tugas cron Linux untuk menghasilkan metrik yang konsisten dari waktu ke waktu.
  - Gunakan solusi Pengujian Beban Terdistribusi AWS untuk menghasilkan lalu lintas puncak atau menguji beban kerja pada tingkat pertumbuhan yang diharapkan.
- Evaluasi metrik dan telemetri untuk mengidentifikasi area kinerja kritis Anda. Tinjau area-area ini dengan tim Anda untuk mendiskusikan pemantauan dan solusi untuk menghindari hambatan.
- Lakukan eksperimen dengan peningkatan kinerja serta ukur perubahannya dengan data. Sebagai contoh, Anda dapat menggunakan CloudWatch Evidently untuk menguji peningkatan baru dan dampak kinerja terhadap beban kerja Anda.

Sumber daya

Dokumen terkait:

- Apa yang baru di Observabilitas AWS pada re:Invent 2023
- Amazon Builders' Library
- Dokumentasi X-Ray
- RUM Amazon CloudWatch
- Amazon DevOps Guru

Video terkait:

- AWS re:Invent 2023 - [PELUNCURAN] Pemantauan aplikasi untuk beban kerja modern
- AWS re:Invent 2023 - Mengimplementasikan observabilitas aplikasi
- AWS re:Invent 2023 - Membangun strategi observabilitas yang efektif
- AWS Summit SF 2022 - Pemantauan aplikasi dan observabilitas tumpukan penuh dengan AWS
- AWS re:Invent 2022 - Optimisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung
- AWS re:Invent 2022 - Pustaka Amazon Builders: 25 tahun keunggulan operasional Amazon
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- Pemantauan Visual Aplikasi dengan Amazon CloudWatch Synthetics

Contoh terkait:

- Ukur waktu pemuatan halaman dengan Amazon CloudWatch Synthetics
- Klien Web RUM Amazon CloudWatch
- X-Ray SDK untuk Python
- Pengujian Beban Terdistribusi di AWS

PERF05-BP03 Menetapkan proses untuk meningkatkan kinerja beban kerja

Menetapkan proses untuk mengevaluasi layanan, pola desain, tipe sumber daya, dan konfigurasi baru saat sudah tersedia. Misalnya, jalankan pengujian kinerja yang sudah ada pada penawaran instans baru untuk menentukan potensinya untuk beban kerja Anda.

Antipola umum:

- Anda berasumsi bahwa arsitektur Anda saat ini statis dan tidak akan diperbarui dari waktu ke waktu.
- Anda memperkenalkan metrik arsitektur seiring waktu tanpa justifikasi metrik.

Manfaat menerapkan praktik terbaik ini: Dengan menetapkan proses Anda untuk membuat perubahan arsitektur, Anda dapat menggunakan data yang terkumpul untuk memengaruhi desain beban kerja Anda dari waktu ke waktu.

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

Panduan implementasi

Kinerja beban kerja Anda memiliki beberapa kendala utama. Dokumentasikan kendala-kendala tersebut untuk mengetahui jenis inovasi apa saja yang mungkin meningkatkan kinerja beban kerja Anda. Gunakan informasi ini ketika mempelajari layanan atau teknologi baru ketika sudah tersedia untuk mengidentifikasi cara-cara untuk menghilangkan kendala atau bottleneck.

Identifikasi kendala kinerja utama untuk beban kerja Anda. Dokumentasikan kendala performa beban kerja Anda sehingga Anda tahu jenis-jenis inovasi apa yang dapat meningkatkan performa beban kerja Anda.

Langkah implementasi

- Identifikasi KPI: Identifikasi KPI kinerja beban kerja Anda seperti yang diuraikan dalam PERF05-BP01 Membuat indikator kinerja utama (KPI) untuk mengukur kesehatan dan kinerja beban kerja untuk menjadi garis acuan beban kerja Anda.
- Implementasikan pemantauan: Gunakan alat observabilitas AWS untuk mengumpulkan metrik kinerja dan mengukur KPI.

