Kerangka Kerja AWS Well-Architected

Pilar Efisiensi Performa

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Pilar Efisiensi Performa: Kerangka Kerja AWS Well-Architected

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Abstrak dan pengantar 1
Pengantar 1
Efisiensi kinerja 3
Prinsip desain 3
Definisi 4
Pemilihan arsitektur 5
PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia 5
Panduan implementasi 6
Sumber daya 7
PERF01-BP02 Menggunakan panduan dari penyedia cloud Anda atau mitra yang tepat untuk mempelajari pola arsitektur dan praktik terbaik 8
Panduan implementasi 6
Sumber daya 7
PERF01-BP03 Faktor biaya ke dalam keputusan arsitektur 10
Panduan implementasi 6
Sumber daya 7
PERF01-BP04 Mengevaluasi bagaimana kompromi berdampak pada pelanggan dan efisiensi arsitektur 12
Panduan implementasi 6
Sumber daya 7
PERF01-BP05 Menggunakan kebijakan dan arsitektur referensi 14
Panduan implementasi 6
Sumber daya 7
PERF01-BP06 Menggunakan tolok ukur untuk mendorong keputusan arsitektur 16
Panduan implementasi 6
Sumber daya 7
PERF01-BP07 Menggunakan pendekatan berbasis data untuk pilihan arsitektur 18
Panduan implementasi 6
Sumber daya 7
Komputasi dan perangkat keras 21
PERF02-BP01 Memilih opsi komputasi terbaik untuk beban kerja Anda 21
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7

PERF02-BP02 Memahami konfigurasi dan fitur komputasi yang tersedia 25
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7
PERF02-BP03 Kumpulkan metrik terkait komputasi 29
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7
PERF02-BP04 Mengonfigurasi dan menyesuaikan ukuran yang tepat untuk sumber daya komputasi 31
Panduan implementasi 6
Sumber daya 7
PERF02-BP05 Menskalakan sumber daya komputasi Anda secara dinamis 34
Panduan implementasi 6
Sumber daya 7
PERF02-BP06 Menggunakan akselerator komputasi berbasis perangkat keras yang dioptimalkan 37
Panduan implementasi 6
Sumber daya 7
Manajemen data 41
PERF03-BP01 Menggunakan penyimpanan data yang dibuat khusus yang paling mendukung persyaratan akses data dan penyimpanan data Anda 41
Panduan implementasi 6
Sumber daya 7
PERF03-BP02 Lakukan evaluasi terhadap opsi konfigurasi yang tersedia untuk penyimpanan data 53
Panduan implementasi 6
Sumber daya 7
PERF03-BP03 Mengumpulkan dan merekam metrik-metrik kinerja penyimpanan data 58
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7
PERF03-BP04 Menerapkan strategi untuk meningkatkan kinerja kueri dalam penyimpanan data 61
Panduan implementasi 6
Sumber daya 7

PERF03-BP05 Mengimplementasikan pola akses data yang memanfaatkan caching 63
Panduan implementasi 6
Sumber daya 7
Jaringan dan Pengiriman Konten 67
PERF04-BP01 Memahami bagaimana jaringan memengaruhi performa 67
Panduan implementasi 6
Sumber daya 7
PERF04-BP02 Mengevaluasi fitur jaringan yang tersedia 71
Panduan implementasi 6
Sumber daya 7
PERF04-BP03 Memilih konektivitas khusus atau VPN yang tepat untuk beban kerja Anda 78
Panduan implementasi 6
Sumber daya 7
PERF04-BP04 Menggunakan penyeimbangan beban untuk mendistribusikan lalu lintas di berbagai sumber daya 81
Panduan implementasi 6
Sumber daya 7
PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa 85
Panduan implementasi 6
Sumber daya 7
PERF04-BP06 Pilih lokasi beban kerja Anda berdasarkan persyaratan jaringan 89
Panduan implementasi 6
Sumber daya 7
PERF04-BP07 Mengoptimalkan konfigurasi jaringan berdasarkan metrik 94
Panduan implementasi 6
Sumber daya 7
Proses dan budaya 100
PERF05-BP01 Membuat indikator kinerja utama (KPI) untuk mengukur kesehatan dan kinerja beban kerja 102
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7
PERF05-BP02 Menggunakan solusi pemantauan untuk memahami area-area yang mementingkan kinerja 105
Panduan implementasi 6
Sumber daya 7

PERF05-BP03 Menetapkan sebuah proses untuk meningkatkan kinerja beban kerja 108
Panduan implementasi 6
Sumber daya 7
PERF05-BP04 Uji beban kerja Anda 110
Panduan implementasi 6
Sumber daya 7
PERF05-BP05 Menggunakan otomatisasi untuk secara proaktif memulihkan masalah terkait kinerja 112
Panduan implementasi 6
Sumber daya 7
PERF05-BP06 Pertahankan beban kerja dan layanan Anda up-to-date 115
Panduan implementasi 6
Langkah-langkah implementasi 6
Sumber daya 7
PERF05-BP07 Meninjau metrik dalam interval yang selaras 117
Panduan implementasi 6
Sumber daya 7
Kesimpulan 120
Kontributor 121
Sumber bacaan lebih lanjut 122
Revisi dokumen 123
Pemberitahuan 125
AWS Glosarium 126

Pilar Efisiensi Kinerja - Kerangka Kerja AWS Well-Architected

Tanggal publikasi: 6 November 2024 (Revisi dokumen)

Laporan resmi ini berfokus pada pilar efisiensi kinerja Kerangka Kerja AWS Well-Architected. Laporan resmi ini menyediakan panduan untuk membantu pelanggan menerapkan praktik-praktik terbaik dalam desain, pengiriman, dan pemeliharaan lingkungan AWS.

Pengantar

Kerangka Kerja AWS Well-Architected akan membantu Anda mengetahui kelebihan dan kekurangan dari keputusan yang Anda ambil saat membangun beban kerja di AWS. Penggunaan Kerangka Kerja ini akan membantu Anda mempelajari praktik-praktik terbaik berkaitan dengan 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-praktik terbaik dan mengidentifikasi area yang perlu diperbaiki. Kami meyakini bahwa memiliki beban kerja yang didesain dengan baik akan meningkatkan peluang keberhasilan bisnis.

Enam pilar landasan kerangka kerja:

- Keunggulan Operasional
- Keamanan
- Keandalan
- Efisiensi Kinerja
- Pengoptimalan Biaya
- Keberlanjutan

Artikel ini berkonsentrasi 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 akan 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 (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 mencakup kemampuan untuk menggunakan sumber daya cloud secara efisien untuk memenuhi persyaratan-persyaratan kinerja, dan untuk mempertahankan efisiensi tersebut seiring dengan perubahan permintaan dan perkembangan teknologi yang terjadi.

Topik

- Prinsip desain
- Definisi

Prinsip desain

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

- Buat teknologi canggih dapat diakses oleh lebih banyak orang: Buat penerapan teknologi canggih lebih mudah dengan mendelegasikan tugas-tugas kompleks kepada penyedia layanan cloud Anda. Daripada bertanya kepada tim IT Anda tentang hosting dan menjalankan teknologi baru, manfaatkan teknologi sebagai layanan. Misalnya, Tidak ada SQL database, transcoding media, dan pembelajaran mesin adalah semua teknologi yang membutuhkan 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.
- Menjadi global dalam hitungan menit: Menyebarkan beban kerja Anda di beberapa AWS Wilayah di seluruh dunia memungkinkan Anda memberikan latensi yang lebih rendah dan pengalaman yang lebih baik bagi pelanggan Anda 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-jenis instans, penyimpanan, atau konfigurasi yang berbeda-beda.

- Pertimbangkan simpati mekanis: 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.

Meninjau pilihan Anda secara teratur, memastikan bahwa Anda memanfaatkan Cloud yang terus berkembang AWS . 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 yang dirancang dengan baik 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 Faktor biaya ke 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.

Anti-pola 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 menerapkan 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 diterapkan: 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-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 dan Sertifikasi-sertifikasi

  - Saluran YouTube AWS
  - Lokakarya AWS
  - Komunitas AWS
- Gunakan Amazon Q untuk mendapatkan informasi dan saran yang relevan tentang layanan.
- 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:

- Ikhtisar Amazon Web Services
- Fitur Amazon EC2
- Pelajari langkah demi langkah dengan Rencana Pembelajaran Mitra AWS
- Pelatihan dan Sertifikasi AWS
- 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:

- AWS Sampel
- AWS Contoh SDK

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-keputusan Anda yang berkaitan dengan arsitektur. Semua sumber daya ini membantu meninjau dan meningkatkan arsitektur Anda untuk kinerja yang optimal.

Anti-pola 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 menerapkan praktik terbaik ini: Menggunakan panduan dari sebuah penyedia cloud atau mitra 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 diterapkan: 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-program pelatihan dan sertifikasi, arsitek solusi, dan layanan profesional yang dapat membantu para pelanggan untuk 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-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 dan Sertifikasi-sertifikasi
  - Saluran YouTube AWS
- Bergabunglah dengan acara-acara mitra AWS (seperti AWS Global Summits, AWS re:Invent, grup pengguna, dan lokakarya) untuk belajar dari para ahli AWS tentang praktik-praktik terbaik untuk menggunakan layanan AWS.
  - Pelajari langkah demi langkah dengan Rencana Pembelajaran Mitra AWS
  - Acara dan Webinar AWS
  - Lokakarya AWS
  - Komunitas AWS
- Hubungi AWS untuk mendapatkan bantuan saat Anda memerlukan panduan tambahan atau informasi produk. AWS Arsitek Solusi dan Layanan Profesional AWS menyediakan panduan untuk implementasi solusi. AWS Mitra menyediakan keahlian AWS untuk membantu Anda mendapat ketangkasan dan inovasi untuk bisnis Anda.
- Gunakan Dukungan jika Anda membutuhkan dukungan teknis untuk menggunakan layanan secara efektif. Rencana Dukungan kami dirancang untuk memberikan Anda paduan alat yang tepat dan akses kepada kemahiran sehingga Anda dapat berhasil dengan AWS sambil mengoptimalkan kinerja, mengelola risiko, dan menjaga biaya tetap terkendali.

Sumber daya

Dokumen terkait:

- AWS Pusat Arsitektur
- AWS Partner Network
- AWS Pustaka Solusi
- Pusat Pengetahuan AWS
- Dukungan Perusahaan AWS

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:

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

PERF01-BP03 Faktor biaya ke 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.

Anti-pola 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 menerapkan 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 diterapkan: 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 yang tepat untuk komponen beban kerja dan menghadirkan elastisitas, yang menghasilkan peningkatan efisiensi kinerja beban kerja cloud.

Langkah-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 Kalkulator Harga dan AWS Cost Explorer untuk mengidentifikasi pendorong biaya utama dalam beban kerja Anda.
- Pahami model penetapan harga di cloud, seperti Sesuai Permintaan, Instans Terpesan, Savings Plans, dan Instans Spot.
- Gunakan praktik terbaik optimasi biaya Well-Architected untuk mengoptimalkan komponen utama 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 AWS Billing and Cost Management?
- Optimalisasi Biaya dengan AWS
- Memilih strategi manajemen AWS biaya
- Panduan Pemula untuk Manajemen AWS Biaya
- 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 optimasi biaya AWS
- AWS re:invent 2023 - Optimalkan biaya dan kinerja dan lacak kemajuan menuju mitigasi
- AWS RE: invent 2023 - AWS praktik terbaik pengoptimalan biaya penyimpanan
- AWS Re:invent 2023 - Optimalkan biaya di lingkungan multi-akun Anda

Contoh terkait:

- AWS Compute Optimizer Kode 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.

Anti-pola 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 menerapkan 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-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 Quick
- Amazon CloudWatch RUM

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

Video terkait:

- Optimalkan aplikasi melalui Amazon CloudWatch RUM
- AWS re:Invent 2023 - Kapasitas, ketersediaan, efisiensi biaya: Pilih ketiganya
- AWS re:Invent 2023 - Pola integrasi tingkat lanjut & kompromi untuk sistem yang digabungkan dengan metode penggabungan 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.

Anti-pola umum:

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

Manfaat menerapkan 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 diterapkan: 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-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.
- Ikuti up-to-date standar dan AWS pembaruan industri untuk memastikan kebijakan dan arsitektur referensi membantu mengoptimalkan beban kerja cloud Anda.

Sumber daya

Dokumen terkait:

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

Video terkait:

- Ini Arsitektur saya
- AWS Re: invent 2022 - Percepat nilai untuk bisnis Anda dengan & arsitektur referensi SAP AWS

Contoh terkait:

- Sampel AWS
- AWS SDKContoh

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.

Anti-pola 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: Melakukan tolok ukur terhadap implementasi Anda saat ini akan memungkinkan Anda untuk mengukur peningkatan kinerja yang berhasil dicapai.

Tingkat risiko yang terjadi jika praktik terbaik ini tidak diterapkan: 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-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
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- Alur kerja genomik, Bagian 5: penolokukuran otomatis
- Lakukan tolok ukur dan optimalkan deployment titik akhir di Amazon SageMaker AI JumpStart

Video terkait:

- AWS re:Invent 2023 - Penolokukuran AWS Lambda cold starts
- Penolokukuran layanan stateful di cloud
- Ini Arsitektur saya
- Optimalkan aplikasi melalui Amazon CloudWatch RUM
- Demo Amazon CloudWatch Synthetics

Contoh terkait:

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

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.

Anti-pola 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 menerapkan 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 diterapkan: 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-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).

- Konsultasikan dengan 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:

- AWS Sampel
- Contoh AWS SDK

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 Kumpulkan metrik terkait komputasi
- PERF02-BP04 Mengonfigurasi dan menyesuaikan ukuran yang tepat untuk 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.