- Lakukan analisis: Lakukan analisis mendalam untuk mengidentifikasi area (seperti konfigurasi dan kode aplikasi) di dalam beban kerja Anda yang berkinerja buruk seperti yang diuraikan dalam PERF05-BP02 Menggunakan solusi pemantauan untuk memahami area dengan kinerja paling penting. Gunakan alat analisis dan kinerja Anda untuk mengidentifikasi strategi perbaikan kinerja.
- Validasi perbaikan: Gunakan sandbox atau lingkungan praproduksi untuk memvalidasi efektivitas strategi perbaikan.
- Implementasikan perubahan: Implementasikan perubahan dalam produksi dan terus pantau kinerja beban kerja. Dokumentasikan perbaikan, dan komunikasikan perubahan kepada para pemangku kepentingan.
- Tinjau ulang dan sempurnakan: Tinjau proses peningkatan kinerja Anda secara rutin untuk mengidentifikasi area yang dapat disempurnakan.

Sumber daya

Dokumen terkait:

- Blog AWS
- Apa yang Baru dengan AWS
- AWS Skill Builder

Video terkait:

- AWS re:Invent 2022 - Menghadirkan arsitektur berkelanjutan dan berkinerja tinggi
- AWS re:Invent 2023 - Mengoptimalkan biaya dan kinerja serta melacak kemajuan menuju mitigasi
- AWS re:Invent 2022 - Optimisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung
- AWS re:Invent 2022 - Optimalkan beban kerja AWS Anda dengan panduan praktik terbaik

Contoh terkait:

- Github AWS

PERF05-BP04 Menguji beban untuk beban kerja Anda

Uji beban untuk beban kerja Anda untuk memverifikasi bahwa beban kerja Anda dapat menangani beban produksi dan mengidentifikasi kemacetan kinerja apa pun.

Antipola umum:

- Anda melakukan uji beban bagian beban kerja secara terpisah-pisah, bukan seluruh beban kerja.
- Anda melakukan uji beban pada infrastruktur yang tidak sama dengan lingkungan produksi Anda.
- Anda hanya melakukan pengujian beban pada beban yang diharapkan, tidak lebih, untuk membantu memperkirakan area yang mungkin akan bermasalah di masa depan.
- Anda melakukan pengujian beban tanpa meninjau Kebijakan Pengujian Amazon EC2 dan mengirimkan Formulir Pengajuan Peristiwa Simulasi. Ini mengakibatkan pengujian Anda gagal dijalankan, karena terlihat seperti peristiwa penolakan layanan.

Manfaat menerapkan praktik terbaik ini: Mengukur kinerja Anda dalam uji beban akan menunjukkan di mana Anda akan terdampak saat beban meningkat. Hal ini bisa memberi Anda kemampuan untuk mengantisipasi perubahan yang diperlukan sebelum berdampak pada beban kerja Anda.

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

Panduan implementasi

Pengujian beban di cloud adalah proses untuk mengukur kinerja beban kerja cloud dalam kondisi realistis dengan beban pengguna yang diharapkan. Proses ini melibatkan penyediaan lingkungan cloud mirip produksi, penggunaan alat pengujian beban untuk menghasilkan beban, dan analisis metrik untuk menilai kemampuan penanganan beban kerja Anda yang realistis. Uji beban harus dijalankan menggunakan versi data produksi yang sintetis atau sudah dibersihkan (menghapus informasi sensitif atau pengidentifikasi). Lakukan uji beban secara otomatis sebagai bagian dari pipeline pengiriman Anda, dan bandingkan hasilnya terhadap KPI dan ambang batas yang telah ditentukan sebelumnya. Proses ini membantu Anda terus mencapai kinerja yang dibutuhkan.

Langkah implementasi

- Tentukan tujuan pengujian Anda: Identifikasi aspek kinerja beban kerja Anda yang ingin Anda evaluasi, seperti throughput dan waktu respons.
- Pilih alat pengujian: Pilih dan konfigurasikan alat pengujian beban yang sesuai dengan beban kerja Anda.