Anti-pola 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 diterapkan: Tinggi

Panduan implementasi

Untuk mengoptimalkan beban kerja cloud Anda demi efisiensi kinerja, penting bagi Anda 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-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 tentang berbagai layanan komputasi AWS untuk beban kerja Anda. Untuk informasi selengkapnya, lihat PERF01-BP01 Mempelajari dan memahami layanan serta fitur cloud yang tersedia. Berikut adalah beberapa opsi komputasi AWS utama, karakteristiknya, dan kasus penggunaan umumnya:

AWSLayanan	Karakteristik utama	Kasus penggunaan umum
Amazon Elastic Compute Cloud (Amazon EC2)	Memiliki opsi khusus untuk perangkat keras, persyarat an lisensi, banyak pilihan rangkaian instans yang	Migrasi angkat dan geser, aplikasi monolitik, lingkungan hybrid, aplikasi perusahaan

AWSLayanan	Karakteristik utama	Kasus penggunaan umum
	berbeda, jenis prosesor, dan akselerator komputasi	
Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS)	Deployment mudah, lingkungan konsisten, dapat diskalakan	Layanan mikro, lingkungan hibrida
AWS Lambda	Layanan komputasi nirserver yang menjalankan kode sebagai respons terhadap peristiwa dan secara otomatis mengelola sumber daya komputasi yang mendasari nya.	Layanan mikro, aplikasi yang didorong peristiwa
AWS Batch	Secara efisien dan dinamis menyediakan dan menskalak an sumber daya komputasi Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernete s Service (Amazon EKS) , dan AWS Fargate, dengan opsi untuk menggunakan Instans Sesuai Permintaan atau Instans Spot berdasark an kebutuhan pekerjaan Anda	HPC, melatih model ML
Amazon Lightsail	Aplikasi Linux dan Windows yang telah dikonfigurasi sebelumnya untuk menjalank an beban kerja kecil	Aplikasi web sederhana, situs web kustom

- 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-alat pemantauan AWS seperti Amazon CloudWatch dan layanan-layanan pengoptimalan seperti AWS Compute Optimizer untuk selalu mengoptimalkan komputasi Anda berdasarkan pola penggunaan dunia nyata secara terus-menerus.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Jenis 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 optimalisasi Amazon Elastic Compute Cloud
- AWS re:Invent 2021 - Mendukung Amazon Elastic Compute Cloud generasi berikutnya: Memahami lebih dalam Sistem Nitro
- AWS re:Invent 2019 - Mengoptimalkan performa dan biaya untuk komputasi AWS Anda
- AWS re:Invent 2019 - Pondasi 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

- Landasan Amazon EC2
- Melakukan deployment model ML untuk inferensi dengan performa tinggi dan biaya rendah

Contoh terkait:

- Memigrasikan Aplikasi web ke kontainer
- Menjalankan Hello World Nirserver
- Lokakarya Amazon EKS
- Lokakarya Amazon EC2
- Beban Kerja yang Efisien dan Tangguh dengan Amazon Elastic Compute Cloud Auto Scaling
- 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.

Anti-pola umum:

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

Manfaat menerapkan 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 diterapkan: 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 seperti apa yang terbaik untuk aplikasi Anda. Contoh dari opsi tersebut meliputi keluarga 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, maka Anda dapat menggunakan AWS Compute Optimizer untuk mengetahui apakah opsi komputasi Anda yang sekarang cocok untuk beban kerja dari perspektif CPU dan memori.

Langkah-langkah implementasi

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

Opsi Konfigurasi	Contoh-contoh
Tipe instans	• Instans komputasi yang dioptimalkan adalah instans yang 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 yang intensif memori. • Instans penyimpanan yang dioptimalkan didesain untuk beban kerja yang memerlukan akses baca dan tulis sekuensial (IOPS) yang tinggi ke penyimpanan lokal.
Model penentuan harga	• Instans Sesuai Permintaan memungkinkan Anda untuk menggunakan kapasitas komputasi per jam atau per detik tanpa perlu membuat komitmen jangka panjang. Instans ini bagus untuk lonjakan di atas kebutuhan dasar kinerja.

Opsi Konfigurasi	Contoh-contoh
	• Savings Plans menawarkan Anda penghematan yang signifikan atas Instans Sesuai Permintaan dengan komitmen untuk menggunakan daya komputasi dalam jumlah tertentu selama jangka waktu satu atau tiga tahun. • Instans spot akan memungkinkan Anda untuk memanfaatkan kapasitas instans yang tidak terpakai untuk beban kerja stateless dan toleran terhadap kesalahan.
Auto Scaling	Gunakan Auto Scaling (penskalaan otomatis) untuk mencocokkan sumber daya komputasi dengan pola lalu lintas.
Penyesuaian ukuran	• Gunakan Pengoptimal Komputasi untuk mendapatkan rekomendasi berbasis machine learning mengenai konfigura si komputasi yang paling cocok dengan karakteristik komputasi yang Anda miliki. • 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 yang berbasis CPU. • Untuk beban kerja machine learning, Anda bisa memanfaatkan 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
- Jenis 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 Konsol Manajemen AWS
- AWS re:Invent 2023 – Yang baru dengan Amazon EC2
- AWS re:Invent 2023 – Penghematan cerdas: Strategi optimalisasi biaya Amazon EC2
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Memahami lebih dalam Sistem Nitro
- AWS re:Invent 2019 – Landasan Amazon EC2
- AWS re:Invent 2022 – Mengoptimalkan Amazon EKS untuk kinerja dan biaya di AWS

Contoh terkait:

- Kode demo Pengoptimal Komputasi
- 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 Kumpulkan metrik terkait komputasi

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

Anti-pola umum:

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

Manfaat menerapkan 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 untuk terus meningkatkan kinerja dan pemanfaatan sumber daya dalam beban kerja Anda.

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

Panduan implementasi

Beban kerja dapat menghasilkan data dalam jumlah besar seperti metrik, log, dan peristiwa. Dalam hal ini AWS Cloud, mengumpulkan metrik merupakan 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 berharga. Metrik seperti CPU pemanfaatan, pemanfaatan memori, disk I/O, dan inbound dan outbound jaringan dapat memberikan wawasan tentang tingkat pemanfaatan atau kemacetan kinerja. Gunakan metrik-metrik tersebut 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 tujuan-tujuan biaya dan operasional.

Langkah-langkah implementasi

- Identifikasi metrik-metrik terkait kinerja apa saja yang relevan dengan beban kerja Anda. Anda harus mengumpulkan metrik seputar pemanfaatan sumber daya dan cara beban kerja cloud Anda beroperasi (seperti waktu respons dan throughput).

  - Metrik EC2 default Amazon
  - Metrik ECS default Amazon
  - Metrik EKS default Amazon
  - Metrik default Lambda
  - EC2Memori Amazon dan metrik disk
- Pilih dan siapkan solusi pembuatan log dan pemantauan yang tepat untuk beban kerja Anda.
  - Observabilitas native AWS
  - AWS Distro untuk OpenTelemetry
  - Layanan Terkelola Amazon untuk Prometheus
- Tentukan filter dan agregasi yang diperlukan untuk metrik-metrik tersebut berdasarkan persyaratan beban kerja Anda.
  - Mengukur metrik aplikasi kustom dengan Amazon CloudWatch Logs dan filter metrik
  - Kumpulkan metrik khusus dengan penandaan CloudWatch strategis Amazon
- Konfigurasikan kebijakan-kebijakan retensi data untuk metrik Anda agar sesuai dengan tujuan-tujuan keamanan dan operasional Anda.
  - Retensi data default untuk CloudWatch metrik
  - Penyimpanan data default untuk CloudWatch Log
- Jika diperlukan, buatlah alarm dan notifikasi untuk metrik Anda agar membantu Anda dalam merespons masalah terkait kinerja secara proaktif.
  - Buat alarm untuk metrik kustom menggunakan deteksi anomali Amazon CloudWatch
  - Buat metrik dan alarm untuk halaman web tertentu dengan Amazon CloudWatch RUM
- Gunakan otomatisasi untuk melakukan deployment agen agregasi log dan metrik Anda.
  - AWS Systems Manager otomatisasi
  - OpenTelemetryKolektor

Sumber daya

Dokumen terkait:

- Pemantauan dan observabilitas
- Praktik terbaik: menerapkan observabilitas dengan AWS
- CloudWatch Dokumentasi Amazon

- Kumpulkan metrik dan log dari EC2 instans Amazon dan server lokal dengan Agen CloudWatch
- Mengakses CloudWatch Log Amazon untuk AWS Lambda
- Menggunakan CloudWatch Log dengan instance kontainer
- Menerbitkan metrik kustom
- Jawaban AWS : Pencatatan Log Terpusat
- AWS Layanan yang Mempublikasikan CloudWatch Metrik
- Memantau Amazon EKS di AWS Fargate

Video terkait:

- AWS RE: invent 2023 - [LAUNCH] Pemantauan aplikasi untuk beban kerja modern
- AWS re:invent 2023 - Menerapkan observabilitas aplikasi
- AWS re:invent 2023 — Membangun strategi observabilitas yang efektif
- AWS Re:invent 2023 - Observabilitas mulus dengan Distro untuk AWS OpenTelemetry
- Manajemen Kinerja Aplikasi pada AWS

Contoh terkait:

- AWS untuk Hari Perendaman Beban Kerja Linux- Amazon CloudWatch
- Memantau ECS cluster dan kontainer Amazon
- Pemantauan dengan CloudWatch dasbor Amazon
- EKSLokakarya Amazon

PERF02-BP04 Mengonfigurasi dan menyesuaikan ukuran yang tepat untuk sumber daya komputasi

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

Anti-pola umum:

- Anda mengabaikan persyaratan kinerja beban kerja yang mengakibatkan sumber daya komputasi mengalami pemanfaatan yang terlalu rendah atau terlalu tinggi.

- Anda hanya memilih instans terbesar atau terkecil untuk semua beban kerja.
- Anda hanya menggunakan satu keluarga instans untuk kemudahan manajemen.
- Anda mengabaikan rekomendasi dari AWS Cost Explorer atau Pengoptimal Komputasi untuk penentuan ukuran yang tepat.
- Anda tidak mengevaluasi ulang beban kerja untuk memeriksa kesesuaian tipe instans baru.
- Anda hanya mengesahkan sejumlah kecil konfigurasi instans untuk organisasi Anda.

Manfaat menerapkan praktik terbaik ini: Penyesuaian ukuran yang tepat untuk sumber daya komputasi akan 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 juga dapat menurunkan biaya.

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

Panduan implementasi

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

Langkah-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 berbasis atribut untuk Amazon EC2 Fleet
  - Membuat grup Auto Scaling dengan menggunakan pemilihan jenis instans berdasarkan atribut
  - Mengoptimalkan biaya komputasi Kubernetes Anda dengan konsolidasi Karpenter
- Lakukan analisis terhadap 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 tujuan-tujuan kinerja Anda.

- Pantau penggunaan sumber daya Anda dengan menggunakan alat-alat pemantauan AWS seperti Amazon CloudWatch.
- Pilih konfigurasi yang tepat untuk sumber daya komputasi.
  - Untuk beban kerja sementara, lakukan evaluasi terhadap metrik instans Amazon CloudWatch seperti `CPUUtilization` untuk mengidentifikasi apakah instans tersebut pemanfaatannya terlalu rendah atau terlalu tinggi.
  - Untuk beban kerja stabil, periksa alat-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 non-produksi sebelum diimplementasikan di lingkungan langsung.
- Lakukan evaluasi ulang secara terus-menerus terhadap penawaran komputasi baru dan bandingkan dengan kebutuhan-kebutuhan beban kerja Anda.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Jenis Instans Amazon EC
- 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:

- Landasan 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 Konsol Manajemen AWS
- AWS re:Invent 2023 – Apa yang baru dengan Amazon EC2
- AWS re:Invent 2023 – Penghematan cerdas: Strategi optimalisasi biaya Amazon EC
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Memahami lebih dalam Sistem Nitro

- AWS re:Invent 2019 – Landasan Amazon EC

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 terlalu tinggi atau terlalu rendah untuk beban kerja Anda.

Anti-pola umum:

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

Manfaat menerapkan 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 diterapkan: Tinggi

Panduan implementasi

AWS memberikan fleksibilitas untuk menaikkan atau menurunkan skala sumber daya Anda secara dinamis melalui berbagai mekanisme penskalaan untuk memenuhi perubahan-perubahan permintaan. Digabungkan dengan metrik-metrik yang terkait dengan komputasi, penskalaan dinamis akan 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 sasaran: Pantau metrik penskalaan Anda dan tingkatkan atau turunkan kapasitas secara otomatis sesuai kebutuhan.
- Penskalaan prediktif: Lakukan pengurangan skala (scale in) dalam mengantisipasi tren harian dan mingguan.
- Pendekatan berbasis jadwal: Tetapkan jadwal penskalaan Anda sendiri sesuai dengan perubahan-perubahan beban yang dapat diprediksi.
- Penskalaan layanan: Pilihlah layanan-layanan (seperti nirserver) yang secara otomatis menskalakan sesuai rancangan.

Anda harus memastikan bahwa deployment beban kerja tersebut dapat menangani peristiwa kenaikan (scale-up) dan penurunan skala (scale-down).

Langkah-langkah implementasi

- Instans, kontainer, dan fungsi komputasi menyediakan mekanisme bagi elastisitas, baik dengan dikombinasikan bersama penskalaan otomatis atau sebagai sebuah fitur layanan. Berikut beberapa contoh mekanisme penskalaan otomatis:

Mekanisme Penskalaan Otomatis	Harus digunakan di mana
Amazon EC2 Auto Scaling	Untuk memastikan Anda memiliki jumlah instans Amazon EC2 yang tepat yang tersedia untuk menangani beban aplikasi Anda.
Penskalaan Otomatis Aplikasi	Untuk secara otomatis menskalakan sumber daya untuk masing-masing layanan AWS di luar Amazon EC2 seperti fungsi AWS Lambda atau layanan Amazon Elastic Container Service (Amazon ECS).
Penskala Otomatis Klaster Kubernetes/ Karpenter	Untuk secara otomatis menskalakan klaster Kubernetes.

- Penskalaan sering kali dibahas terkait dengan layanan-layanan komputasi seperti Instans Amazon EC2 atau fungsi AWS Lambda. Pastikan juga untuk mempertimbangkan konfigurasi layanan non-komputasi seperti AWS Glue untuk mengimbangi permintaan.
- Pastikan bahwa metrik-metrik untuk penskalaan cocok dengan karakteristik beban kerja yang sedang di-deploy. Jika Anda men-deploy sebuah 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 dikustom untuk kebijakan penskalaan Anda, jika diperlukan. Untuk memilih metrik yang tepat, pertimbangkan panduan berikut untuk Amazon EC2:
  - Metrik tersebut harus merupakan metrik pemanfaatan yang valid dan mendeskripsikan tingkat kesibukan suatu instans.
  - Nilai metrik harus meningkatkan atau menurunkan jumlah instans secara proporsional dalam grup Auto Scaling.
- Pastikan Anda menggunakan penskalaan dinamis alih-alih penskalaan manual untuk grup Auto Scaling Anda. Kami juga menyarankan agar Anda menggunakan kebijakan penskalaan pelacakan target dalam penskalaan dinamis Anda.
- Pastikan deployment beban kerja dapat menangani dua jenis peristiwa penskalaan (naik dan turun). Sebagai contoh, Anda dapat menggunakan Riwayat aktivitas untuk melakukan verifikasi terhadap aktivitas penskalaan untuk sebuah grup Auto Scaling.
- Lakukan evaluasi terhadap beban kerja Anda untuk memeriksa pola-pola terprediksi dan secara proaktif skalakan saat Anda mengantisipasi perubahan permintaan yang terencana dan terprediksi. Dengan penskalaan prediktif, Anda dapat menghilangkan kebutuhan untuk menyediakan kapasitas secara berlebih. Untuk detail selengkapnya, silakan lihat Penskalaan Prediktif dengan Amazon EC2 Auto Scaling.

Sumber daya

Dokumen terkait:

- Komputasi Cloud dengan AWS
- Jenis 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
- Memahami Lebih Dalam Penskalaan Otomatis Klaster Amazon ECS
- Memperkenalkan Karpenter – Penskala Otomatis Klaster Kubernetes Sumber Terbuka dan Performa 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 optimalisasi biaya Amazon EC2
- AWS re:Invent 2021 – Mendukung Amazon EC2 generasi berikutnya: Memahami lebih dalam 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-fungsi tertentu secara lebih efisien daripada menggunakan alternatif yang berbasis CPU.

Anti-pola 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 yang lebih rendah.
- Anda menggunakan akselerator komputasi berbasis perangkat keras untuk tugas-tugas yang bisa lebih efisien jika menggunakan alternatif yang 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 diterapkan: Sedang

Panduan implementasi

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

Langkah-langkah implementasi

- Identifikasi instans komputasi terakselerasi yang dapat memenuhi kebutuhan Anda.
- Untuk beban kerja machine learning, manfaatkan perangkat keras yang dibuat khusus untuk beban kerja Anda, seperti AWS Trainium, AWS Inferentia, dan Amazon EC2 DL1. AWS Instans-instans Inferentia seperti instans Inf2 menawarkan kinerja/watt hingga 50% lebih baik daripada instans Amazon EC2 yang setara.
- Kumpulkan metrik-metrik penggunaan untuk instans komputasi terakselerasi Anda. Misalnya, Anda dapat menggunakan agen CloudWatch untuk mengumpulkan metrik seperti metrik `utilization_gpu` dan `utilization_memory` untuk GPU Anda seperti yang ditunjukkan dalam Mengumpulkan metrik GPU NVIDIA dengan Amazon CloudWatch.
- Optimalkan kode, operasi jaringan, dan pengaturan akselerator perangkat keras untuk memastikan perangkat keras yang mendasarinya dimanfaatkan sepenuhnya.
  - Mengoptimalkan pengaturan GPU
  - Pemantauan dan Pengoptimalan GPU dalam AMI Deep Learning
  - Mengoptimalkan I/O untuk penyetelan kinerja GPU pelatihan deep learning di Amazon SageMaker AI
- 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 Layanan Kontainer Elastis Amazon
- Instans GPU
- Instans dengan AWS Trainium
- Instans dengan AWS Inferentia
- Mari Merancang! Merancang arsitektur dengan chip dan akselerator kustom

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

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 Inferensia2
- AWS re:Invent 2022 - Mempercepat deep learning dan berinovasi lebih cepat dengan AWS Trainium
- AWS re:Invent 2022 - Deep learning di AWS dengan NVIDIA: Dari pelatihan hingga deployment

Contoh terkait:

- Amazon SageMaker AI dan NVIDIA GPU Cloud (NGC)
- Gunakan SageMaker AI dengan Trainium dan Inferentia untuk pelatihan deep learning yang dioptimalkan dan menyimpulkan beban kerja

- Mengoptimalkan model NLP dengan instans Amazon Elastic Compute Cloud Inf1 di Amazon SageMaker AI

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 data dan penyimpanan data Anda
- PERF03-BP02 Lakukan evaluasi terhadap opsi konfigurasi yang tersedia untuk penyimpanan data
- PERF03-BP03 Mengumpulkan dan merekam metrik-metrik kinerja penyimpanan data
- PERF03-BP04 Menerapkan strategi untuk meningkatkan kinerja kueri dalam penyimpanan data
- PERF03-BP05 Mengimplementasikan pola akses data yang memanfaatkan caching

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

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

Anti-pola umum:

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

Manfaat menerapkan 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 diterapkan: Tinggi

Panduan implementasi

Saat memilih dan menerapkan penyimpanan data, pastikan karakteristik penyimpanan, kueri, dan penskalaan mendukung persyaratan-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 bisa Anda gunakan 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 sehingga Anda bisa berkonsentrasi pada manajemen data yang tepat.

Langkah-langkah implementasi

- Lakukan inventarisasi 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 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. Beberapa contoh teknologi penyimpanan AWS serta karakteristik utamanya antara lain:

Tipe	Layanan AWS	Karakteristik utama
Penyimpanan objek	Amazon S3	Skalabilitas tak terbatas, ketersediaan tinggi, dan berbagai opsi aksesibilitas. Mentransfer dan mengakses objek masuk dan keluar dari Amazon S3 dapat dilakukan dengan menggunakan layanan-layanan, seperti Akselerasi Transfer atau Titik Akses, untuk mendukung lokasi, kebutuhan keamanan, dan pola akses Anda.
Penyimpanan pengarsipan	Amazon Glacier	Dirancang untuk pengarsipan data.
Penyimpanan streaming	Amazon Kinesis Amazon Managed Streaming for Apache Kafka (Amazon MSK)	Penyerapan dan penyimpan an data streaming yang efisien.
Sistem file bersama	Amazon Elastic File System (Amazon EFS)	Sistem file yang dapat dipasang dan dapat diakses oleh berbagai jenis solusi komputasi.

Tipe	Layanan AWS	Karakteristik utama
Sistem file bersama	Amazon FSx	Dibangun berdasarkan solusi komputasi AWS terbaru untuk mendukung empat sistem file yang umum digunakan: NetApp ONTAP, OpenZFS, Windows File Server, dan Lustre. Latensi, throughput, dan IOPS Amazon FSx berbeda-beda untuk setiap sistem file dan hal ini harus menjadi pertimbangan Anda saat memilih sistem file yang tepat untuk kebutuhan beban kerja.
Penyimpanan blok	Amazon Elastic Block Store (Amazon EBS)	Layanan penyimpanan blok yang dapat diskalakan dan mempunyai performa tinggi yang dirancang untuk Amazon Elastic Compute Cloud (Amazon EC2). Amazon EBS mencakup penyimpanan yang didukung SSD untuk beban kerja transaksional, intensif IOPS, dan penyimpanan yang didukung HDD untuk beban kerja yang intensif throughput.

Tipe	Layanan AWS	Karakteristik utama
Basis data relasional	Amazon Aurora, Amazon RDS, Amazon Redshift.	Didesain untuk mendukung transaksi ACID (atomisitas, konsistensi, isolasi, durabilit as), dan mempertahankan integritas referensial serta konsistensi data yang tinggi. Banyak aplikasi konvensio nal, perencanaan sumber daya perusahaan (ERP), manajemen hubungan pelanggan (CRM), dan ecommerce yang menggunak an basis data relasional untuk menyimpan data mereka.
Basis data nilai-kunci	Amazon DynamoDB	Dioptimalkan untuk pola akses umum, biasanya digunakan untuk menyimpan dan mengambil data dalam volume besar. Aplikasi web dengan lalu lintas tinggi, sistem ecommerce, dan aplikasi gaming merupakan kasus penggunaan umum untuk basis data nilai-kunci.
Basis data dokumen	Amazon DocumentDB	Dirancang untuk menyimpan data semi-terstruktur sebagai dokumen mirip JSON. Basis data ini membantu para pengembang untuk dengan cepat membangun dan memperbarui aplikasi seperti manajemen konten, katalog, dan profil pengguna.

Tipe	Layanan AWS	Karakteristik utama
Basis data dalam memori	Amazon ElastiCache, Amazon MemoryDB for Redis	Digunakan untuk aplikasi-aplikasi yang memerlukan akses waktu nyata ke data, latensi rendah, dan throughput paling tinggi. Anda dapat menggunakan basis data dalam memori untuk melakukan caching aplikasi, manajemen sesi, papan peringkat game, penyimpanan fitur ML latensi rendah, sistem olah pesan layanan mikro, dan mekanisme streaming throughput tinggi
Basis data grafik	Amazon Neptune	Digunakan untuk aplikasi-aplikasi yang harus menavigasi dan melakukan kueri jutaan hubungan antara set data grafik yang sangat terhubung dengan latensi milidetik dalam skala besar. Banyak perusahaan menggunakan basis data grafik untuk mesin rekomendasi, jaringan sosial, dan deteksi penipuan.

Tipe	Layanan AWS	Karakteristik utama
Basis Data Deret Waktu	Amazon Timestream	Digunakan untuk mengumpulkan, mempersatukan, dan mengambil wawasan secara efisien dari data yang berubah seiring waktu. Aplikasi internet untuk segala (IoT), DevOps, dan telemetri industri dapat menggunakan basis data deret waktu.
Kolom lebar	Amazon Keyspaces (untuk Apache Cassandra)	Menggunakan tabel, baris, dan kolom, tetapi tidak seperti basis data relasional, nama dan format kolomnya berbeda-beda dari baris ke baris di tabel yang sama. Biasanya Anda akan melihat penyimpanan kolom lebar di aplikasi industri skala tinggi untuk melakukan pemeliharaan perlengkapan, pengelolaan armada, dan pengoptimalan rute.

Tipe	Layanan AWS	Karakteristik utama
Buku besar	Amazon Quantum Ledger Database (Amazon QLDB)	Memberikan otoritas tersentralisasi yang tepercaya untuk mempertahankan data transaksi yang dapat diskalakan, tetap, dan dapat diverifikasi secara kriptografis untuk setiap aplikasi. Basis data buku besar digunakan untuk sistem catatan, rantai pasokan, registrasi, dan bahkan transaksi perbankan.

- Jika Anda membangun sebuah 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:

Pertanyaan	Hal-hal yang perlu dipertimbangkan
Bagaimana data terstruktur?	• Jika data tidak terstruktur, pertimbangkan penyimpanan objek seperti Amazon S3 atau basis data NoSQL seperti Amazon DocumentDB • Untuk data nilai kunci, pertimbangkan untuk menggunakan DynamoDB, Amazon ElastiCache (Redis OSS) atau Amazon MemoryDB
Apa tingkat integritas referensial yang dibutuhkan?	• Untuk kendala kunci asing, basis data relasional seperti Amazon RDS dan Aurora dapat memberikan tingkat integritas seperti ini.

Pertanyaan	Hal-hal yang perlu dipertimbangkan
	• Biasanya, dalam model data NoSQL, Anda akan melakukan de-normalisasi data menjadi satu dokumen atau kumpulan dokumen untuk diambil dalam satu permintaan, bukan digabungkan dalam berbagai dokumen atau tabel.
Apakah diperlukan kepatuhan terhadap ACID (atomisitas, konsistensi, isolasi, durabilitas)?	• Jika diperlukan sifat ACID yang terkait dengan basis data relasional, pertimbangkan untuk menggunakan basis data relasional seperti Amazon RDS dan Aurora. • Jika diperlukan konsistensi yang tinggi untuk basis data NoSQL, Anda dapat menggunakan bacaan sangat konsisten dengan DynamoDB.
Bagaimana persyaratan penyimpanan akan berubah seiring waktu? Bagaimana dampaknya pada skalabilitas?	• Basis data nirserver seperti DynamoDB dan Amazon Quantum Ledger Database (Amazon QLDB) akan diskalakan secara dinamis. • Basis data relasional memiliki batas atas terkait penyimpanan yang tersedia, dan sering kali harus dipartisi secara horizontal dengan menggunakan mekanisme seperti serpihan (sharding) setelah penyimpanan tersebut mencapai batas ini.
Berapakah proporsi kueri baca dibandingkan dengan kueri tulis? Apakah caching akan meningkatkan performa?	• Beban kerja read-heavy bisa mendapatkan keuntungan dari lapisan caching, seperti ElastiCache atau DAX jika basis datanya menggunakan DynamoDB. • Bacaan juga dapat dilimpahkan ke replika baca dengan basis data relasional seperti Amazon RDS.

Pertanyaan	Hal-hal yang perlu dipertimbangkan
Apakah penyimpanan dan modifikasi (OLTP - Pemrosesan Transaksi Online) atau pengambilan dan pelaporan (OLAP - Pemrosesan Analitik Online) memiliki prioritas yang lebih tinggi?	• 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 analitik, pertimbangkan untuk menggunakan basis data kolom seperti Amazon Redshift atau mengekspor data ke Amazon S3 dan melakukan analisis menggunakan Athena atau Amazon Quick.
Tingkat durabilitas apa yang diperlukan data?	• 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, yang mendukung ekspor data ke Amazon S3 untuk pengarsipan dan retensi jangka panjang.
Apakah ada keinginan untuk beralih dari mesin basis data komersial atau biaya lisensi?	• 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 mesin sumber terbuka

Pertanyaan	Hal-hal yang perlu dipertimbangkan
Apa harapan operasional untuk basis data tersebut? Apakah beralih ke layanan terkelola menjadi perhatian utama?	• Memanfaatkan Amazon RDS dan bukannya Amazon EC2, dan DynamoDB atau Amazon DocumentDB dan bukannya basis data NoSQL hosting mandiri, dapat mengurangi biaya tambahan operasional.
Bagaimana basis data diakses saat ini? Apakah hanya akses aplikasi, atau adakah pengguna inteligensi bisnis (BI) dan aplikasi siap pakai lain yang terhubung?	• Jika Anda memiliki dependensi pada peralatan eksternal maka Anda mungkin harus menjaga kompatibilitas dengan basis data yang mendukungnya. Amazon RDS sepenuhnya kompatibel dengan versi mesin yang berbeda yang didukungnya termasuk Microsoft SQL Server, Oracle, MySQL, dan PostgreSQL.

- Lakukan uji coba dan uji tolok ukur di lingkungan non-produksi 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
- Performa Amazon FSx for Windows File Server
- Amazon Glacier: Dokumentasi Amazon Glacier
- Amazon S3: Pertimbangan Tingkat Permintaan dan Performa
- Penyimpanan Cloud dengan AWS
- Karakteristik I/O Amazon EBS
- Basis Data Cloud dengan AWS
- AWS Caching Basis Data

- DynamoDB Accelerator
- Video praktik terbaik Amazon Aurora
- Kinerja Amazon Redshift
- 10 kiat performa teratas 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: Meningkatkan 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 tingkat lanjut dengan Amazon DynamoDB
- AWS re:Invent 2022: Melakukan modernisasi aplikasi dengan basis data yang dibuat khusus
- Memahami lebih dalam Amazon DynamoDB: Pola desain tingkat lanjut

Contoh terkait:

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

Anti-pola umum:

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

Manfaat menerapkan praktik terbaik ini: Dengan menjelajahi dan melakukan eksperimen dengan konfigurasi penyimpanan data, Anda mungkin dapat mengurangi biaya infrastruktur, meningkatkan performa, serta mengurangi upaya pengelolaan beban kerja.