- Siapkan lingkungan Anda: Siapkan lingkungan pengujian berdasarkan lingkungan produksi Anda. Anda dapat menggunakan layanan AWS untuk menjalankan lingkungan skala produksi untuk menguji arsitektur Anda.
- Implementasikan pemantauan: Gunakan alat pemantauan seperti Amazon CloudWatch untuk mengumpulkan metrik di seluruh sumber daya di arsitektur Anda. Anda juga dapat mengumpulkan dan menerbitkan metrik kustom.
- Tentukan skenario: Tentukan skenario dan parameter pengujian beban (seperti durasi pengujian dan jumlah pengguna).
- Lakukan pengujian beban: Lakukan skenario pengujian dalam skala besar. Manfaatkan AWS Cloud untuk menguji beban kerja Anda untuk mengetahui di mana letak kesalahan penskalaannya, atau apakah penskalaannya berada di jalur nonlinier. Misalnya, gunakan Instans Spot untuk menghasilkan beban dengan biaya rendah dan temukan hambatan sebelum dialami di lingkungan produksi.
- Analisis hasil pengujian: Analisis hasil untuk mengidentifikasi hambatan kinerja dan area untuk perbaikan.
- Dokumentasikan dan bagikan temuan: Dokumentasikan dan laporkan temuan serta rekomendasi. Bagikan informasi ini kepada pemangku kepentingan untuk membantu mereka mengambil keputusan yang cerdas mengenai strategi optimisasi kinerja.
- Lakukan iterasi terus-menerus: Pengujian beban harus dilakukan dengan frekuensi rutin, terutama setelah perubahan pembaruan sistem.

Sumber daya

Dokumen terkait:

- RUM Amazon CloudWatch
- Amazon CloudWatch Synthetics
- Pengujian Beban Terdistribusi di AWS

Video terkait:

- AWS Summit ANZ 2023: Lakukan akselerasi dengan percaya diri melalui Pengujian Beban Terdistribusi AWS
- AWS re:Invent 2022 - Penskalaan di AWS untuk 10 juta pengguna pertama Anda

- Memecahkan Masalah dengan Solusi AWS: Pengujian Beban Terdistribusi

- AWS re:Invent 2021 - Mengoptimalkan aplikasi melalui wawasan pengguna akhir dengan Amazon CloudWatch RUM
- Demo Amazon CloudWatch Synthetics

Contoh terkait:

- Pengujian Beban Terdistribusi di AWS

PERF05-BP05 Menggunakan otomatisasi untuk secara proaktif memulihkan masalah terkait kinerja

Gunakan indikator kinerja utama (KPI), yang digabungkan dengan sistem pamantauan dan peringatan, untuk menangani masalah terkait kinerja secara proaktif.

Antipola umum:

- Anda hanya membekali staf operasional dengan kemampuan untuk membuat perubahan operasional pada beban kerja.
- Anda membiarkan semua alarm disaring ke tim operasi tanpa perbaikan proaktif.

Manfaat menerapkan praktik terbaik ini: Perbaikan tindakan alarm yang proaktif memungkinkan staf dukungan untuk berkonsentrasi pada item-item yang tidak dapat ditindaklanjuti secara otomatis. Ini membantu staf operasi menangani semua alarm tanpa kewalahan dan mereka hanya berkonsentrasi pada alarm yang kritis.

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

Panduan implementasi

Gunakan alarm untuk memicu tindakan otomatis untuk memperbaiki masalah ketika memungkinkan. Teruskan alarm ke personel yang mampu merespons jika respons otomatis tidak memungkinkan. Misalnya, Anda mungkin memiliki sistem yang dapat memprediksi nilai dan alarm indikator kinerja utama (KPI) yang diharapkan ketika melanggar ambang batas tertentu, atau alat yang dapat

menghentikan atau membatalkan deployment secara otomatis jika KPI berada di luar nilai yang diharapkan.

Implementasikan proses yang menyediakan visibilitas tentang kinerja saat beban kerja Anda berjalan. Bangun dasbor pemantauan dan buat norma acuan untuk harapan kinerja guna menentukan apakah beban kerja berkinerja secara optimal.