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

Panduan implementasi

Beban kerja dapat memiliki satu atau beberapa penyimpanan data yang digunakan berdasarkan persyaratan-persyaratan penyimpanan data dan akses data. Untuk mengoptimalkan biaya dan efisiensi kinerja, Anda harus melakukan evaluasi terhadap pola akses data untuk menentukan konfigurasi penyimpanan data yang sesuai. Saat mencoba berbagai opsi penyimpanan data tersebut, Anda harus mempertimbangkan beberapa aspek seperti opsi penyimpanan, memori, komputasi,

replika baca, persyaratan konsistensi, pooling koneksi, dan opsi cache. Cobalah berbagai opsi konfigurasi ini untuk meningkatkan metrik efisiensi kinerja.

Langkah-langkah implementasi

- Pahami konfigurasi (seperti tipe instans, ukuran penyimpanan, atau versi mesin basis data) penyimpanan data Anda saat ini.
- Tinjau dokumentasi dan praktik terbaik AWS untuk mempelajari opsi konfigurasi yang disarankan yang dapat membantu meningkatkan kinerja penyimpanan data Anda. Berikut ini adalah opsi-opsi penyimpanan data utama yang perlu dipertimbangkan:

Opsi Konfigurasi	Contoh-contoh
Melimpahkan beban baca (seperti replika baca dan caching)	• Untuk tabel DynamoDB, Anda dapat meringankan beban pembacaan menggunakan DAX untuk caching. • Anda dapat membuat sebuah klaster Amazon ElastiCache (Redis OSS) dan mengonfigurasikan aplikasi Anda untuk membaca dari cache terlebih dahulu, yang kembali ke basis data jika item yang diminta tidak tersedia. • Basis data relasional seperti Amazon RDS dan Aurora, serta basis data NoSQL yang disediakan seperti Neptune dan Amazon DocumentDB, semuanya mendukung penambahan replika baca untuk mengurang i porsi baca beban kerja. • Basis data nirserver seperti DynamoDB akan menskalakan secara otomatis. Pastikan bahwa unit kapasitas baca (RSU) yang disediakan cukup untuk mengatasi beban kerja.

Opsi Konfigurasi	Contoh-contoh
Menskalakan penulisan (seperti serpihan (sharding) kunci partisi atau memperkenalkan antrean)	• Untuk basis data relasional, Anda dapat memperbesar ukuran instans untuk mengakomodasi tambahan beban kerja atau menambah IOPS yang disediakan untuk memfasilitasi kenaikan throughput pada penyimpanan yang mendasari. • Anda juga dapat membuat antrean di depan basis data Anda, bukan menulis secara langsung ke basis data. Dengan pola-pola ini, Anda dapat memisahkan penyerapa n 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 Anda meningkatkan throughput dalam basis data relasional dengan volume penulisan yang 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 dimitigasi dengan memilih distribusi kunci partisi yang lebih merata atau dengan serpihan (sharding) penulisan kunci partisi (write-sharding).

Opsi Konfigurasi	Contoh-contoh
Kebijakan untuk mengelola siklus hidup set data Anda	• Anda dapat menggunakan Siklus Hidup Amazon S3 untuk mengelola objek-objek Anda di sepanjang siklus hidupnya. Jika pola akses Anda tidak diketahui, berubah, atau tidak dapat diprediksi, maka Anda dapat menggunakan Amazon S3 Intelligent-Tiering, yang akan memantau pola akses dan secara otomatis memindahkan objek-objek yang belum diakses ke tingkat akses yang berbiaya lebih rendah. Anda dapat memanfaatkan metrik Lensa Penyimpanan Amazon S3 untuk melakukan identifikasi terhadap peluang dan celah pengoptimalan yang ada dalam manajemen siklus hidup. • Manajemen siklus hidup Amazon EFS secara otomatis akan mengelola penyimpanan file untuk sistem file Anda.
Manajemen koneksi dan 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 non-produksi untuk mengidentifikasi opsi konfigurasi mana yang dapat memenuhi persyaratan-persyaratan beban kerja Anda.
- Setelah melakukan uji coba, rencanakan migrasi dan validasi metrik-metrik kinerja Anda.
- Gunakan alat-alat pemantauan AWS (seperti Amazon CloudWatch) dan pengoptimalan (seperti Lensa Penyimpanan Amazon S3) untuk terus mengoptimalkan penyimpanan data Anda dengan menggunakan pola penggunaan dunia nyata.

Sumber daya

Dokumen terkait:

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

Video terkait:

- AWS re:Invent 2023: Meningkatkan efisiensi Amazon Elastic Block Store dan menjadi lebih hemat biaya
- AWS re:Invent 2023: Optimalisasi 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: Yang baru dengan penyimpanan file AWS
- AWS re:Invent 2023: Memahami lebih dalam tentang Amazon DynamoDB

Contoh terkait:

- AWS Lokakarya Basis Data yang Dibuat Khusus
- Basis Data untuk Pengembang
- Hari Imersi Arsitektur Data Modern AWS
- Amazon EBS Autoscale
- Contoh-contoh Amazon S3
- Contoh Amazon DynamoDB
- Sampel migrasi Basis Data AWS
- Lokakarya Modernisasi Basis Data
- Menggunakan parameter pada Amazon RDS for Postgress DB Anda

PERF03-BP03 Mengumpulkan dan merekam metrik-metrik kinerja penyimpanan data

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

Anti-pola umum:

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

Manfaat menerapkan 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 akan meningkatkan kinerja dan keandalan penyimpanan data.

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

Panduan implementasi

Untuk memantau kinerja penyimpanan data, Anda harus merekam beberapa metrik kinerja secara selama periode waktu tertentu. Dengan begitu Anda dapat mendeteksi anomali yang terjadi dan mengukur kinerja berdasarkan metrik bisnis untuk memastikan bahwa kebutuhan beban kerja Anda terpenuhi.

Metrik harus menyertakan sistem yang medasari yang mendukung metrik penyimpanan data dan metrik basis data. Metrik sistem yang mendasari dapat meliputi metrik pemanfaatan CPU, memori, penyimpanan diska yang tersedia, I/O diska, 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-data ini sangat penting untuk memahami kinerja beban kerja dan bagaimana solusi manajemen data digunakan. Gunakan metrik-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 yang terkait dengan kinerja basis data.

Langkah-langkah implementasi

- Identifikasi metrik-metrik kinerja utama yang perlu dilacak oleh penyimpanan data Anda.
  - Metrik dan dimensi Amazon S3
  - Memantau metrik dalam instans Amazon RDS
  - Memantau muatan DB dengan Wawasan Performa di Amazon RDS
  - Ringkasan Pemantauan yang Ditingkatkan
  - Dimensi dan Metrik DynamoDB
  - Memantau DynamoDB Accelerator
  - Memantau Amazon MemoryDB dengan Amazon CloudWatch
  - Metrik Apa yang Harus Saya Pantau?
  - Memantau kinerja klaster Amazon Redshift
  - Metrik dan Dimensi Timestream
  - Metrik Amazon CloudWatch untuk Amazon Aurora
  - Pencatatan log dan pemantauan di Amazon Keyspaces (untuk Apache Cassandra)

    - Memantau Sumber Daya Amazon Neptune
- Gunakan solusi pencatatan log dan pemantauan yang disetujui untuk mengumpulkan metrik-metrik ini. Amazon CloudWatch dapat mengumpulkan metrik di seluruh sumber daya dalam arsitektur Anda. Anda juga dapat mengumpulkan dan mempublikasikan metrik-metrik kustom untuk menampilkan metrik-metrik bisnis atau turunan. Gunakan CloudWatch atau solusi pihak ketiga untuk menetapkan alarm yang memberikan indikasi saat ada ambang batas yang terlampaui.
- Periksa apakah pemantauan penyimpanan data dapat terbantu dengan solusi machine learning yang mendeteksi adanya anomali kinerja.
    - Amazon DevOps Guru untuk Amazon RDS menyediakan visibilitas terhadap masalah-masalah kinerja dan memberikan saran tindakan perbaikan.
- Atur konfigurasi retensi data dalam solusi pemantauan dan pencatatan log Anda agar sesuai dengan tujuan-tujuan keamanan dan operasional Anda.
    - Retensi data default untuk metrik CloudWatch
    - Retensi data default untuk Log CloudWatch

Sumber daya

Dokumen terkait:

- Caching Basis Data AWS
- 10 kiat kinerja teratas Amazon Athena
- Video praktik terbaik Amazon Aurora
- DynamoDB Accelerator
- Praktik terbaik Amazon DynamoDB
- Praktik terbaik Amazon Redshift Spectrum
- Kinerja Amazon Redshift
- Basis Data Cloud dengan AWS
- Wawasan Performa 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 untuk Amazon RDS
- AWS re:Invent 2023 - Apa yang baru dengan penyimpanan file AWS
- AWS re:Invent 2023 - Memahami lebih dalam tentang 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 lebih dalam tentang 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 Dibuat Khusus AWS

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

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

Anti-pola umum:

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

Manfaat menerapkan 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 diterapkan: Sedang

Panduan implementasi

Optimalisasi data dan penyetelan kueri merupakan aspek-aspek penting efisiensi kinerja dalam sebuah penyimpanan data, karena berdampak pada kinerja dan responsivitas seluruh beban kerja cloud. Kueri yang tidak dioptimalkan dapat menyebabkan terjadinya penggunaan dan kemacetan sumber daya yang lebih besar, yang dapat mengurangi efisiensi penyimpanan data secara keseluruhan.

Pengoptimalan data mencakup beberapa teknik untuk memastikan penyimpanan data dan akses data yang efisien. Hal ini juga dapat membantu Anda meningkatkan performa kueri dalam penyimpanan data. Strategi utamanya mencakup partisi data, kompresi data, dan denormalisasi data, yang akan membantu optimalisasi penyimpanan dan akses data.

Langkah-langkah implementasi

- Pahami dan analisis kueri data penting yang dilakukan di dalam penyimpanan data Anda.
- Identifikasi kueri-kueri yang berjalan lambat di dalam 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-strategi untuk meningkatkan kinerja kueri. Beberapa strategi utamanya meliputi:
  - Menggunakan format file kolom (seperti Parquet atau ORC).
  - Mengompresi data di dalam penyimpanan data untuk mengurangi ruang penyimpanan dan operasi I/O.
  - Melakukan partisi data untuk membagi data menjadi bagian-bagian yang lebih kecil dan mengurangi waktu pemindaian data.
    - Melakukan partisi data di Athena
    - Partisi dan distribusi data
  - Pengindeksan data pada kolom umum dalam kueri.
  - Gunakan tampilan terwujud untuk kueri yang sering dibuat.
    - 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 berada di sisi kiri gabungan dan tabel yang lebih kecil berada di sisi kanan gabungan.

  - Solusi cache terdistribusi untuk meningkatkan latensi dan mengurangi jumlah operasi I/O basis data.
  - Pemeliharaan rutin seperti pemvakuman, pengindeksan ulang, dan menjalankan statistik.
- Lakukan eksperimen dan uji strategi di sebuah lingkungan non-produksi.

Sumber daya

Dokumen terkait:

- Video praktik terbaik Amazon Aurora
- Kinerja Amazon Redshift
- 10 kiat kinerja teratas Amazon Athena
- Caching Basis Data AWS
- Praktik Terbaik untuk Mengimplementasikan Amazon ElastiCache
- Melakukan partisi data di Athena

Video terkait:

- AWS re:Invent 2023 - Praktik terbaik optimalisasi 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:

- AWS Lokakarya Basis Data yang Dibuat Khusus

PERF03-BP05 Mengimplementasikan pola akses data yang memanfaatkan caching

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

Anti-pola 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 dari implementasi caching Anda.

Manfaat menerapkan 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 diterapkan: Sedang

Panduan implementasi

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

Caching data dapat menjadi salah satu strategi yang 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 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 cache jarak jauh.

Caching sisi klien

Dengan melakukan 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 yang mungkin digunakan (klien itu sendiri), sehingga menghasilkan latensi yang serendah mungkin. Klien juga dapat terus melayani beberapa kueri ketika penyimpanan data backend tidak tersedia, sehingga akan meningkatkan ketersediaan sistem secara keseluruhan.

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

Caching jarak jauh

Untuk mengatasi masalah duplikat data yang terjadi antar klien, suatu layanan eksternal cepat, atau cache jarak jauh, dapat digunakan untuk menyimpan data yang di-kueri. Alih-alih memeriksa penyimpanan data lokal, masing-masing klien akan memeriksa cache jarak jauh sebelum mengkueri penyimpanan data backend. Strategi ini memungkinkan respons yang lebih konsisten di antara klien, efisiensi yang lebih baik pada data yang disimpan, dan volume data cache yang lebih tinggi karena ruang penyimpanannya diskalakan secara independen tanpa terikat klien.

Kelemahan cache jarak jauh adalah sistem tersebut mungkin mengalami latensi yang lebih tinggi secara keseluruhan karena diperlukan lompatan jaringan tambahan untuk memeriksa cache jarak jauh. Caching sisi klien dapat digunakan bersama caching jarak jauh untuk melakukan caching multi-tingkat sehingga dapat meningkatkan latensi.

Langkah-langkah implementasi

- Identifikasi basis data, API, dan layanan-layanan jaringan yang dapat memanfaatkan caching. Layanan-layanan yang memiliki beban kerja baca yang berat, memiliki rasio baca-tulis yang tinggi, atau beban kerja yang mahal untuk diskalakan dapat memanfaatkan caching.
  - Caching Basis Data
  - Mengaktifkan caching API untuk meningkatkan responsivitas
- Identifikasi jenis strategi caching yang tepat yang paling sesuai dengan pola akses Anda.
  - Strategi pembuatan cache
  - Solusi Penerapan Cache AWS
- Ikuti Praktik Terbaik Caching untuk penyimpanan data Anda.
- Konfigurasikan sebuah strategi pembatalan cache, seperti time-to-live (TTL), untuk semua data yang menyeimbangkan kesegaran data dan mengurangi tekanan pada penyimpanan data backend.
- Aktifkan fitur seperti percobaan ulang koneksi otomatis, penundaan 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 (Redis OSS)
- 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 melakukan caching.
  - Metrik Apa yang Harus Saya Pantau?
  - Praktik Terbaik untuk Memantau Beban Kerja Redis di Amazon ElastiCache
  - Praktik terbaik pemantauan dengan Amazon ElastiCache (Redis OSS) dengan menggunakan Amazon CloudWatch
- Implementasikan replikasi data untuk melimpahkan beban baca ke beberapa instans dan meningkatkan performa dan ketersediaan pembacaan data.