Langkah implementasi

- Identifikasi alur kerja perbaikan: Identifikasi dan pahami masalah kinerja yang dapat diperbaiki secara otomatis. Gunakan solusi pemantauan AWS seperti Amazon CloudWatch atau AWS X-Ray untuk membantu Anda lebih memahami akar penyebab masalah.
- Tentukan proses otomatisasi: Buat proses perbaikan langkah demi langkah yang dapat digunakan untuk memperbaiki masalah secara otomatis.
- Konfigurasikan peristiwa inisiasi: Konfigurasikan peristiwa agar memulai proses perbaikan secara otomatis. Misalnya, Anda dapat menentukan pemicu untuk memulai ulang instans secara otomatis ketika mencapai ambang batas pemanfaatan CPU tertentu.
- Otomatiskan perbaikan: Gunakan layanan dan teknologi AWS untuk mengotomatiskan proses perbaikan. Misalnya, Otomatisasi AWS Systems Manager menyediakan cara yang aman dan dapat diskalakan untuk mengotomatiskan proses perbaikan. Pastikan menggunakan logika pemulihan mandiri untuk mengembalikan perubahan jika masalah tidak berhasil diselesaikan.
- Uji alur kerja Uji proses perbaikan otomatis di lingkungan praproduksi.
- Implementasikan alur kerja: Implementasikan perbaikan otomatis di lingkungan produksi.
- Kembangkan playbook: Kembangkan dan dokumentasikan playbook yang menguraikan langkah-langkah untuk rencana perbaikan, termasuk peristiwa inisiasi, logika perbaikan, dan tindakan yang dilakukan. Pastikan melatih pemangku kepentingan untuk membantu mereka merespons peristiwa perbaikan otomatis secara efektif.
- Tinjau dan sempurnakan: Secara rutin nilai efektivitas alur kerja perbaikan otomatis. Sesuaikan peristiwa inisiasi dan logika perbaikan jika perlu.

Sumber daya

Dokumen terkait:

- Dokumentasi CloudWatch
- Partner AWS Partner Network Pemantauan, Pencatatan Log, dan Kinerja

- Dokumentasi X-Ray
- Menggunakan Alarm dan Tindakan Alarm di CloudWatch
- Membangun Praktik Otomatisasi Cloud untuk Keunggulan Operasional: Praktik Terbaik dari AWS Managed Services
- Otomatiskan penyesuaian kinerja Amazon Redshift Anda dengan optimisasi tabel otomatis

Video terkait:

- AWS re:Invent 2023 - Strategi untuk penskalaan otomatis, perbaikan, dan pemulihan mandiri yang cerdas
- AWS re:Invent 2023 - [PELUNCURAN] Pemantauan aplikasi untuk beban kerja modern
- AWS re:Invent 2023 - Mengimplementasikan observabilitas aplikasi
- AWS re:Invent 2021 - Mengotomatiskan operasi cloud secara cerdas
- AWS re:Invent 2022 - Menyiapkan kontrol dalam skala besar di lingkungan AWS Anda
- AWS re:Invent 2022 - Mengotomatiskan manajemen dan kepatuhan patch menggunakan AWS
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- AWS re:Invent 2023 - Lepaskan beban: Lakukan diagnosis & selesaikan masalah kinerja dengan Amazon RDS
- AWS re:Invent 2021 - {Peluncuran Baru} Secara otomatis mendeteksi dan menyelesaikan masalah dengan Amazon DevOps Guru
- AWS re:Invent 2023 - Sentralisasikan operasi Anda

Contoh terkait:

- CloudWatch Logs Kustomisasi Alarm

PERF05-BP06 Menjaga kemutakhiran beban kerja dan layanan Anda

Terus ikuti informasi tentang layanan dan fitur cloud baru untuk mengadopsi fitur yang efisien, menghilangkan masalah, dan meningkatkan efisiensi kinerja beban kerja Anda secara keseluruhan.

Antipola umum:

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

Manfaat menjalankan praktik terbaik ini: Dengan menetapkan proses untuk tetap mutakhir pada layanan dan penawaran baru, Anda dapat menerapkan fitur dan kemampuan baru, menyelesaikan masalah, dan meningkatkan kinerja beban kerja.

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

Panduan implementasi

Evaluasi cara meningkatkan performa saat layanan, pola desain, dan fitur produk baru tersedia. Tentukan mana hal-hal yang dapat meningkatkan kinerja atau menambah efisiensi beban kerja melalui evaluasi, diskusi internal, atau analisis eksternal. Tentukan proses untuk mengevaluasi pembaruan, fitur baru, dan layanan yang relevan dengan beban kerja Anda. Misalnya, bangun bukti konsep yang memanfaatkan teknologi baru atau berkonsultasi dengan grup internal. Saat mencoba layanan atau ide baru, jalankan pengujian kinerja untuk mengukur pengaruhnya terhadap kinerja beban kerja.

Langkah implementasi

- Lakukan inventarisasi beban kerja: Buat inventaris perangkat lunak dan arsitektur beban kerja Anda dan identifikasi komponen yang perlu diperbarui.
- Identifikasi sumber pembaruan: Identifikasi sumber berita dan pembaruan yang terkait dengan komponen beban kerja Anda. Sebagai contoh, Anda dapat berlangganan blog What's New at AWS untuk produk yang sesuai dengan komponen beban kerja Anda. Anda dapat berlangganan umpan RSS atau mengelola langganan email Anda.
- Tentukan jadwal pembaruan: Tentukan jadwal untuk mengevaluasi layanan dan fitur baru untuk beban kerja Anda.
  - Anda dapat menggunakan AWS Systems Manager Inventory untuk mengumpulkan metadata sistem operasi (OS), aplikasi, dan instans dari instans Amazon EC2 Anda 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.
- Nilai pembaruan baru: Pahami cara memperbarui komponen beban kerja Anda. Manfaatkan ketangkasan di cloud untuk menguji dengan cepat bagaimana fitur baru dapat meningkatkan beban kerja Anda untuk mendapatkan efisiensi performa.

- Gunakan otomatisasi: Gunakan otomatisasi untuk 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 mengotomatiskan proses pembaruan sistem, dan menjadwalkan aktivitas menggunakan AWS Systems Manager Maintenance Windows.
- Dokumentasikan proses: Dokumentasikan proses Anda untuk mengevaluasi pembaruan dan layanan baru. Bekali pemilik Anda dengan waktu dan ruang yang dibutuhkan untuk meneliti, menguji, bereksperimen, serta memvalidasi pembaruan dan layanan baru. Lihat kembali persyaratan dan KPI bisnis terdokumentasi untuk membantu memprioritaskan pembaruan mana yang akan menciptakan dampak bisnis yang positif.

Sumber daya

Dokumen terkait:

- Blog AWS
- Apa yang Baru dengan AWS
- Mengimplementasikan image terbaru dengan pipeline EC2 Image Builder otomatis

Video terkait:

- AWS re:Inforce 2022 - Mengotomatiskan manajemen dan kepatuhan patch menggunakan AWS
- All Things Patch: AWS Systems Manager | AWS Events

Contoh terkait:

- Manajemen Inventaris dan Patch
- One Observability Workshop

PERF05-BP07 Meninjau metrik dalam interval yang selaras

Sebagai bagian pemeliharaan rutin, atau sebagai respons terhadap peristiwa atau insiden, tinjau metrik mana yang dikumpulkan. Gunakan tinjauan ini untuk mengidentifikasi metrik mana yang penting untuk menangani masalah dan metrik mana yang merupakan tambahan. Jika dilacak, metrik tersebut dapat memudahkan Anda mengidentifikasi, mengatasi, dan mencegah masalah.

Antipola umum:

- Anda mengizinkan metrik untuk tetap dalam status alarm selama periode waktu yang lebih lama.
- Anda memberikan alarm yang tidak dapat ditindaklanjuti oleh sistem otomatisasi.