Sumber daya

Dokumen terkait:

- Menggunakan Lensa Well-Architected Amazon ElastiCache
- Praktik terbaik pemantauan dengan Amazon ElastiCache (Redis OSS) dengan menggunakan Amazon CloudWatch
- Metrik Apa yang Harus Saya Pantau?
- Laporan Resmi Performa dalam 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 lebih dalam Amazon ElasticAche (Redis OSS)

Contoh terkait:

- Meningkatkan kinerja basis data MySQL dengan Amazon ElastiCache (Redis OSS)

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, Jejaring yang Ditingkatkan, instans yang dioptimalkan jaringan Amazon EC2, akselerasi transfer Amazon S3, dan Amazon CloudFront yang dinamis) untuk mengoptimalkan lalu lintas jaringan. AWS juga menawarkan fitur-fitur jaringan (misalnya perutean latensi Amazon Route 53, titik akhir VPC Amazon, AWS Direct Connect, dan AWS Global Accelerator) untuk mengurangi jarak atau jitter jaringan.

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 Pilih lokasi beban kerja Anda berdasarkan persyaratan jaringan
- PERF04-BP07 Mengoptimalkan konfigurasi jaringan berdasarkan metrik

PERF04-BP01 Memahami bagaimana jaringan memengaruhi performa

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

Anti-pola 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-persyaratan penggunaan aktual.
- Anda tidak mempertimbangkan karakteristik beban kerja dan biaya overhead enkripsi ketika menentukan solusi-solusi jaringan Anda.
- Anda menggunakan konsep dan strategi on-premise untuk solusi-solusi jaringan di cloud.

Manfaat menerapkan 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 diterapkan: 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-aturan perutean.

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

Langkah-langkah implementasi:

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

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

- Buat tolok ukur dan uji kinerja jaringan:
  - Lakukan uji tolok ukur terhadap throughput jaringan, karena beberapa faktor yang dapat memengaruhi kinerja jaringan Amazon EC2 saat instans berada di VPC yang sama. Ukur bandwidth jaringan yang ada di antara instans Linux Amazon EC2 di VPC yang sama.
  - Lakukan uji beban untuk bereksperimen dengan solusi dan opsi jaringan.

Sumber daya

Dokumen terkait:

- Penyeimbang Beban Aplikasi
- Peningkatan Jaringan EC2 di Linux
- Peningkatan Jaringan EC2 di Windows
- Grup Penempatan EC
- Memungkinkan Jaringan yang Ditingkatkan dengan Adaptor Jaringan Elastis (ENA) di Instans Linux
- Penyeimbang Beban Jaringan
- Produk Jaringan dengan AWS
- Transit Gateway
- Transisi ke latensi berbasis perutean 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 hibrida
- 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 Temuan dan Mengatasi Kesalahan Konfigurasi Jaringan pada AWS

PERF04-BP02 Mengevaluasi fitur jaringan yang tersedia

Evaluasi fitur jaringan yang ada di cloud yang dapat meningkatkan performa. 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).

Anti-pola 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 untuk 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 menerapkan 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 untuk memberikan pengalaman jaringan yang optimal bagi pelanggan Anda.

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

Panduan implementasi

AWS menawarkan layanan-layanan seperti AWS Global Accelerator dan Amazon CloudFront yang dapat membantu Anda dalam 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-langkah implementasi

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

Peluang peningkatan	Solusi
Jalur atau rute jaringan	Gunakan Penganalisis Akses Jaringan untuk mengidentifikasi jalur atau rute.
Protokol jaringan	Lihat PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa

Peluang peningkatan	Solusi
Topologi jaringan	Lakukan evaluasi terhadap kompromi operasional dan performa antara VPC Peering dan AWS Transit Gateway saat menghubungkan beberapa akun. AWS Transit Gateway menyederhanakan cara Anda melakukan penyambungan silang (inter-connection) semua VPC Anda, yang bisa saja berada di ribuan Akun AWS dan ke dalam jaringan on-premise. Bagikan AWS Transit Gateway Anda di antara beberapa akun dengan menggunakan AWS Resource Access Manager. Lihat PERF04-BP03 Memilih konektivitas khusus atau VPN yang tepat untuk beban kerja Anda

Peluang peningkatan	Solusi
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 kedekatan geografis, dan opsi perutean berbasis IP untuk membantu Anda meningkat kan performa beban kerja untuk audiens global. Identifikasi opsi-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 sebuah fitur yang memungkinkan pengguna eksternal untuk mendapatkan manfaat pengoptimalan jaringan CloudFront untuk mengunggah data ke Amazon S3. Hal ini akan meningkatkan kemampuan transfer data dalam jumlah besar dari lokasi jarak jauh yang tidak memiliki koneksi khusus ke AWS Cloud. Amazon S3 Multi-Region Access Points 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 dengan latensi terendah.

Peluang peningkatan	Solusi
Fitur sumber daya komputasi	Antarmuka Jaringan Elastis (ENI) yang digunakan oleh instans Amazon EC2, kontainer, dan fungsi Lambda dibatasi berdasarkan per alur. Tinjau grup penempatan Anda untuk mengoptimalkan throughput jaringan EC2 Anda. Untuk menghindari hambatan berdasarkan alur, rancang aplikasi Anda sedemikian rupa agar bisa menggunakan beberapa alur. Untuk memantau dan mendapatkan visibilitas tentang metrik-metrik jaringan terkait komputasi Anda, gunakan CloudWatch Metrics dan ethtool. Perintah `ethtool` disertakan dalam driver ENA dan mengekspos metrik-metrik terkait jaringan tambahan yang dapat dipublikasikan sebagai metrik kustom ke CloudWatch. Adaptor Jaringan Elastis (ENA) Amazon menyediakan optimalisasi lebih lanjut dengan menghadirkan throughput yang lebih baik untuk instans Anda di dalam sebuah grup penempatan klaster. Elastic Fabric Adapter (EFA) adalah sebuah antarmuka jaringan untuk instans Amazon EC2 yang memungkinkan Anda menjalankan beban kerja yang membutuhkan komunikasi antar-simpul 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:

- Penyeimbang Beban Aplikasi
- Peningkatan Jaringan EC2 di Linux
- Peningkatan Jaringan EC2 di Windows
- Grup Penempatan EC2
- Memungkinkan Jaringan yang Ditingkatkan dengan Adaptor Jaringan Elastis (ENA) di Instans Linux
- Penyeimbang Beban Jaringan
- Produk Jaringan dengan AWS
- Transisi ke Latensi Berbasis Perutean di Amazon Route 53
- Titik Akhir VPC
- Log Alur VPC

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 lebih dalam infrastruktur jaringan AWS
- AWS re:Invent 2019 – Konektivitas ke AWS dan arsitektur jaringan AWS hibrida
- AWS re:Invent 2018 – Mengoptimalkan Performa 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 temuan dan mengatasi kesalahan konfigurasi jaringan pada AWS

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

Ketika diperlukan konektivitas hibrida 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 hibrida Anda. Angka-angka ini akan mendorong persyaratan penyesuaian ukuran Anda.

Anti-pola umum:

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

Manfaat menerapkan praktik terbaik ini: Memilih dan mengonfigurasi solusi konektivitas yang tepat akan meningkatkan keandalan beban kerja dan memaksimalkan performa. Dengan mengidentifikasi persyaratan-persyaratan beban kerja, membuat perencanaan ke depan, dan mengevaluasi solusi hibrida, 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 diterapkan: Tinggi

Panduan implementasi

Kembangkan sebuah arsitektur jaringan hibrida berdasarkan kebutuhan bandwidth Anda. Direct Connect akan memungkinkan Anda untuk menghubungkan jaringan on-premise Anda secara privat dengan AWS. Layanan ini ideal ketika Anda memerlukan bandwidth yang tinggi dan latensi yang rendah sekaligus 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 kontinjensi sambil menunggu terbentuknya konektivitas jaringan fisik yang kuat saat menggunakan Direct Connect.

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

Langkah-langkah implementasi

- Perkirakan persyaratan-persyaratan bandwidth dan latensi aplikasi yang sudah Anda miliki.
  - Untuk beban kerja lama yang akan beralih ke AWS, manfaatkan data dari sistem pemantauan jaringan internal Anda.
  - Untuk beban kerja baru atau lama yang data pemantauannya tidak Anda miliki, hubungi pemilik produk untuk menentukan metrik-metrik performa yang memadai dan memberikan pengalaman pengguna yang baik.
- 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 Site-to-Site VPN (atau keduanya). Diagram berikut dapat membantu Anda memilih jenis sambungan yang tepat.
  - 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 memberikan Anda latensi yang terkelola dan terkontrol serta bandwidth yang tersedia agar beban kerja Anda dapat terhubung ke lingkungan-lingkungan lain secara efisien. Dengan menggunakan partner AWS Direct Connect, Anda dapat memiliki konektivitas yang 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).
  - VPN Site-to-Site AWS akan memberikan sebuah layanan VPN terkelola yang mendukung keamanan protokol internet (IPsec). Ketika sambungan VPN dibuat, setiap sambungan VPN mencakup dua terowongan untuk ketersediaan tinggi.
- Ikuti dokumentasi AWS untuk memilih opsi konektivitas yang sesuai:
  - Jika Anda memutuskan untuk menggunakan Direct Connect, pilih bandwidth yang sesuai untuk konektivitas Anda.
  - Jika Anda menggunakan AWS Site-to-Site VPN di beberapa lokasi untuk terhubung ke Wilayah AWS, gunakan koneksi VPN Site-to-Site yang dipercepat untuk kesempatan meningkatkan kinerja jaringan.
  - Jika desain jaringan Anda terdiri dari koneksi VPN IPsec melalui AWS Direct Connect, pertimbangkan untuk menggunakan Private IP VPN untuk meningkatkan keamanan dan mencapai segmentasi. AWS VPN IP Privat Site-to-Site digunakan di atas antarmuka virtual transit (VIF).

  - AWS Direct Connect SiteLink memungkinkan Anda untuk membuat koneksi latensi rendah dan redundan antara semua pusat data Anda di seluruh dunia dengan mengirimkan data melalui jalur tercepat antara lokasi AWS Direct Connect, yang mem-bypass Wilayah AWS.
- Lakukan validasi penyiapan konektivitas Anda sebelum deployment ke lingkungan produksi. Lakukan pengujian keamanan dan performa untuk memastikan persyaratan-persyaratan bandwidth, keandalan, latensi, dan kepatuhan Anda terpenuhi.
- Pantau performa dan penggunaan konektivitas Anda secara rutin 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 Dapat Diskalakan dan Aman
- Client VPN

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 hibrida AWS
- AWS re:Invent 2020 – AWS Transit Gateway Connect

Contoh terkait:

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

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, dan untuk mengelola serta merutekan lalu lintas secara efektif.

Anti-pola umum:

- Anda tidak mempertimbangkan persyaratan-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 menerapkan 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 diterapkan: 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. Mulailah 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 yang bisa digunakan dengan penyeimbangan beban. Penyeimbang Beban Aplikasi sangat ideal untuk menyeimbangkan beban

lalu lintas HTTP dan HTTPS dan menyediakan perutean permintaan lanjutan yang ditPenyeimbang Beban Aplikasiargetkan pada pengiriman arsitektur aplikasi modern, termasuk layanan mikro dan kontainer.

Penyeimbang Beban Jaringan sangat ideal untuk menyeimbangkan beban lalu lintas TCP yang memerlukan kinerja ekstrem. Penyeimbangan 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.

Penyeimbangan Beban Elastis 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 Penyeimbang Beban Aplikasi (ALB) dan Penyeimbang Beban Jaringan (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 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.

Penyeimbang Beban Aplikasi 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 arsitekturnya. Sebagai contoh, jika Anda memiliki aplikasi yang sensitif latensi, Anda dapat memutuskan untuk menggunakan Penyeimbang Beban Jaringan, yang menawarkan latensi yang sangat rendah. Alternatifnya, Anda dapat memutuskan untuk membawa beban kerja lebih dekat ke pelanggan dengan memanfaatkan Penyeimbang Beban Aplikasi di Zona Lokal AWS atau bahkan di 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 (penskalaan otomatis) yang terintegrasi dengan penyeimbang beban Anda. Salah satu aspek penting dari sebuah 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 (penskalaan otomatis), 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 terkontainerisasi.

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

Langkah-langkah implementasi

- Tentukan persyaratan-persyaratan penyeimbangan beban Anda, termasuk volume lalu lintas, ketersediaan, dan skalabilitas aplikasi.
- Pilih jenis penyeimbang beban yang tepat untuk aplikasi Anda.
  - Gunakan Penyeimbang Beban Aplikasi untuk beban kerja HTTP/HTTPS.
  - Gunakan Penyeimbang Beban Jaringan untuk beban kerja non-HTTP yang dijalankan di TCP atau UDP.
  - Gunakan kombinasi keduanya (ALB sebagai target NLB) jika Anda ingin memanfaatkan fitur dari kedua produk ini. 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 sebuah AWS PrivateLink.
  - Untuk perbandingan lengkap penyeimbang beban, silakan lihat Perbandingan produk ELB.
- Gunakan pemindahan beban SSL/TLS, jika memungkinkan.
  - Konfigurasikan pendengar HTTPS/TLS dengan Penyeimbang Beban Aplikasi dan Penyeimbang Beban Jaringan yang terintegrasi dengan AWS Certificate Manager.
  - Perhatikan, beberapa beban kerja mungkin memerlukan enkripsi menyeluruh (end-to-end) karena alasan kepatuhan. Jika demikian, enkripsi wajib diaktifkan di target.
  - Untuk praktik terbaik keamanan, silakan lihat SEC09-BP02 Menerapkan enkripsi saat transit.
- 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. Misalnya, ALB menyediakan dua opsi untuk algoritma perutean:
  - Permintaan paling tidak menonjol: Gunakan untuk mendapatkan distribusi beban yang lebih baik ke target backend Anda untuk kasus ketika permintaan-permintaan untuk aplikasi Anda mempunayi tingkat kompleksitas yang berbeda-beda atau target Anda kemampuan pemrosesannya berbeda-beda.
  - Round robin: Gunakan ketika permintaan dan target serupa, atau jika Anda harus mendistribusikan permintaan secara sama rata di antara banyak target.
- Pertimbangkan isolasi zona atau lintas zona.
  - Gunakan penonaktifan lintas zona (isolasi zona) untuk meningkatkan latensi dan domain kegagalan zona. Ini dimatikan secara default di NLB dan di ALB Anda dapat mematikannya per grup target.
  - Gunakan pengaktifan lintas zona untuk meningkatkan ketersediaan dan fleksibilitas. Secara 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 kembali sambungan backend sampai waktu tetap aktif (keep-alive) habis, sehingga meningkatkan waktu respons dan permintaan HTTP Anda serta mengurangi pemanfaatan sumber daya yang ada di target backend Anda. Untuk detail tentang cara melakukan hal ini untuk Apache dan Nginx, silakan lihat Apa pengaturan optimal untuk menggunakan Apache atau NGINX sebagai server backend untuk ELB?
- Aktifkan pemantauan untuk penyeimbang beban Anda.
  - Aktifkan log akses untuk Penyeimbang Beban Aplikasi dan Penyeimbang Beban Jaringan Anda.
  - Bidang utama yang harus 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 pada 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
- Kueri Log Penyeimbang Beban Aplikasi
- Memantau Penyeimbang Beban Aplikasi Anda
- Memantau Penyeimbang Beban Jaringan Anda
- Gunakan Penyeimbangan Beban Elastis Untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling Anda

Video terkait:

- AWS re:Invent 2023: Apa manfaat jaringan untuk aplikasi Anda?
- AWS re:Inforce 20 - Cara menggunakan Penyeimbangan Beban Elastis untuk meningkatkan postur keamanan Anda dalam skala besar
- AWS re:Invent 2018: Elastic Load Balancing: Memahami Lebih Dalam 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:

- Penyeimbang Beban Gateway
- Contoh CDK dan CloudFormation untuk Analisis Log dengan Amazon Athena

PERF04-BP05 Memilih protokol jaringan untuk meningkatkan performa

Buatlah 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 beberapa pendekatan untuk memperbaiki hal ini dengan penyesuaian TCP dan pengoptimalan protokol transfer, tetapi salah satu solusinya adalah menggunakan User Datagram Protocol (UDP)).