Manfaat menerapkan praktik terbaik ini: Tinjau secara terus-menerus metrik yang dikumpulkan untuk memastikan metrik tersebut dapat mengidentifikasi, mengatasi, atau mencegah masalah. Metrik juga dapat kedaluwarsa jika Anda membiarkannya berada dalam status alarm untuk waktu yang lama.

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

Panduan implementasi

Terus-menerus tingkatkan pemantauan dan pengumpulan metrik. Sebagai bagian dari tindakan merespons insiden atau peristiwa, evaluasikan mana metrik yang berguna untuk mengatasi masalah dan mana metrik yang dapat membantu tetapi saat ini tidak terdeteksi. Gunakan metode ini untuk meningkatkan kualitas metrik yang Anda kumpulkan agar Anda dapat mencegah, atau agar dapat lebih cepat menangani, insiden pada masa mendatang.

Sebagai bagian dari tindakan merespons insiden atau peristiwa, evaluasikan mana metrik yang berguna untuk mengatasi masalah dan mana metrik yang dapat membantu tetapi saat ini tidak terdeteksi. Gunakan ini untuk meningkatkan kualitas metrik yang Anda kumpulkan agar dapat mencegah atau dapat lebih cepat mengatasi insiden di masa mendatang.

Langkah implementasi

- Tentukan metrik: Tentukan metrik kinerja penting yang perlu dipantau, yang selaras dengan tujuan beban kerja Anda, termasuk metrik seperti waktu respons dan pemanfaatan sumber daya.
- Tetapkan garis acuan: Tetapkan garis acuan dan nilai yang diinginkan untuk setiap metrik. Garis acuan harus memberikan titik-titik referensi untuk mengidentifikasi penyimpangan atau anomali.

- Atur frekuensi: Tetapkan frekuensi (seperti mingguan atau bulanan) untuk meninjau metrik-metrik penting.
- Identifikasi masalah kinerja: Dalam setiap tinjauan, lakukan penilaian tren dan penyimpangan dari nilai garis acuan. Cari setiap anomali atau hambatan performa. Untuk masalah yang teridentifikasi, lakukan analisis akar penyebab secara mendalam untuk memahami alasan utama di balik masalah tersebut.
- Identifikasi tindakan korektif: Gunakan analisis Anda untuk mengidentifikasi tindakan korektif. Tindakan tersebut antara lain penyesuaian parameter, perbaikan bug, dan penskalaan sumber daya.
- Dokumentasikan temuan: Dokumentasikan temuan Anda, termasuk masalah yang teridentifikasi, akar masalah, dan tindakan korektif.
- Lakukan iterasi dan perbaiki: Terus nilai dan perbaiki proses peninjauan metrik. Gunakan pelajaran yang dipetik dari tinjauan sebelumnya untuk menyempurnakan proses dari waktu ke waktu.

Sumber daya

Dokumen terkait:

- Dokumentasi CloudWatch
- Kumpulkan metrik dan log dari Instans Amazon EC2 serta server on-premise dengan Agen CloudWatch
- Mengueri metrik dengan Wawasan Metrik CloudWatch
- Partner AWS Partner Network Pemantauan, Pencatatan Log, dan Kinerja
- Dokumentasi X-Ray

Video terkait:

- AWS re:Invent 2022 - Menyiapkan kontrol dalam skala besar di lingkungan AWS Anda
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- AWS re:Invent 2023 - Membangun strategi observabilitas yang efektif
- AWS Summit SF 2022 - Pemantauan aplikasi dan observabilitas tumpukan penuh dengan AWS
- AWS re:Invent 2023 - Lepaskan beban: Lakukan diagnosis & selesaikan masalah kinerja dengan Amazon RDS

Contoh terkait:

- Membuat dasbor dengan Amazon QuickSight
- Dasbor CloudWatch

Kesimpulan

Untuk mencapai dan mempertahankan efisiensi kinerja, diperlukan pendekatan yang didorong data. Anda harus aktif mempertimbangkan pola akses dan kompromi yang akan memungkinkan Anda melakukan optimalisasi untuk kinerja yang lebih tinggi. Dengan menggunakan proses peninjauan berdasarkan tolok ukur dan uji beban, Anda dapat memilih tipe dan konfigurasi sumber daya yang tepat. Dengan memperlakukan infrastruktur Anda sebagai kode, Anda dapat mengembangkan arsitektur dengan cepat dan aman sambil menggunakan data untuk mengambil keputusan berbasis fakta terkait arsitektur Anda. Melakukan pemantauan aktif dan pasif secara bersamaan dapat memastikan bahwa kinerja arsitektur Anda tidak mengalami penurunan.

AWS berupaya membantu Anda membangun arsitektur yang memiliki kinerja efisien sambil menghadirkan nilai bisnis. Gunakan alat dan teknik yang dibahas dalam artikel ini untuk memastikan keberhasilan.

Kontributor

Individu dan organisasi berikut ini memiliki kontribusi dalam dokumen ini:

- Sam Mokhtari, Senior Efficiency Lead Solutions Architect, Amazon Web Services
- Josh Hart, Solutions Architect (Arsitek Solusi), Amazon Web Services
- Richard Trabing, Solutions Architect (Arsitek Solusi), Amazon Web Services
- Brett Looney, Principal Solutions Architect, Amazon Web Services
- Nina Vogl, Principal Solutions Architect, Amazon Web Services
- Eric Pullen, Solutions Architect (Arsitek Solusi), Amazon Web Services
- Julien Lépine, Specialist SA Manager (Manajer SA Spesialis), Amazon Web Services
- Ronnen Slasky, Solutions Architect (Arsitek Solusi), Amazon Web Services

Bacaan lebih lanjut

Untuk bantuan tambahan, pelajari sumber berikut:

- AWS Well-Architected Framework
- Pusat Arsitektur AWS

Revisi Dokumen

Berlangganan umpan RSS untuk memperoleh pemberitahuan tentang pembaruan laporan resmi ini.

Perubahan	Deskripsi	Tanggal
Laporan resmi diperbarui	Praktik terbaik diperbarui dengan panduan implementasi baru.	June 27, 2024
Pembaruan dan restrukturisasi besar	Pilar direstrukturisasi menjadi lima area praktik terbaik (turun dari delapan). Konten telah dikonsolidasikan ke dalam lima area dan diperbarui. Area praktik terbaik baru adalah Pemilihan arsitektur, Komputasi dan perangkat keras, Manajemen Data, Jaringan dan pengiriman konten, dan Proses dan budaya.	October 3, 2023
Pembaruan kecil	Bahasa non-inklusif dihilangkan.	April 13, 2023
Pembaruan untuk Kerangka Kerja baru	Praktik terbaik diperbarui dengan panduan preskriptif dan praktik terbaik baru ditambahkan.	April 10, 2023
Laporan resmi diperbarui	Praktik terbaik diperbarui dengan panduan implementasi baru.	December 15, 2022

Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan ditambahkan.	October 20, 2022
Pembaruan kecil	Bahasa noninklusif dihilangkan.	April 22, 2022
Pembaruan kecil	Penambahan Pilar Pelestarian Lingkungan ke pengantar.	December 2, 2021
Pembaruan kecil.	Tautan diperbarui.	March 10, 2021
Pembaruan kecil.	Waktu habis AWS Lambda diubah menjadi 900 detik dan nama Amazon Keyspaces (for Apache Cassandra) telah dikoreksi.	October 5, 2020
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	July 15, 2020
Pembaruan untuk Kerangka Kerja baru	Peninjauan dan pembaruan besar konten	July 8, 2020
Laporan resmi diperbarui	Pembaruan kecil masalah gramatikal	July 1, 2018
Laporan resmi diperbarui	Laporan resmi disegarkan untuk mencerminkan perubahan di AWS	November 1, 2017
Publikasi awal	Pilar Efisiensi Kinerja - AWS Well-Architected Framework diterbitkan.	November 1, 2016

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the Glosarium AWS Reference.