Anti-pola umum:

- Anda menggunakan TCP untuk semua beban kerja terlepas dari persyaratan kinerjanya.

Manfaat menerapkan praktik terbaik ini: Memverifikasi bahwa sebuah protokol yang tepat telah digunakan untuk komunikasi antara pengguna dan bahwa komponen beban kerja akan membantu Anda dalam meningkatkan pengalaman pengguna secara keseluruhan untuk aplikasi Anda. Misalnya, UDP tanpa koneksi memungkinkan kecepatan tinggi, tetapi tidak menawarkan transmisi ulang atau keandalan yang tinggi. TCP adalah sebuah protokol yang memiliki fitur 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-beda. Perlu diingat 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 adalah pemahaman Anda terhadap persyaratan latensi dan throughput, dan kemudian pemilihan protokol jaringan yang mengoptimalkan performa.

Kapan Anda harus mempertimbangkan penggunaan TCP

TCP memberikan pengiriman data yang andal, dan dapat digunakan untuk komunikasi yang terjadi antara komponen beban kerja di mana keandalan dan jaminan pengiriman data merupakan hal yang penting. Banyak sekali aplikasi berbasis web yang mengandalkan protokol berbasis TCP, seperti HTTP dan HTTPS, untuk membuka soket TCP untuk komunikasi yang terjadi antara komponen-komponen aplikasi. Email dan transfer data file adalah penerapan umum yang juga menggunakan protokol 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 terjadinya peningkatan latensi dan pengurangan throughput, tetapi protokol ini memiliki keunggulan dari segi keamanan. Overhead ini terutama berasal dari penambahan overhead untuk proses handshake, yang dapat memerlukan beberapa perjalanan pulang pergi agar selesai. Setelah handshake selesai, overhead enkripsi dan dekripsi data relatif kecil.

Kapan Anda harus mempertimbangkan untuk menggunakan UDP

UDP adalah sebuah protokol dengan orientasi tanpa koneksi, oleh karena itu, protokol ini 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 sudah disederhanakan. DTLS juga menambahkan sejumlah kecil overhead ke paket UDP, karena protokol ini mencakup bidang-bidang tambahan untuk menunjukkan parameter keamanan dan untuk mendeteksi manipulasi.

Kapan Anda harus mempertimbangkan penggunaan SRD

Scalable reliable datagram (SRD) adalah sebuah protokol transportasi jaringan yang dioptimalkan untuk beban kerja throughput tinggi karena kemampuannya untuk menjalankan lalu lintas penyeimbang beban melintasi beberapa jalur dan dapat pulih dengan cepat dari penurunan paket atau kegagalan tautan. Oleh karena itu, protokol 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-langkah implementasi

- Gunakan layanan AWS Global Accelerator dan AWS Transfer Family untuk memperbaiki throughput aplikasi transfer file online Anda. Layanan AWS Global Accelerator membantu Anda mendapatkan latensi lebih yang rendah antara perangkat klien dan beban kerja Anda yang ada 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-layanan penyimpanan AWS.
- 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 hal itu akan memengaruhi responsivitas aplikasi Anda. Metrik-metrik seperti waktu ke byte pertama (TTFB) dan waktu pulang pergi (RTT) dapat digunakan untuk mengukur latensi jaringan. Jika beban kerja Anda melayani konten dinamis untuk pengguna, pertimbangkan untuk menggunakan Amazon CloudFront, yang membuat sambungan persisten ke masing-masing asal konten dinamis untuk menyingkirkan waktu penyiapan sambungan yang akan memperlambat setiap permintaan klien.

- Menggunakan TLS dengan TCP atau UDP dapat mengakibatkan terjadinya peningkatan latensi dan pengurangan throughput untuk beban kerja Anda karena dampak yang ditimbulkan oleh enkripsi dan dekripsi. Untuk beban kerja semacam itu, pertimbangkan untuk menggunakan 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 Anda mengurangi pemanfaatan CPU di instans backend, yang dapat meningkatkan performa dan kapasitas.
- Gunakan Penyeimbang Beban Jaringan (NLB) untuk melakukan deployment layanan-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 dari satu target tunggal.
- Untuk beban kerja Komputasi Performa Tinggi (HPC) Anda, sebaiknya Anda menggunakan fungsionalitas Adaptor Jaringan Elastis (ENA) Express yang menggunakan protokol SRD untuk meningkatkan kinerja jaringan dengan menyediakan bandwidth aliran tunggal yang lebih tinggi (25 Gbps) dan latensi ekor yang lebih rendah (99,9 persentil) untuk lalu lintas jaringan antara instans EC2.
- Gunakan Penyeimbang Beban Aplikasi (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 transportasi dan gRPC memberikan manfaat terkait performa, seperti jejak jaringan yang 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
- Penyeimbang Beban Aplikasi

- Peningkatan Jaringan EC2 di Linux
- Peningkatan Jaringan EC2 di Windows
- Grup Penempatan EC2
- Memungkinkan Jaringan yang Ditingkatkan dengan Adaptor Jaringan Elastis (ENA) di Instans Linux
- Penyeimbang Beban Jaringan
- Produk Jaringan dengan AWS
- Transisi ke Latensi Berbasis Perutean 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 Pilih lokasi beban kerja Anda berdasarkan persyaratan jaringan

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

Anti-pola 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 menerapkan praktik terbaik ini: Pengalaman pengguna sangat mereka sangat terpengaruh oleh latensi yang ada antara pengguna dan aplikasi Anda. Dengan menggunakan jaringan global yang sesuai Wilayah AWS dan AWS pribadi, Anda dapat mengurangi latensi dan memberikan pengalaman yang lebih baik kepada pengguna jarak jauh.

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

Panduan implementasi

Sumber daya, seperti EC2 instans Amazon, ditempatkan ke Availability Zone di dalam Wilayah AWS, AWS Local Zones AWS Outposts, atau AWS Wavelengthzona. Pemilihan lokasi ini akan memengaruhi latensi jaringan dan throughput dari lokasi pengguna tertentu. Layanan Edge seperti Amazon CloudFront dan juga AWS Global Acceleratordapat digunakan untuk meningkatkan kinerja jaringan dengan menyimpan konten di lokasi tepi atau memberi pengguna jalur optimal ke beban kerja melalui jaringan AWS global.

Amazon EC2 menyediakan grup penempatan untuk jaringan. Grup penempatan adalah sebuah pengelompokan logis instans untuk mengurangi latensi. Menggunakan grup penempatan dengan tipe instans yang didukung dan Elastic Network Adapter (ENA) memungkinkan beban kerja untuk berpartisipasi dalam jaringan jitter 25 Gbps dengan latensi rendah dan dikurangi. 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 tepi menggunakan jaringan AWS global, seperti Amazon. CloudFront Lokasi tepi ini biasanya menyediakan layanan seperti jaringan pengiriman konten (CDN) dan sistem nama domain (DNS). Dengan memiliki layanan ini di tepi, beban kerja dapat merespons dengan latensi rendah untuk permintaan konten atau DNS resolusi. 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 para pengguna akhir ke Wilayah terdekat (latensi minimum).

Gunakan layanan-layanan edge untuk mengurangi latensi dan memungkinkan caching konten. Konfigurasikan kontrol cache dengan benar untuk keduanya DNS danHTTP/HTTPSuntuk mendapatkan manfaat maksimal dari pendekatan ini.

Langkah-langkah implementasi

- Rekam informasi tentang lalu lintas IP ke dan dari antarmuka jaringan.
  - Mencatat lalu lintas IP menggunakan VPC Flow Logs

  - Bagaimana alamat IP klien dipertahankan di AWS Global Accelerator
- Lakukan analisis terhadap pola akses jaringan di beban kerja Anda untuk mengidentifikasi cara pengguna menggunakan aplikasi Anda.
  - Gunakan alat pemantauan, seperti Amazon CloudWatch dan AWS CloudTrail, untuk mengumpulkan data tentang aktivitas jaringan.
  - Analisis data untuk mengidentifikasi pola akses jaringan.
- Pilih Wilayah untuk deployment beban kerja Anda berdasarkan elemen-elemen utama berikut:
  - Dimana lokasi data Anda: Untuk aplikasi-aplikasi dengan banyak data (seperti big data dan machine learning), kode aplikasi harus dijalankan sedekat mungkin dengan data.
  - Dimana lokasi pengguna Anda: Untuk aplikasi-aplikasi yang ditampilkan kepada pengguna, pilihlah sebuah Wilayah (Wilayah-wilayah) yang dekat dengan para pengguna beban kerja Anda.
  - Kendala lainnya: Pertimbangkan kendala-kendala seperti biaya dan kepatuhan sebagaimana yang dijelaskan dalam Hal-Hal yang Perlu Dipertimbangkan saat Memilih Wilayah untuk Beban Kerja Anda.
- Gunakan Zona Lokal AWS untuk menjalankan beban kerja seperti rendering video. Zona Lokal memungkinkan Anda untuk mendapatkan semua manfaat dari komputasi dan sumber daya penyimpanan yang lebih dekat dengan para 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 dengan beban kerja Anda yang lain yang ada di AWS.
- Aplikasi seperti streaming video langsung resolusi tinggi, audio kesetiaan tinggi, dan augmented reality atau virtual reality (AR/VR) memerlukan perangkat 5G. ultra-low-latency Untuk aplikasi semacam itu, pertimbangkan AWS Wavelength. AWS Wavelength AWS menyematkan layanan komputasi dan penyimpanan dalam jaringan 5G, menyediakan infrastruktur komputasi tepi seluler untuk mengembangkan, menyebarkan, dan menskalakan aplikasi. ultra-low-latency
- Gunakan caching lokal atau Solusi Penerapan Cache AWS untuk aset-aset yang sering digunakan untuk meningkatkan performa, mengurangi perpindahan data, dan mengurangi dampak pada lingkungan.

Layanan	Kapan harus digunakan
Amazon CloudFront	Gunakan untuk menyimpan konten statis seperti gambar, skrip, dan video, serta konten dinamis seperti API respons atau aplikasi web.

Layanan	Kapan harus digunakan
Amazon ElastiCache	Gunakan untuk meng-cache konten bagi aplikasi web.
DynamoDB Accelerator	Gunakan untuk menambahkan percepatan dalam memori ke tabel DynamoDB Anda.

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

Layanan	Kapan harus digunakan
Lambda@Edge	Gunakan untuk operasi-operasi yang memiliki banyak komputasi yang dimulai saat objek tidak ada dalam cache.
CloudFront Fungsi Amazon	Gunakan untuk kasus penggunaan sederhana seperti HTTP permintaan atau manipulasi respons yang dapat dimulai oleh fungsi berumur pendek.
AWS IoT Greengrass	Gunakan untuk menjalankan komputasi lokal, olahpesan, dan caching data untuk perangkat yang terhubung.

- Beberapa aplikasi memerlukan titik masuk tetap atau performa yang lebih tinggi dengan mengurangi jitter dan latensi bita pertama, dan meningkatkan throughput. Aplikasi ini dapat mengambil manfaat dari layanan jaringan yang menyediakan alamat IP anycast statis dan TCP penghentian di lokasi tepi. AWS Global Acceleratordapat meningkatkan kinerja untuk aplikasi Anda hingga 60% dan memberikan failover cepat untuk arsitektur multi-wilayah. AWS Global Accelerator memberi Anda alamat IP anycast statis yang berfungsi sebagai titik masuk tetap untuk aplikasi Anda yang dihosting dalam satu atau lebih Wilayah AWS. Alamat IP ini memungkinkan lalu lintas masuk ke jaringan AWS global sedekat mungkin dengan pengguna Anda. AWS Global Accelerator mengurangi waktu penyiapan koneksi awal dengan membuat TCP koneksi antara klien dan lokasi AWS tepi yang paling dekat dengan klien. Tinjau penggunaan AWS Global Accelerator untuk meningkatkan kinerja TCP UDP /beban kerja Anda dan menyediakan failover cepat untuk arsitektur Multi-region.

Sumber daya

Praktik-praktik terbaik terkait:

- COST07-BP02 Melaksanakan Daerah berdasarkan biaya
- COST08-BP03 Menerapkan layanan untuk mengurangi biaya transfer data
- REL10-BP01 Menyebarkan beban kerja ke beberapa lokasi
- REL10-BP02 Pilih lokasi yang sesuai untuk penyebaran multi-lokasi Anda
- SUS01-BP01 Pilih Wilayah berdasarkan persyaratan bisnis dan tujuan keberlanjutan
- SUS02-BP04 Optimalkan penempatan geografis beban kerja berdasarkan persyaratan jaringan mereka
- SUS04-BP07 Meminimalkan pergerakan data di seluruh jaringan

Dokumen terkait:

- AWS Infrastruktur Global
- AWS Local Zones dan AWS Outposts, memilih teknologi yang tepat untuk beban kerja edge Anda
- Grup penempatan
- AWS Local Zones
- AWS Outposts
- AWS Wavelength
- Amazon CloudFront
- AWS Global Accelerator
- AWS Direct Connect
- AWS Site-to-Site VPN
- Amazon Route 53

Video terkait:

- AWS Video Penjelasan Local Zones
- AWS Outposts: Ikhtisar dan Cara Kerjanya
- AWS re:invent 2023 - Strategi migrasi untuk beban kerja edge dan lokal

- AWS Re:invent 2021 - AWS Outposts: Membawa pengalaman di tempat AWS
- AWS re:invent 2020: AWS Wavelength: Jalankan aplikasi dengan latensi ultra-rendah di tepi 5G
- AWS re:invent 2022 - AWS Local Zones: Membangun aplikasi untuk tepi terdistribusi
- AWS re:invent 2021 - Membangun situs web latensi rendah dengan Amazon CloudFront
- AWS re:invent 2022 - Tingkatkan kinerja dan ketersediaan dengan AWS Global Accelerator
- AWS re:invent 2022 - Bangun jaringan area luas global Anda menggunakan AWS
- AWS re: invent 2020: Manajemen lalu lintas global dengan Amazon Route 53

Contoh terkait:

- AWS Global Accelerator Lokakarya Perutean Kustom
- Menangani Penulisan Ulang dan Pengarahan Ulang dengan 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.

Anti-pola umum:

- Anda beranggapan bahwa semua masalah yang berkaitan dengan kinerja disebabkan oleh aplikasi.
- Anda hanya menguji performa jaringan dari sebuah 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 menerapkan 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 diterapkan: Rendah

Panduan implementasi

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

Langkah-langkah implementasi

- Identifikasikan metrik kinerja utama seperti latensi atau kehilangan paket yang akan dikumpulkan. AWS menyediakan beberapa alat yang dapat membantu Anda mengumpulkan metrik-metrik ini. Dengan menggunakan alat-alat berikut, Anda dapat melakukan memeriksa informasi lebih lanjut tentang penggunaan lalu lintas, akses jaringan, dan log:

Alat AWS	Harus digunakan di mana
Pengelola Alamat IP Amazon VPC.	Gunakan IPAM untuk merencanakan, melacak, dan memantau alamat IP untuk AWS dan beban kerja on-premise Anda. Ini adalah praktik terbaik yang bisa digunakan untuk mengoptimalkan alokasi dan penggunaa n alamat IP.
Log Alur VPC	Gunakan VPC Flow Log untuk merekam 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.
Log Alur AWS Transit Gateway	Gunakan Log Alur AWS Transit Gateway untuk merekam informasi tentang lalu lintas IP yang masuk dan keluar gateway transit Anda.
Pencatatan log kueri DNS	Informasi log tentang kueri DNS publik atau privat yang diterima Route 53. Dengan log DNS, Anda dapat mengoptimalkan konfigura

Alat AWS	Harus digunakan di mana
	si DNS dengan memahami domain atau sub-domain yang diminta atau lokasi EDGE Route 53 yang merespons kueri DNS.
Reachability Analyzer	Reachability Analyzer akan membantu Anda menganalisis dan men-debug jangkauan jaringan. Reachability Analyzer adalah sebuah alat analisis konfigurasi yang memungkin kan Anda melakukan pengujian konektivi tas antara sumber daya sumber dan sumber daya destinasi di VPC Anda. Alat ini dapat membantu Anda memverifikasi bahwa konfigurasi jaringan Anda sesuai dengan konektivitas yang ditarget.
Penganalisis Akses Jaringan	Penganalisis Akses Jaringan akan membantu Anda memahami akses jaringan ke sumber daya Anda. Anda dapat menggunak an Penganalisis Akses Jaringan untuk menentukan persyaratan-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 yang ada di AWS memenuhi persyaratan kepatuhan Anda.

Alat AWS	Harus digunakan di mana
Amazon CloudWatch	Gunakan 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, Penyeimbangan Beban Elastis, dan AWS Direct Connect. Melakukan pemantauan metrik secara terus-menerus 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 riwayat Jaringan GlobalAWS untuk tujuan operasional dan perencanaan. Network Manager menyediakan 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-metrik yang memberi Anda wawasan dan membantu Anda mengidentifikasi, memahami, dan meningkatkan pengalaman pengguna.

- Identifikasikan sumber data terbesar dan pola lalu lintas aplikasi dengan menggunakan VPC dan AWS Transit Gateway Flow Logs.
- Lakukan penilaian 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 yang ada dalam arsitektur Anda.
- Nilai jalur perutean di jaringan Anda untuk memastikan digunakannya jalur terpendek antartujuan. Penganalisis Akses Jaringan dapat membantu Anda melakukan ini.

Sumber daya

Dokumen terkait:

- Pencatatan log kueri DNS publik
- Apa itu IPAM?
- Apa itu Reachability Analyzer?
- Apa itu Penganalisis Akses Jaringan?
- Metrik CloudWatch untuk VPC Anda
- Mengoptimalkan performa dan mengurangi biaya untuk analitik jaringan dengan VPC Flow Logs dalam format Apache Parquet
- Memantau jaringan inti dan global 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 lebih dalam 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 temuan 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.
- Pipeline deployment: Gunakan pipeline deployment yang berkelanjutan/terintegrasi terus-menerus (CI/CD) (misalnya, repositori 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 ditentukan dengan baik: Atur dan pantau metrik untuk merekam 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.
- Jalankan tes kinerja secara otomatis: Sebagai bagian dari proses deployment Anda, mulai tes kinerja secara otomatis setelah tes yang lebih cepat berhasil dijalankan. 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.

- simulasi pengujian beban kerja: Anda harus membuat serangkaian skenario pengujian yang mereplikasi atau merekam perjalanan pengguna yang direkayasa. Skrip ini harus idempoten dan tidak dipasangkan, dan Anda mungkin harus menyertakan skrip prapemanasan 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 — yang merupakan cara yang hemat untuk simulasi pengujian beban kerja.
- Visibilitas kinerja: Metrik utama harus dapat dilihat oleh tim Anda, khususnya metrik untuk setiap versi yang dibangun. 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.
- Tinjau proses secara 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.
- Optimalisasi terus-menerus: 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-area yang mementingkan kinerja
- PERF05-BP03 Menetapkan sebuah proses untuk meningkatkan kinerja beban kerja
- PERF05-BP04 Uji beban kerja Anda
- PERF05-BP05 Menggunakan otomatisasi untuk secara proaktif memulihkan masalah terkait kinerja
- PERF05-BP06 Pertahankan beban kerja dan layanan Anda up-to-date
- 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 untuk mengukur kesehatan dan kinerja beban kerja yang terkait dengan tujuan bisnis.

Anti-pola umum:

- Anda hanya memantau metrik tingkat sistem untuk memperoleh wawasan tentang beban kerja Anda dan tidak memahami dampak bisnis yang diakibatkan 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 kondisi kesehatan dan performa beban kerja dapat membantu Anda dalam menyelaraskan tim pada prioritas mereka dan menentukan hasil bisnis yang sukses. Ketika metrik-metrik tersebut dibagikan kepada semua departemen, akan ada visibilitas dan kesepakatan tentang ambang batas, harapan, dan dampak bisnis.

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

Panduan implementasi

KPI akan 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 sebuah 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 juga 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 Anda menetapkan KPI.

Tim Anda harus mengevaluasi KPI beban kerja Anda dengan menggunakan data terperinci 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 dan dipertahankan ketika tujuan bisnis, strategi, dan kebutuhan pengguna akhir berubah.

Langkah-langkah implementasi

- Identifikasi pemangku kepentingan: Mengidentifikasi dan mendokumentasikan para pemangku kepentingan bisnis utama, termasuk tim pengembangan dan operasi.
- Tentukan tujuan: Bekerjalah dengan para pemangku kepentingan ini untuk menentukan dan mendokumentasikan tujuan beban kerja Anda. Pertimbangkan aspek-aspek kinerja penting dari 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 tujuan-tujuan beban kerja Anda.
- Identifikasi metrik: Identifikasi metrik-metrik yang selaras dengan sasaran beban kerja Anda dan dapat membantu Anda mengukur kinerja dan tujuan-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 tujuan-tujuan 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 dari suatu KPI yang tidak terpenuhi.
- Menerapkan pemantauan: Gunakan alat-alat pemantauan seperti Amazon CloudWatch atau AWS Config untuk mengumpulkan metrik dan mengukur KPI.
- Komunikasikan KPI secara visual: Gunakan alat-alat dasbor seperti Amazon Quick untuk memvisualisasikan dan mengomunikasikan KPI dengan para pemangku kepentingan.
- Analisa dan optimalkan: Tinjau dan lakukan analisis KPI secara rutin untuk mengidentifikasi area beban kerja Anda yang perlu ditingkatkan. Bekerjalah dengan para pemangku kepentingan untuk mengimplementasikan perbaikan-perbaikan tersebut.
- Pertahankan dan perbaiki: Tinjau metrik dan KPI secara teratur untuk menilai efektivitasnya, terutama ketika tujuan bisnis atau kinerja beban kerja berubah.

Sumber daya

Dokumen terkait:

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

Video terkait:

- AWS re:Invent 2023 - Mengoptimalkan biaya dan kinerja serta melacak kemajuan menuju mitigasi
- AWS re:invent 2023 - Mengelola peristiwa siklus hidup sumber daya sesuai skala dengan AWS Health
- AWS re:Invent 2023 - Kinerja & efisiensi di Pinterest: Mengoptimalkan instans terbaru
- AWS re:Invent 2022 - Optimalisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung
- AWS re:Invent 2023 - Membangun strategi observabilitas yang efektif
- AWS Summit SF 2022 - Observabilitas tumpukan penuh (full-stack) dan pemantauan aplikasi dengan AWS
- AWS re:Invent 2023 - Menskalakan di AWS untuk 10 juta pengguna pertama
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- Membuat Strategi Metrik yang Efektif untuk Bisnis Anda | Peristiwa AWS

Contoh terkait:

- Membuat dasbor dengan Quick

PERF05-BP02 Menggunakan solusi pemantauan untuk memahami area-area yang mementingkan kinerja

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 diuntungkan oleh penggunaan layanan edge untuk membuat penyampaian konten ke pelanggan menjadi lebih dekat.

Anti-pola umum:

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

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

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

Panduan implementasi

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

Tingkatkan efisiensi kinerja dengan memahami arsitektur, pola lalu lintas, dan pola akses data Anda, serta lakukan identifikasi latensi dan waktu pemrosesan Anda. Lakukan juga identifikasi terhadap 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-masalah kinerja tersebut.

Langkah-langkah implementasi

- Siapkan pemantauan secara menyeluruh (ujung-ke-ujung) untuk mengetahui semua komponen dan metrik beban kerja. Berikut adalah contoh solusi pemantauan di AWS.

Layanan	Harus digunakan di mana
Pemantauan Pengguna Nyata (RUM) Amazon CloudWatch	Untuk merekam metrik-metrik performa aplikasi dari sesi sisi klien dan frontend pengguna nyata.
AWS X-Ray	Untuk melacak lalu lintas melalui lapisan-l apisan aplikasi dan mengidentifikasi latensi yang ada antara komponen dan dependens i. Gunakan peta layanan X-Ray untuk melihat hubungan dan latensi yang ada antara komponen beban kerja.
Wawasan Performa Layanan Basis Data Relasional Amazon	Untuk melihat metrik-metrik kinerja basis data dan mengidentifikasi peningkatan kinerja.
Pemantauan yang Ditingkatkan Amazon RDS	Untuk melihat metrik-metrik kinerja OS basis data.
Amazon DevOps Guru	Untuk mendeteksi pola operasi yang tidak normal sehingga Anda dapat mengidentifikasi setiap masalah operasional sebelum masalah tersebut berdampak pada para pelanggan Anda.

- Lakukan pengujian untuk membuat metrik, mengidentifikasi pola lalu lintas, hambatan, dan mengidentifikasi area-area kinerja kritis. Berikut adalah beberapa contoh cara melakukan pengujian:
  - Siapkan Canary CloudWatch Synthetic untuk meniru aktivitas pengguna berbasis browser secara terprogram dengan 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-area kinerja kritis Anda. Tinjau area-area ini bersama 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-peningkatan baru dan dampak kinerja yang ditimbulkan terhadap beban kerja Anda.

Sumber daya

Dokumen terkait:

- Yang baru di Observabilitas AWS di re:Invent 2023
- Amazon Builders' Library
- Dokumentasi X-Ray
- Amazon CloudWatch RUM
- 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 - Observabilitas tumpukan penuh (full-stack) dan pemantauan aplikasi dengan AWS
- AWS re:Invent 2022 - Optimisasi AWS: Langkah-langkah yang dapat ditindaklanjuti untuk hasil langsung
- AWS re:Invent 2022 - Amazon Builders' Library: 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 Amazon CloudWatch RUM
- SDK X-Ray untuk Python
- Pengujian Beban Terdistribusi di AWS

PERF05-BP03 Menetapkan sebuah proses untuk meningkatkan kinerja beban kerja

Menetapkan sebuah 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.

Anti-pola 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: Setelah proses untuk membuat perubahan arsitektur ditetapkan, Anda dapat menggunakan data yang dikumpulkan untuk memengaruhi desain beban kerja Anda seiring waktu.

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

Panduan implementasi

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

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

Langkah-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 dijadikan patokan dasar beban kerja Anda.
- Terapkan pemantauan: Gunakan alat-alat observabilitas AWS untuk mengumpulkan metrik-metrik kinerja dan mengukur KPI.
- Lakukan analisis: Lakukan analisis mendalam untuk mengidentifikasi area-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-area yang mementingkan kinerja. Gunakan alat-alat analisis dan kinerja Anda untuk mengidentifikasi strategi perbaikan kinerja.
- Validasi perbaikan: Gunakan sandbox atau lingkungan pra-produksi untuk memvalidasi efektivitas strategi perbaikan.
- Terapkan perubahan: Implementasikan perubahan-perubahan dalam lingkungan produksi dan terus pantau kinerja beban kerja. Dokumentasikan perbaikan, dan komunikasikan perubahan-perubahan kepada para pemangku kepentingan.
- Pertahankan dan perbaiki: Tinjau proses peningkatan kinerja Anda secara rutin untuk mengidentifikasi area-area yang bisa ditingkatkan lagi.

Sumber daya

Dokumen terkait:

- AWS Blog
- 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:

- AWS Github

PERF05-BP04 Uji 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.

Anti-pola 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 Anda memperkirakan area-area yang mungkin akan bermasalah di masa depan.
- Anda melakukan pengujian beban tanpa berkonsultasi dengan Kebijakan EC2 Pengujian Amazon dan mengirimkan Formulir Pengiriman Acara Simulasi. Ini mengakibatkan pengujian Anda gagal dijalankan, karena terlihat seperti denial-of-service acara.

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

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

Panduan implementasi

Pengujian beban di cloud adalah sebuah proses untuk mengukur kinerja beban kerja cloud dalam kondisi realistis dengan beban pengguna yang diharapkan. Proses ini melibatkan penyediaan lingkungan cloud yang mirip lingkungan produksi, penggunaan alat-alat pengujian beban untuk menghasilkan beban, dan analisis metrik untuk menilai kemampuan penanganan beban kerja Anda yang realistis. Pengujian beban harus dijalankan menggunakan versi data produksi yang sintetis atau sudah dibersihkan (menghapus informasi sensitif atau pengidentifikasi). Lakukan pengujian beban secara otomatis sebagai bagian dari jalur pengiriman Anda, dan bandingkan hasilnya dengan yang

telah ditentukan sebelumnya KPIs dan ambang batas. Proses ini akan membantu Anda untuk terus mencapai kinerja yang dibutuhkan.

Langkah-langkah implementasi

- Tentukan tujuan pengujian Anda: Identifikasi aspek-aspek kinerja beban kerja Anda yang ingin Anda evaluasi, misalnya 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 AWS layanan untuk menjalankan lingkungan skala produksi untuk menguji arsitektur Anda.
- Terapkan pemantauan: Gunakan alat pemantauan seperti Amazon CloudWatch untuk mengumpulkan metrik di seluruh sumber daya dalam arsitektur Anda. Anda juga dapat mengumpulkan dan menerbitkan metrik-metrik kustom.
- Tentukan skenario: Tentukan skenario dan parameter pengujian beban (seperti durasi pengujian dan jumlah pengguna).
- Melakukan pengujian beban: Melakukan skenario pengujian dalam skala besar. Manfaatkan AWS Cloud untuk menguji beban kerja Anda untuk menemukan di mana ia gagal untuk menskalakan, atau jika diskalakan dengan cara non-linier. Misalnya, gunakan Instans Spot untuk menghasilkan beban dengan biaya rendah dan temukan hambatan sebelum hambatan tersebut dialami di lingkungan produksi.
- Analisis hasil: Analisis hasil untuk mengidentifikasi hambatan kinerja dan area untuk perbaikan.
- Dokumentasikan dan bagikan temuan: Buatlah dokumentasi dan laporan mengenai temuan dan rekomendasi. Bagikan informasi ini kepada para pemangku kepentingan untuk membantu mereka mengambil keputusan yang cerdas mengenai strategi optimalisasi kinerja.
- Ulangi terus-menerus: Pengujian beban harus dilakukan pada irama reguler, terutama setelah perubahan pembaruan sistem.

Sumber daya

Dokumen terkait:

- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics

- Pengujian Beban Terdistribusi pada AWS

Video terkait:

- AWS Summit ANZ 2023: Mempercepat dengan percaya diri melalui Pengujian Beban AWS Terdistribusi
- AWS re:invent 2022 - Menskalakan AWS untuk 10 juta pengguna pertama Anda
- Memecahkan dengan AWS Solusi: Pengujian Beban Terdistribusi

- AWS re:invent 2021 - Optimalkan aplikasi melalui wawasan pengguna akhir dengan Amazon CloudWatch RUM
- Demo dari Amazon CloudWatch Synthetics

Contoh terkait:

- Pengujian Beban Terdistribusi pada 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.

Anti-pola umum:

- Anda hanya membekali staf operasional dengan kemampuan untuk membuat perubahan-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 akan memungkinkan staf dukungan untuk berkonsentrasi pada item-item yang tidak dapat ditindaklanjuti secara otomatis. Hal ini akan membantu staf operasi dalam menangani semua alarm tanpa merasa kewalahan dan mereka hanya berkonsentrasi pada alarm yang kritis.

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

Panduan implementasi

Gunakan alarm untuk memicu tindakan-tindakan otomatis untuk memperbaiki masalah ketika memungkinkan. Teruskan eskalasi alarm ke personel yang mampu merespons jika respons otomatis tidak memungkinkan. Misalnya, Anda mungkin memiliki sebuah sistem yang dapat memprediksi nilai dan alarm indikator kinerja utama (KPI) yang diharapkan ketika nilai dan alarm 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 mempunyai performa yang optimal.

Langkah-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 untuk memulai proses remediasi secara otomatis. Misalnya, Anda dapat menentukan pemicu untuk memulai ulang instans secara otomatis ketika instans tersebut mencapai ambang batas pemanfaatan CPU tertentu.
- Otomatiskan remediasi: Gunakan layanan dan teknologi AWS untuk mengotomatiskan proses perbaikan. Sebagai contoh, AWS Systems Manager Automation 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.
- Terapkan alur kerja: Terapkan remediasi otomatis di lingkungan produksi.
- Kembangkan playbook: Kembangkan dan dokumentasikan playbook yang menguraikan langkah-langkah untuk rencana remediasi, termasuk peristiwa inisiasi, logika remediasi, dan tindakan yang diambil. Pastikan Anda melatih pemangku kepentingan untuk membantu mereka merespons peristiwa-peristiwa perbaikan otomatis secara efektif.
- Tinjau dan perbaiki: Secara teratur lakukan evaluasi terhadap efektivitas alur kerja remediasi otomatis. Sesuaikan peristiwa inisiasi dan logika perbaikan jika perlu.

Sumber daya

Dokumen terkait:

- Dokumentasi CloudWatch
- Partner AWS Partner Network Pemantauan, Pencatatan Log, dan Performa
- Dokumentasi X-Ray
- Menggunakan Alarm dan Tindakan Alarm di CloudWatch
- Membangun Praktik Otomatisasi Cloud untuk Keunggulan Operasional: Praktik Terbaik dari AWS Managed Services
- Otomatiskan penyetelan kinerja Amazon Redshift Anda dengan pengoptimalan tabel otomatis

Video terkait:

- AWS re:Invent 2023 - Strategi untuk penskalaan otomatis, remediasi, dan penyembuhan 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 - Melakukan otomatisasi cerdas terhadap operasi cloud
- AWS re:Invent 2022 - Menyiapkan kontrol dalam skala besar di lingkungan AWS Anda
- AWS re:Invent 2022 - Melakukan otomatisasi manajemen dan kepatuhan patch dengan menggunakan AWS
- AWS re:Invent 2022 - Bagaimana Amazon menggunakan metrik yang lebih baik untuk meningkatkan kinerja situs web
- AWS re:Invent 2023 - Matikan beban: Mendiagnosis & menyelesaikan 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 - Pusatkan operasi Anda

Contoh terkait:

- Alarm Log Cloudwatch yang Disesuaikan

PERF05-BP06 Pertahankan beban kerja dan layanan Anda up-to-date

Tetap up-to-date gunakan layanan dan fitur cloud baru untuk mengadopsi fitur yang efisien, menghapus masalah, dan meningkatkan efisiensi kinerja keseluruhan beban kerja Anda.

Anti-pola umum:

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

Manfaat membangun praktik terbaik ini: Dengan membangun proses untuk tetap menggunakan layanan dan up-to-date penawaran baru, Anda dapat mengadopsi fitur dan kemampuan baru, menyelesaikan masalah, dan meningkatkan kinerja beban kerja.

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

Panduan implementasi

Evaluasi cara-cara yang dilakukan untuk 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 pelaksanaan evaluasi, diskusi internal, atau analisis eksternal. Tentukan sebuah proses untuk mengevaluasi pembaruan, fitur baru, dan layanan yang relevan dengan beban kerja Anda. Misalnya, bangunlah sebuah bukti konsep yang memanfaatkan teknologi baru atau berkonsultasi dengan grup internal. Saat Anda mencoba layanan atau ide baru, jalankan pengujian kinerja untuk mengukur pengaruhnya terhadap kinerja beban kerja.

Langkah-langkah implementasi

- Buat inventaris 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 AWS Blog What's New at untuk produk yang sesuai dengan komponen beban kerja Anda. Anda dapat berlangganan RSS feed 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 Inventaris untuk mengumpulkan sistem operasi (OS), aplikasi, dan metadata instans dari instans Amazon Anda dan dengan cepat memahami EC2 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 melakukan uji cepat mengenai bagaimana fitur-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 memperbaruiAMIs, gambar kontainer, dan artefak lain yang terkait dengan aplikasi cloud Anda secara otomatis.
  - Anda dapat menggunakan alat-alat seperti AWS Systems Manager Patch Manager untuk melakukan otomatisasi terhadap proses pembaruan sistem, dan menjadwalkan aktivitas dengan menggunakan AWS Systems Manager Windows Maintenance.
- Dokumentasikan proses: Dokumentasikan proses Anda untuk mengevaluasi pembaruan dan layanan baru. Bekali para pemilik Anda dengan waktu dan ruang yang dibutuhkan untuk meneliti, menguji, melakukan eksperimen, serta memvalidasi pembaruan dan layanan baru. Lihat kembali persyaratan bisnis yang terdokumentasi dan KPIs untuk membantu memprioritaskan pembaruan mana yang akan membuat dampak bisnis yang positif.

Sumber daya

Dokumen terkait:

- Blog AWS
- Apa yang baru dengan AWS
- Menerapkan up-to-date gambar dengan pipeline EC2 Image Builder otomatis

Video terkait:

- AWS Re: Inforce 2022 - Mengotomatiskan manajemen patch dan kepatuhan menggunakan AWS
- Semua Hal Patch: AWS Systems Manager | AWS Acara

Contoh terkait:

- Manajemen Inventaris dan Patch
- Lokakarya Satu Observabilitas

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.

Anti-pola umum:

- Anda mengizinkan metrik-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: Lakukan peninjauan secara terus-menerus terhadap metrik yang sedang dikumpulkan untuk memverifikasi bahwa metrik tersebut dapat mengidentifikasi, mengatasi, atau mencegah masalah. Metrik juga dapat mengalami kedaluwarsa jika Anda membiarkannya berada dalam status alarm untuk waktu yang lama.

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

Panduan implementasi

Lakukan peningkatan pemantauan dan pengumpulan metrik secara konstan. 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 Anda dapat menangani insiden pada masa mendatang dengan lebih cepat.

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 Anda dapat mencegah atau dapat mengatasi insiden di masa mendatang dengan lebih cepat.

Langkah-langkah implementasi

- Tentukan metrik: Tentukan metrik kinerja penting untuk memantau apakah mereka 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.
- Tetapkan frekuensi: Tetapkan frekuensi (seperti mingguan atau bulanan) untuk meninjau metrik penting.
- Identifikasi masalah performa: Dalam setiap tinjauan, nilai tren dan penyimpangan dari nilai garis acuan. Cari setiap anomali atau hambatan performa. Untuk masalah-masalah yang berhasil diidentifikasi, lakukan analisis akar penyebab secara mendalam untuk memahami alasan utama di balik masalah tersebut.
- Identifikasi tindakan-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 diidentifikasi, akar penyebab, dan tindakan korektif.
- Ulangi dan tingkatkan: Terus nilai dan tingkatkan proses peninjauan metrik. Gunakan pelajaran yang dipetik dari tinjauan sebelumnya untuk menyempurnakan proses dari waktu ke waktu.

Sumber daya

Dokumen terkait:

- Dokumentasi CloudWatch
- Mengumpulkan metrik dan log dari server Instans Amazon EC2 dan server on-premise dengan Agen CloudWatch
- Buat kueri metrik Anda dengan Wawasan Metrik CloudWatch
- Partner AWS Partner Network Pemantauan, Pencatatan Log, dan Performa
- 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 - Observabilitas tumpukan penuh (full-stack) dan pemantauan aplikasi dengan AWS
- AWS re:Invent 2023 - Matikan beban: Mendiagnosis & menyelesaikan masalah kinerja dengan Amazon RDS

Contoh terkait:

- Membuat dasbor dengan Quick
- 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 berusaha untuk membantu Anda membangun arsitektur yang berkinerja efisien sambil memberikan nilai bisnis. Gunakan alat dan teknik yang dibahas dalam artikel ini untuk memastikan keberhasilan.

Kontributor

Individu dan organisasi berikut berkontribusi terhadap dokumen ini:

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

Sumber bacaan lebih lanjut

Untuk mendapatkan bantuan tambahan, konsultasikan dengan sumber berikut:

- Kerangka Kerja AWS Well-Architected
- Pusat Arsitektur AWS

Revisi dokumen

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

Perubahan	Deskripsi	Tanggal
Pembaruan kecil praktik terbaik	PERF03-BP04 diperbarui dengan rekomendasi layanan baru.	6 November 2024
Panduan praktik terbaik yang sudah diperbarui	Beberapa pembaruan kecil di seluruh pilar.	27 Juni 2024
Pembaruan dan restrukturisasi besar	Pilar direstrukturisasi menjadi lima area praktik terbaik (turun dari delapan). Konten telah dikonsolidasikan ke dalam lima area dan diperbarui. Bidang-bidang praktik terbaik yang baru adalah Pemilihan arsitektur, Komputasi dan perangkat keras, Manajemen data, Jaringan dan pengiriman konten, serta Proses dan budaya.	3 Oktober 2023
Pembaruan kecil	Bahasa non-inklusif dihilangkan.	13 April 2023
Pembaruan untuk Kerangka Kerja baru	Praktik terbaik diperbarui dengan panduan preskriptif dan praktik terbaik baru ditambahkan.	10 April 2023

Laporan resmi diperbarui	Praktik terbaik sudah diperbarui dengan panduan implementasi yang baru.	15 Desember 2022
Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan sudah ditambahkan.	20 Oktober 2022
Pembaruan kecil	Bahasa noninklusif dihilangkan.	22 April 2022
Pembaruan kecil	Tautan diperbarui.	10 Maret 2021
Pembaruan kecil	Mengubah batas waktu AWS Lambda menjadi 900 detik dan perbaikan nama Amazon Keyspaces (untuk Apache Cassandra).	5 Oktober 2020
Pembaruan kecil	Tautan yang bermasalah diperbaiki.	15 Juli 2020
Pembaruan untuk Kerangka Kerja baru	Peninjauan dan pembaruan besar konten	8 Juli 2020
Laporan resmi diperbarui	Pembaruan kecil masalah gramatikal	1 Juli 2018
Laporan resmi diperbarui	Laporan resmi disegarkan untuk mencerminkan perubahan di AWS	1 November 2017
Publikasi awal	Pilar Efisiensi Kinerja - Kerangka Kerja AWS Well-Architected diterbitkan.	1 November 2016

Pemberitahuan

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

© 2023 Amazon Web Services, Inc. atau afiliasinya. Semua hak dilindungi undang-undang.

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS