Praktik terbaik untuk merampingkan observabilitas Amazon EKS

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Praktik terbaik untuk merampingkan observabilitas Amazon EKS

Table of Contents

Pengantar 1
Tujuan 2
Pencatatan log 4
Jenis penebangan 4
Log sistem 5
Log komponen Kubernetes 6
Log runtime kontainer 7
Log aplikasi 8
Praktik terbaik 8
Pertimbangan penting 9
Memantau 12
Jenis pemantauan 12
Pemantauan Infrastruktur 12
Pemantauan aplikasi 13
Pemantauan keamanan 14
Alat 15
AWS layanan 15
Solusi open source atau proprietary 17
Alat khusus 18
Menerapkan ketersediaan tinggi 19
Redundansi dan skalabilitas arsitektur 19
Strategi penyimpanan data yang tangguh 19
Manajemen peringatan redundan 19
Load balancing dan penemuan layanan 20
Pertimbangan HA tambahan 20
Praktik terbaik 21
Pendekatan implementasi strategis 21
Manajemen data yang efektif 22
Konfigurasi dan manajemen peringatan 22
Optimalisasi sumber daya 23
Keamanan 14
Pertimbangan lanjutan 24
Pelacakan 25
Alat 27

Layanan AWS 27
Solusi sumber terbuka 27
Praktik terbaik 28
Peringatan 30
Alat 30
Praktik terbaik 31
Langkah berikutnya 36
Sumber daya 37
AWS dokumentasi 37
AWS posting blog 37
Sumber daya lainnya 37
Riwayat dokumen 38
Glosarium 39
..... 39
A 40
B 43
C 45
D 48
E 52
F 54
G 56
H 57
I 59
L 61
M 63
O 67
P 70
Q 73
R 73
D 76
T 80
U 81
V 82
W 82
Z 83
..... lxxxv

Praktik terbaik untuk merampingkan observabilitas Amazon EKS

Ishwar Chauthaiwale, Naveen Suthar, dan Pratap Kumar Nanda, Amazon Web Services ()AWS

Maret 2026 (sejarah dokumen)

Amazon Elastic Kubernetes Service (Amazon EKS) memerlukan solusi observabilitas yang komprehensif untuk memantau dan memecahkan masalah beban kerja kontainer secara efektif. Sistem terdistribusi dan layanan mikro memiliki arsitektur yang kompleks di lingkungan Amazon EKS, sehingga menerapkan praktik pengamatan yang tepat sangat penting untuk mempertahankan operasi yang andal. Observabilitas yang efektif di lingkungan Amazon EKS memungkinkan tim memperoleh wawasan mendalam tentang kinerja aplikasi, memecahkan masalah secara efisien, dan menjaga kesehatan klaster yang optimal.

Tantangannya terletak pada menavigasi ekosistem alat dan teknik yang luas yang tersedia untuk observabilitas Amazon EKS sambil mengikuti praktik terbaik yang selaras dengan tujuan organisasi dan standar industri. Strategi observabilitas yang efektif harus menyeimbangkan pengumpulan data yang komprehensif dengan pertimbangan kinerja, efektivitas biaya, dan skalabilitas.

Panduan ini dirancang untuk membantu organisasi mengoptimalkan observabilitas Amazon EKS mereka di seluruh area berikut:

- Membangun mekanisme penebangan yang efisien
- Menerapkan solusi pemantauan yang kuat
- Menggunakan penelusuran terdistribusi untuk arsitektur yang kompleks
- Menerapkan strategi peringatan dan respons insiden

Dengan mengadopsi praktik terbaik ini, organisasi Anda dapat meningkatkan kemampuan mereka untuk mendapatkan wawasan mendalam tentang lingkungan Amazon EKS mereka, yang mengarah pada peningkatan keandalan, kinerja, dan efisiensi operasional. Pendekatan yang efisien untuk observabilitas ini membantu dalam pemecahan masalah dan pemeliharaan, dan mendukung pengambilan keputusan berbasis data untuk perbaikan berkelanjutan aplikasi dan infrastruktur berbasis Kubernetes. (Untuk informasi rinci tentang Amazon EKS, lihat dokumentasi layanan.)

Panduan ini menyelami setiap aspek observabilitas Amazon EKS dan mengeksplorasi alat dan strategi yang dapat Anda sesuaikan untuk memenuhi kebutuhan spesifik penerapan Amazon EKS Anda, mulai dari aplikasi skala kecil hingga arsitektur layanan mikro yang besar dan kompleks.

Dalam panduan ini:

- Masuk Amazon EKS
- Pemantauan di Amazon EKS
- Menelusuri di Amazon EKS
- Peringatan di Amazon EKS
- Langkah selanjutnya
- Sumber Daya

Tujuan

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

- Visibilitas operasional yang ditingkatkan — Dapatkan wawasan komprehensif tentang kluster dan aplikasi Amazon EKS Anda melalui praktik observabilitas yang efektif.

  Tujuan ini menekankan pentingnya menjaga visibilitas lengkap di seluruh lingkungan Amazon EKS Anda. Alat seperti AWS X-RayAmazon CloudWatch Container Insights, dan AWS Distro untuk OpenTelemetry membantu Anda memahami perilaku sistem, mengidentifikasi masalah dengan cepat, dan mempertahankan kinerja optimal.
- Peningkatan efisiensi pemecahan masalah — Mengurangi mean time to detection (MTTD) dan mean time to resolution (MTTR) melalui strategi penelusuran dan pemantauan yang efektif.

  Tujuan ini berfokus pada penerapan praktik observabilitas yang memungkinkan identifikasi cepat dan penyelesaian masalah. Teknik seperti penelusuran terdistribusi, pencatatan yang efektif, dan pengumpulan metrik yang komprehensif adalah kunci untuk mencapai tujuan ini.
- Manajemen kinerja proaktif — Aktifkan deteksi dini potensi masalah sebelum memengaruhi pengguna akhir.

  Pemantauan proaktif sangat penting untuk menjaga ketersediaan dan kinerja layanan yang tinggi. Tujuan ini membahas pentingnya menerapkan peringatan yang tepat, analisis tren, dan pemantauan prediktif untuk mencegah gangguan layanan.

- Observabilitas hemat biaya — Optimalkan biaya observabilitas sambil mempertahankan visibilitas sistem yang komprehensif.

  Optimalisasi biaya mencakup penerapan strategi pengambilan sampel yang efisien, kebijakan retensi data yang tepat, dan pendekatan instrumentasi yang optimal. Tujuannya adalah untuk menyeimbangkan kebutuhan observabilitas dengan pertimbangan biaya sambil memastikan pemantauan sistem yang efektif.
- Arsitektur pemantauan yang dapat diskalakan — Pastikan bahwa solusi observabilitas Anda menskalakan secara mulus dengan lingkungan Amazon EKS Anda.

  Tujuan ini berfokus pada penerapan solusi pemantauan yang dapat tumbuh dengan aplikasi Anda. Baik Anda menjalankan satu cluster atau multi-cluster, penerapan Multi-region, strategi observabilitas Anda harus menskalakan sesuai

Masuk Amazon EKS

Logging adalah aspek penting dalam mengelola dan memelihara aplikasi yang berjalan di Amazon EKS. Praktik pencatatan yang efektif di lingkungan Amazon EKS membantu pengembang, tim operasi, dan administrator sistem mendapatkan wawasan berharga tentang perilaku, kinerja, dan kesehatan aplikasi kontainer dan infrastruktur dasarnya.

Menerapkan strategi logging yang kuat di Amazon EKS sangat penting karena beberapa alasan:

- Pemecahan masalah: Log membantu mengidentifikasi dan mendiagnosis masalah dengan cepat, yang mengurangi waktu henti dan meningkatkan keandalan sistem secara keseluruhan.
- Kepatuhan: Banyak industri memerlukan penebangan komprehensif untuk tujuan audit dan regulasi.
- Keamanan: Analisis log dapat membantu Anda mendeteksi dan menyelidiki potensi ancaman atau pelanggaran keamanan.
- Optimalisasi kinerja: Log memberikan wawasan tentang kinerja aplikasi dan sistem, sehingga Anda dapat mengidentifikasi kemacetan dan mengoptimalkan pemanfaatan sumber daya.
- Pemantauan dan peringatan: Data log dapat digunakan untuk mengatur sistem pemantauan dan memicu peringatan untuk peristiwa atau kondisi tertentu.

Di bagian ini:

- Jenis logging di Amazon EKS
- Praktik terbaik untuk masuk ke Amazon EKS
- Pertimbangan penting untuk masuk ke Amazon EKS

Jenis logging di Amazon EKS

Di Amazon EKS, logging melibatkan pengambilan, penyimpanan, dan analisis berbagai jenis data log yang dihasilkan oleh berbagai komponen cluster Kubernetes, termasuk:

- Log sistem: Informasi tentang instans atau node Amazon Elastic Compute Cloud (Amazon EC2) yang mendasari AWS Fargate
- Log komponen Kubernetes : Data dari komponen inti Kubernetes seperti server API, scheduler, dan manajer pengontrol

- Log runtime kontainer: Informasi dari runtime container, seperti Docker atau containerd
- Log aplikasi: Keluaran dari aplikasi kontainer

Untuk mengelola log di lingkungan Amazon EKS Anda secara efektif, Anda biasanya menggunakan kombinasi Layanan AWS, alat pihak ketiga, dan praktik terbaik. Ini mungkin termasuk menggunakan Amazon CloudWatch, Fluent Bit, Elasticsearch, Kibana, dan alat pencatatan dan analisis lainnya untuk mengumpulkan, menyimpan, dan memvisualisasikan data log.

Bagian berikut mengeksplorasi berbagai aspek logging di Amazon EKS, termasuk praktik terbaik, alat, dan teknik untuk menerapkan strategi logging komprehensif di klaster Kubernetes Anda. AWS

Log sistem

Logging untuk instans EC2 yang mendasari atau node Fargate di Amazon EKS melibatkan pendekatan yang berbeda tergantung pada jenis node.

Untuk menerapkan logging untuk instans EC2 di Amazon EKS, Anda dapat menggunakan alat berikut:

- CloudWatch agent: Instal dan konfigurasikan CloudWatch agen pada instans EC2 Anda. Konfigurasikan untuk mengumpulkan log sistem seperti `/var/log/messages` dan`/var/log/secure`. Anda dapat menggunakan skrip data pengguna atau alat manajemen konfigurasi untuk mengotomatiskan proses ini.
- Fluent Bit: Terapkan Fluent Bit sebagai a DaemonSet untuk mengumpulkan log dari semua node. Konfigurasikan untuk meneruskan CloudWatch log ke Log atau sistem logging terpusat lainnya.
- Wawasan Kontainer: Aktifkan Wawasan Kontainer di kluster EKS Anda untuk secara otomatis mengumpulkan metrik dan log dari instans EC2.
- Skrip khusus: Kembangkan skrip khusus untuk mengumpulkan log tertentu dan mengirimkannya ke tujuan pencatatan pilihan Anda.
- Agen SSM: Gunakan AWS Systems Manager Agen (Agen SSM) untuk mengumpulkan dan meneruskan log ke CloudWatch Log.

Untuk mengimplementasikan logging untuk node Fargate di Amazon EKS, gunakan alat ini:

- Fargate logging: Fargate secara otomatis mengumpulkan `stdout` dan `stderr` mencatat dari wadah Anda. Konfigurasikan profil Fargate Anda untuk mengirim log ini ke CloudWatch Log.

- Fluent Bit for Fargate AWS : menyediakan gambar Fluent Bit khusus untuk logging Fargate. Terapkan sebagai wadah sespan di pod Fargate Anda untuk mengumpulkan dan meneruskan log.
- Wawasan Kontainer untuk Fargate: Aktifkan Wawasan Kontainer untuk mengumpulkan metrik dan log dari node Fargate.

Log komponen Kubernetes

Mengumpulkan log dari komponen Kubernetes seperti server API, scheduler, dan manajer pengontrol di Amazon EKS memerlukan pendekatan yang sedikit berbeda dari pencatatan aplikasi. Komponen-komponen ini berjalan sebagai bagian dari bidang kontrol Amazon EKS, yang dikelola oleh AWS. Berikut cara mengumpulkan dan mengakses log ini:

- Aktifkan pencatatan bidang kontrol: Anda dapat mengaktifkan pencatatan bidang kontrol untuk cluster EKS Anda melalui alat Konsol Manajemen AWS, AWS Command Line Interface (AWS CLI), atau infrastruktur sebagai kode (IAc) seperti AWS CloudFormationatau Terraform. Saat Anda mengaktifkan pencatatan bidang kontrol, log akan dikirim ke Amazon CloudWatch Logs. Anda dapat melihatnya di CloudWatch konsol di grup `/aws/eks/<cluster-name>/cluster` log. Dalam grup log ini, setiap komponen bidang kontrol memiliki aliran lognya sendiri sebagai berikut:

Nama aliran	Deskripsi
apiserver kube	Log server API Kubernetes
penjadwal kube	Log keputusan penjadwal
kube-controller-manager	Log manajer pengontrol
autentikator	Log autentikator IAM
audit	Log audit Kubernetes (harus diaktifkan secara eksplisit)

  Untuk melihat log untuk komponen tertentu, navigasikan ke grup log cluster dan filter berdasarkan nama aliran log target.
- Gunakan Wawasan CloudWatch Log: Anda dapat menggunakan Wawasan CloudWatch Log untuk melakukan kueri kompleks pada log Anda.

- Ekspor log ke Amazon S3: Untuk penyimpanan jangka panjang atau analisis lebih lanjut, Anda dapat mengekspor log ke Amazon Simple Storage Service (Amazon S3).
- Gunakan alat pihak ketiga: Anda dapat menggunakan alat seperti Fluent Bit untuk mengumpulkan log ini dan meneruskannya ke sistem logging lain seperti Elasticsearch atau Splunk.
- Penggunaan AWS CloudTrail: AWS CloudTrailLayanan ini dapat memberikan wawasan tambahan tentang panggilan API yang dilakukan ke kluster EKS Anda.

Log runtime kontainer

Pencatatan log runtime kontainer di Amazon EKS melibatkan pengambilan dan pengelolaan log dari runtime container, yang biasanya untuk `containerd` Amazon EKS. Inilah cara Anda mendekati log runtime pencatatan container di Amazon EKS:

- Akses langsung log di node Amazon EC2. Untuk node EC2 yang dikelola sendiri, Anda dapat langsung mengakses log runtime container di host dari lokasi berikut:
  - `containerd`log: `/var/log/containers/`
  - Log Docker (jika Anda menggunakan runtime Docker): `/var/log/docker.log`
- Gunakan koleksi DaemonSet untuk log.
- Menyebarkan agen pengumpulan log (seperti Fluent Bit) sebagai DaemonSet untuk mengumpulkan log dari semua node.
- Konfigurasikan CloudWatch agen untuk mengumpulkan log runtime kontainer.
- Aktifkan Wawasan Kontainer untuk mengumpulkan metrik dan log runtime kontainer.
- Gunakan Fargate. Untuk node Fargate, log runtime kontainer dikumpulkan secara otomatis dan dapat diakses melalui Log. CloudWatch
- Menerapkan solusi logging kustom dengan menggunakan alat seperti Fluent Bit atau Logstash. Siapkan CloudWatchalarm atau gunakan alat seperti Prometheus untuk memantau pola atau masalah tertentu dalam log runtime kontainer. Pertimbangkan untuk menggunakan solusi logging pihak ketiga yang terintegrasi dengan baik dengan Kubernetes dan Amazon EKS, seperti Datadog, Splunk, atau Elastic Stack (ELK Stack). Gunakan alat agregasi log untuk mengumpulkan log dari berbagai sumber dan meneruskannya ke sistem logging terpusat.

Log aplikasi

Log aplikasi di Amazon EKS adalah bagian penting dari pemeliharaan dan pemecahan masalah aplikasi Anda. Untuk menerapkan pencatatan aplikasi di Amazon EKS, Anda dapat memilih dari opsi ini:

- Tulis log k`stdout`/`stderr`: Cara paling sederhana dan paling asli Kubernetes untuk menangani log aplikasi adalah dengan menuliskannya ke dan. `stdout` `stderr` Kubernetes secara otomatis menangkap aliran ini.
- Terapkan agregasi log: Gunakan agregator log seperti Fluent Bit untuk mengumpulkan log dari semua pod Anda.
- Konfigurasikan perutean log: Konfigurasikan agregator log Anda untuk merutekan log ke tujuan yang Anda inginkan (seperti CloudWatch Log atau Elasticsearch).
- Gunakan CloudWatch Wawasan Kontainer: Aktifkan Wawasan Kontainer untuk pencatatan dan pemantauan komprehensif.

Praktik terbaik untuk masuk ke Amazon EKS

Praktik terbaik berikut membantu menciptakan sistem logging yang kuat, dapat diskalakan, dan efisien untuk lingkungan Amazon EKS Anda, serta memberikan pemecahan masalah, pemantauan, dan pengelolaan keseluruhan klaster Kubernetes yang lebih baik.

- Memusatkan koleksi log: Gunakan solusi logging terpusat seperti CloudWatch Log, Elasticsearch, atau layanan pihak ketiga untuk mengumpulkan log dari semua komponen. Ini menyediakan satu titik akses untuk analisis log dan menyederhanakan manajemen.
- Terapkan logging terstruktur: Gunakan format log terstruktur seperti JSON sehingga log dapat diurai dan dicari dengan lebih mudah. Sertakan metadata yang relevan seperti stempel waktu, tingkat log, dan pengidentifikasi sumber.
- Gunakan tingkat log dengan tepat: Menerapkan tingkat log yang tepat (sepertiDEBUG,INFO,WARN, danERROR) dalam aplikasi Anda. Konfigurasikan lingkungan produksi untuk log pada tingkat yang sesuai untuk menghindari logging yang berlebihan.
- Aktifkan pencatatan kontainer: Konfigurasikan kontainer Anda untuk masuk `stdout` dan`stderr`. Hal ini memungkinkan Kubernetes untuk menangkap dan meneruskan log ini ke solusi logging pilihan Anda.

- Aktifkan pencatatan aplikasi: Konfigurasikan aplikasi untuk menulis log ke `stdout` dan `stderr` alih-alih menulis ke file log. Ini mengikuti metodologi aplikasi 12 faktor dan selaras dengan praktik terbaik cloud-native.
- Gunakan Kubernetes DaemonSets untuk pengumpulan log: Menerapkan agen pengumpulan log (seperti Fluent Bit) DaemonSets untuk memastikan bahwa mereka berjalan di setiap node di klaster Anda.
- Menerapkan kebijakan retensi: Menentukan dan menegakkan kebijakan penyimpanan log untuk mematuhi peraturan dan mengelola biaya penyimpanan.
- Data log aman: Enkripsi log saat transit dan saat istirahat. Menerapkan kontrol akses untuk membatasi siapa yang dapat melihat dan mengelola log.
- Memantau konsumsi log: Siapkan peringatan untuk kegagalan atau penundaan konsumsi log untuk memastikan pencatatan berkelanjutan.
- Gunakan anotasi dan label Kubernetes: Gunakan anotasi dan label Kubernetes untuk menambahkan metadata ke log Anda, untuk meningkatkan kemampuan pencarian dan pemfilteran.
- Menerapkan penelusuran terdistribusi: Gunakan alat penelusuran terdistribusi seperti AWS X-Rayatau Jaeger untuk menghubungkan log di seluruh layanan mikro.
- Optimalkan volume log: Selektif tentang apa yang Anda log untuk menghindari biaya yang tidak perlu dan masalah kinerja. Gunakan sampling untuk log bervolume tinggi dan bernilai rendah.
- Terapkan agregasi log: Gunakan alat seperti Logstash untuk mengumpulkan log dari berbagai sumber sebelum mengirimnya ke sistem logging pusat Anda.
- Gunakan Layanan AWS bila memungkinkan: Layanan seperti CloudWatch Log dan Wawasan Kontainer menyediakan integrasi tanpa batas dengan yang lain. Layanan AWS
- Terapkan analisis dan visualisasi log: Gunakan alat seperti Wawasan CloudWatch Log, Elasticsearch dengan Kibana, atau solusi pihak ketiga untuk analisis dan visualisasi log.
- Terapkan analisis log otomatis: Gunakan pembelajaran mesin dan alat yang didukung AI untuk mendeteksi anomali dan pola di log Anda secara otomatis.
- Dokumentasikan strategi logging Anda: Pertahankan dokumentasi yang jelas tentang arsitektur, praktik, dan alat logging Anda untuk tim Anda.

Pertimbangan penting untuk masuk ke Amazon EKS

Bagian ini membahas pertimbangan penting yang perlu diingat saat Anda menerapkan logging di Amazon EKS.

- Dampak kinerja: Pencatatan yang berlebihan dapat memengaruhi kinerja aplikasi. Perhatikan volume dan frekuensi log yang dihasilkan.
- Manajemen biaya: Penyimpanan dan pemrosesan log dapat menimbulkan biaya yang signifikan, terutama dalam skala besar. Menerapkan kebijakan penyimpanan log dan pertimbangkan untuk menggunakan agregasi log untuk mengurangi biaya.
- Keamanan dan kepatuhan: Pastikan log tidak berisi informasi sensitif seperti kata sandi atau data pribadi. Menerapkan enkripsi untuk log dalam perjalanan dan saat istirahat. Pertimbangkan persyaratan kepatuhan seperti Peraturan Perlindungan Data Umum (GDPR) atau Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA) saat Anda menangani log.
- Skalabilitas: Pastikan solusi logging Anda dapat menskalakan dengan ukuran cluster dan volume log Anda. Pertimbangkan untuk menggunakan buffering dan batching untuk transmisi log.
- Retensi log: Tentukan dan terapkan periode retensi log yang sesuai. Menyeimbangkan persyaratan kepatuhan terhadap biaya penyimpanan.
- Kontrol akses: Menerapkan peran dan kebijakan yang tepat AWS Identity and Access Management (IAM) untuk akses log. Ikuti prinsip hak istimewa paling sedikit untuk manajemen log.
- Konsistensi log: Gunakan format log yang konsisten di berbagai aplikasi dan layanan. Gunakan logging terstruktur untuk penguraian dan analisis yang lebih mudah.
- Sinkronisasi waktu: Sinkronisasi waktu di semua node untuk mendapatkan stempel waktu yang konsisten di log.
- Alokasi sumber daya: Alokasikan sumber daya yang sesuai (seperti CPU dan memori) untuk agen logging. Pantau penggunaan sumber daya komponen logging.
- Pertimbangan Fargate: Fargate memiliki mekanisme logging spesifik yang berbeda dari node berbasis EC2. Memahami keterbatasan dan kemampuan logging Fargate.
- Cluster multi-penyewa: Di lingkungan multi-penyewa, pastikan log diisolasi dengan benar di antara penyewa.
- Penguraian dan analisis log: Pertimbangkan alat dan keterampilan yang diperlukan untuk analisis log yang efektif. Menerapkan penguraian log untuk ekstraksi data terstruktur.
- Memantau sistem logging: Siapkan pemantauan untuk infrastruktur logging itu sendiri. Hasilkan peringatan untuk kegagalan sistem logging atau backlog.
- Dampak jaringan: Waspadai bandwidth jaringan yang digunakan oleh transmisi log. Pertimbangkan untuk menggunakan kompresi untuk data log.
- Peristiwa Kubernetes: Jangan mengabaikan peristiwa Kubernetes sebagai sumber informasi penting.

- Kontrol logging pesawat: Memahami implikasi dan biaya mengaktifkan logging pesawat kontrol.
- Kemampuan debugging: Pastikan bahwa solusi logging Anda memungkinkan debugging dan pemecahan masalah yang mudah.
- Integrasi dengan alat yang ada: Pertimbangkan bagaimana solusi logging Amazon EKS Anda terintegrasi dengan alat pemantauan dan peringatan yang ada.
- Pengujian: Uji pengaturan logging Anda secara teratur, terutama setelah peningkatan klaster.
- Dokumentasi: Pertahankan dokumentasi yang jelas tentang arsitektur dan praktik logging Anda.
- Latensi agregasi log: Waspadai latensi apa pun dalam agregasi log dan bagaimana hal itu dapat memengaruhi pemantauan waktu nyata.

Pemantauan di Amazon EKS

Pemantauan di Amazon EKS memberikan visibilitas kritis terhadap kesehatan, kinerja, dan keamanan beban kerja Kubernetes Anda. Tanpa pemantauan yang tepat, Anda berisiko mengalami gangguan layanan, pelanggaran keamanan, dan pemanfaatan sumber daya yang tidak efisien yang dapat memengaruhi operasi bisnis dan meningkatkan biaya. Pemantauan yang efektif memungkinkan Anda mengidentifikasi dan menyelesaikan masalah secara proaktif, mengoptimalkan penggunaan sumber daya, dan mempertahankan persyaratan kepatuhan di seluruh aplikasi kontainer Anda. Dengan menerapkan solusi pemantauan komprehensif, Anda dapat memastikan ketersediaan tinggi, mendeteksi anomali lebih awal, dan membuat keputusan berdasarkan data untuk penskalaan dan peningkatan infrastruktur Amazon EKS Anda.

Bagian ini mengeksplorasi berbagai aspek pemantauan Amazon EKS, termasuk berbagai jenis pemantauan, alat yang tersedia, dan praktik terbaik untuk membantu Anda membangun strategi pemantauan yang kuat untuk lingkungan Kubernetes Anda.

Di bagian ini:

- Jenis pemantauan di Amazon EKS
- Alat pemantauan untuk Amazon EKS
- Menerapkan ketersediaan tinggi untuk solusi pemantauan Amazon EKS
- Praktik terbaik untuk pemantauan di Amazon EKS
- Pertimbangan pemantauan lanjutan di Amazon EKS

Jenis pemantauan di Amazon EKS

Observabilitas yang efektif di Amazon EKS melibatkan aktivitas pemantauan infrastruktur, aplikasi, dan keamanan.

Pemantauan Infrastruktur

Pemantauan infrastruktur adalah komponen fundamental dari observabilitas Amazon EKS yang memberikan wawasan mendalam tentang kesehatan dan kinerja elemen dasar klaster Kubernetes Anda. Pada intinya, ini melibatkan pelacakan tanda-tanda vital dari komponen bidang kontrol dan node pekerja, dan memastikan bahwa platform yang mendasarinya tetap stabil dan efisien.

- Pemantauan bidang kontrol sangat penting karena mengawasi komponen utama seperti server API, database etcd, dan penjadwal. Dengan memantau latensi server API, Anda dapat dengan cepat mengidentifikasi hambatan kinerja yang mungkin memengaruhi penerapan aplikasi atau operasi penskalaan. Pemantauan kinerja Etcd memvalidasi bahwa database status cluster beroperasi secara efisien dan mencegah masalah konsistensi data yang dapat berdampak pada seluruh cluster.
- Pemantauan tingkat simpul sama pentingnya karena berfokus pada sumber daya komputasi yang menjalankan beban kerja kontainer Anda. Ini termasuk melacak pemanfaatan CPU, konsumsi memori, disk I/O, dan kinerja jaringan di semua node pekerja. Memahami metrik ini membantu mencegah kehabisan sumber daya, mengoptimalkan keputusan penskalaan node, dan memastikan perencanaan kapasitas yang tepat.
- Pemantauan jaringan memainkan peran penting dalam menjaga komunikasi yang andal antara pod, layanan, dan sumber daya eksternal. Dengan memantau throughput jaringan, latensi, dan status koneksi, Anda dapat mengidentifikasi masalah konektivitas lebih awal dan memastikan komunikasi aplikasi yang lancar. Pemantauan penyimpanan melengkapi pemantauan jaringan dengan melacak kinerja volume, pemanfaatan kapasitas, dan I/O pola, untuk membantu mencegah kemacetan terkait data.

Pemantauan infrastruktur berfungsi sebagai sistem peringatan dini untuk masalah potensial, memungkinkan pemeliharaan proaktif, dan memastikan alokasi sumber daya yang optimal. Tanpa pemantauan infrastruktur yang kuat, Anda berisiko mengalami downtime yang tidak terduga, kinerja yang menurun, dan penggunaan sumber daya yang tidak efisien yang dapat berdampak signifikan pada operasi dan biaya bisnis.

Pemantauan aplikasi

Pemantauan aplikasi sangat penting untuk menjaga aplikasi kontainer yang sehat, berkinerja, dan andal di lingkungan Amazon EKS Anda. Tingkat pemantauan ini berfokus pada beban kerja aktual yang berjalan di dalam klaster Anda dan memberikan wawasan penting tentang bagaimana aplikasi Anda berperilaku, berkinerja, dan berinteraksi dengan layanan lain.

Pemantauan aplikasi meliputi pemantauan tingkat kontainer, pemantauan tingkat layanan, dan penelusuran terdistribusi.

- Pada tingkat kontainer, pemantauan aplikasi melacak metrik penting seperti status kesehatan kontainer, jumlah restart, dan pola konsumsi sumber daya. Metrik ini membantu Anda mengidentifikasi wadah bermasalah yang mungkin menghabiskan sumber daya berlebihan

atau sering mengalami restart, yang dapat menunjukkan masalah mendasar seperti kebocoran memori atau masalah konfigurasi. Dengan memantau peristiwa siklus hidup kontainer, Anda dapat memastikan perilaku aplikasi yang tepat dan memecahkan masalah penerapan dengan cepat.

- Pemantauan tingkat layanan memberikan visibilitas ke metrik kinerja dan keandalan aplikasi seperti waktu respons, tingkat kesalahan, dan throughput permintaan. Metrik ini sangat penting untuk mempertahankan tujuan tingkat layanan (SLOs) dan memastikan pengalaman pengguna akhir yang positif. Anda dapat melacak latensi di berbagai titik akhir layanan, mengidentifikasi kemacetan kinerja, dan memantau pola kesalahan untuk mempertahankan keandalan aplikasi.
- Penelusuran terdistribusi adalah aspek penting lain dari pemantauan aplikasi, terutama dalam arsitektur layanan mikro. Dengan menerapkan tracing, Anda dapat mengikuti permintaan saat mereka mengalir melalui layanan yang berbeda, memahami dependensi, dan mengidentifikasi hambatan kinerja. end-to-endVisibilitas ini membantu Anda mengoptimalkan interaksi layanan dan memecahkan masalah kompleks yang mencakup beberapa komponen.

Metrik aplikasi khusus memainkan peran penting dalam memberikan wawasan khusus bisnis. Ini mungkin termasuk metrik seperti tingkat pemrosesan pesanan, frekuensi login pengguna, atau tingkat keberhasilan transaksi. Anda dapat mengkorelasikan metrik kustom ini dengan metrik infrastruktur dan kontainer untuk lebih memahami bagaimana kinerja infrastruktur memengaruhi operasi bisnis dan membuat keputusan berdasarkan data untuk penskalaan dan pengoptimalan.

Pentingnya pemantauan aplikasi terletak pada kemampuannya untuk memberikan pandangan komprehensif tentang kesehatan dan kinerja aplikasi. Pemantauan ini memungkinkan Anda mempertahankan kualitas layanan yang tinggi, menyelesaikan masalah dengan cepat, dan terus mengoptimalkan aplikasi Anda untuk memenuhi tujuan bisnis.

Pemantauan keamanan

Pemantauan keamanan di Amazon EKS adalah aktivitas penting yang membantu organisasi menjaga integritas, kerahasiaan, dan kepatuhan lingkungan Kubernetes mereka. Pendekatan keamanan komprehensif ini menggabungkan pengawasan berkelanjutan, deteksi ancaman, dan pemantauan kepatuhan untuk melindungi beban kerja dalam peti kemas dari potensi risiko keamanan dan akses tidak sah. Ini mencakup pemantauan otentikasi dan otorisasi, pemantauan keamanan jaringan, dan pemantauan konfigurasi dan kepatuhan.

- Pemantauan otentikasi dan otorisasi membentuk garis pertahanan pertama dengan melacak semua upaya untuk mengakses cluster. Ini termasuk memantau permintaan server API, melacak upaya login yang berhasil dan gagal, dan mengaudit perubahan kontrol akses berbasis peran

(RBAC). Dengan memelihara log audit terperinci tentang siapa yang mengakses sumber daya dan kapan, Anda dapat dengan cepat mendeteksi potensi pelanggaran keamanan, upaya akses yang tidak sah, atau aktivitas eskalasi hak istimewa. Ini sangat penting dalam lingkungan multi-penyewa di mana mempertahankan kontrol akses yang ketat sangat penting.

- Pemantauan keamanan jaringan berfokus pada mendeteksi dan mencegah komunikasi yang tidak sah antara pod dan layanan. Dengan memantau pelanggaran kebijakan jaringan dan pola lalu lintas yang tidak biasa, Anda dapat mengidentifikasi potensi ancaman keamanan seperti upaya pelarian kontainer atau pergerakan lateral dalam cluster. Ini termasuk melacak komunikasi klaster internal dan pola lalu lintas eksternal untuk memastikan bahwa kontainer hanya berkomunikasi dengan titik akhir resmi dan mengikuti kebijakan keamanan yang ditentukan.
- Konfigurasi dan pemantauan kepatuhan sangat penting untuk menjaga garis dasar keamanan dan memenuhi persyaratan peraturan. Ini melibatkan pemindaian gambar kontainer secara terus menerus untuk mencari kerentanan, memantau keamanan runtime, dan melacak perubahan konfigurasi yang mungkin memengaruhi postur keamanan. Audit kepatuhan reguler memastikan kepatuhan terhadap standar industri dan kebijakan keamanan organisasi, dan deteksi penyimpangan konfigurasi membantu mencegah perubahan yang tidak sah yang dapat menimbulkan risiko keamanan.

Pemantauan keamanan di Amazon EKS memberikan visibilitas dan kontrol yang diperlukan untuk membantu melindungi dari ancaman keamanan modern sambil memastikan kepatuhan terhadap persyaratan peraturan. Dengan menerapkan pemantauan keamanan yang komprehensif, organisasi Anda dapat mempertahankan postur keamanan yang kuat, merespons insiden keamanan dengan cepat, dan menunjukkan kepatuhan terhadap berbagai standar peraturan.

Alat pemantauan untuk Amazon EKS

Bagian ini membahas tiga kategori alat pemantauan Amazon EKS: layanan AWS pemantauan, solusi open source atau proprietary, dan alat khusus.

AWS layanan

- Amazon CloudWatch: Layanan pemantauan dan pencatatan yang komprehensif

  CloudWatch membentuk tulang punggung solusi AWS pemantauan dan menyediakan kemampuan ekstensif untuk lingkungan Amazon EKS. Ini memberikan Wawasan Kontainer untuk kontainer granular dan metrik cluster, sehingga Anda dapat memantau kinerja, pemanfaatan sumber daya,

dan kesehatan aplikasi. Layanan ini unggul dalam agregasi dan analisis log, dan mendukung logging terpusat di seluruh kontainer dan node. CloudWatch terintegrasi secara alami dengan Layanan AWS. Ini menyediakan konfigurasi alarm otomatis dan mendukung metrik dan dasbor khusus, yang menjadikannya alat penting untuk pemantauan Amazon EKS.

- AWS X-Ray: Platform penelusuran terdistribusi tingkat lanjut

  X-Ray meningkatkan observabilitas dengan menyediakan kemampuan penelusuran terdistribusi yang canggih. Visualisasi peta layanannya menawarkan wawasan yang jelas tentang arsitektur dan dependensi aplikasi, dan pelacakan permintaan terperinci membantu mengidentifikasi kemacetan kinerja di seluruh layanan. X-Ray dapat melacak permintaan melalui arsitektur layanan mikro yang kompleks, yang membuatnya sangat berharga untuk pemecahan masalah dan pengoptimalan, terutama dalam sistem terdistribusi yang menjangkau banyak. Layanan AWS

- AWS Distro untuk OpenTelemetry: Kerangka observabilitas terpadu

  Distro untuk OpenTelemetry menyediakan kemampuan pengumpulan data terpadu dengan dukungan lintas platform, yang membuatnya ideal untuk lingkungan hybrid. Layanan ini terintegrasi dengan yang lain Layanan AWS, mendukung instrumentasi khusus, dan menawarkan fleksibilitas dalam menerapkan solusi pemantauan komprehensif sambil mempertahankan kompatibilitas dengan standar industri.

- Grafana yang Dikelola Amazon: Visualisasi tingkat perusahaan

  Grafana yang Dikelola Amazon menyediakan layanan yang dikelola sepenuhnya untuk visualisasi dan analitik data. Ini menawarkan integrasi tanpa batas dengan fitur keamanan bawaan lainnya Layanan AWS, dan skalabilitas tingkat perusahaan. Layanan ini menyederhanakan pembuatan dan manajemen dasbor sambil menyediakan fitur-fitur canggih seperti akses sumber data lintas akun dan integrasi dengan. AWS IAM Identity Center

- Layanan Terkelola Amazon untuk Prometheus: Pemantauan yang sangat tersedia, aman, dan terkelola

  Layanan Terkelola Amazon untuk Prometheus adalah layanan pemantauan yang kompatibel dengan Prometheus yang dikelola sepenuhnya. Ini menyediakan penskalaan otomatis, ketersediaan tinggi, dan penyerapan dan kueri metrik yang aman. Layanan ini terintegrasi secara mulus dengan Amazon EKS dan menghilangkan biaya operasional pengelolaan server Prometheus.

Solusi open source atau proprietary

AWS Alat yang dijelaskan di bagian sebelumnya menawarkan integrasi yang mulus dan layanan terkelola. Alat open source yang tercantum di bagian ini melengkapi Layanan AWS dengan memberikan fleksibilitas dan opsi penyesuaian yang luas. Memahami kemampuan dan kasus penggunaan setiap alat membantu Anda merancang strategi pemantauan yang paling memenuhi persyaratan spesifik Anda.

- Prometheus: Toolkit pengumpulan metrik

  Prometheus adalah solusi open source untuk pengumpulan metrik di lingkungan Kubernetes. Database deret waktu dan bahasa kueri PromQL memungkinkan analisis metrik yang canggih. Kemampuan penemuan layanan platform secara otomatis beradaptasi dengan lingkungan Kubernetes yang dinamis, dan sistem manajemen peringatannya memberi Anda informasi tentang masalah kritis. Prometheus menyediakan opsi integrasi yang luas, yang menjadikannya pilihan serbaguna untuk pemantauan metrik yang komprehensif.

- Grafana: Mesin visualisasi tingkat lanjut

  Grafana mengubah data pemantauan yang kompleks menjadi wawasan yang dapat ditindaklanjuti melalui kemampuan visualisasinya. Platform ini membuat dasbor khusus yang menggabungkan data dari berbagai sumber dan memberikan tampilan terpadu infrastruktur dan metrik aplikasi. Dukungannya untuk berbagai sumber data dan fitur manajemen peringatan memberikan pemantauan yang komprehensif. Grafana dapat membantu Anda memvisualisasikan data real-time dan historis, sehingga Anda dapat mengidentifikasi tren dan membuat keputusan yang tepat.

- Fluent Bit: Lapisan logging terpadu

  Solusi logging ini menyediakan pengumpulan dan pengelolaan log untuk lingkungan Kubernetes. Integrasi asli Kubernetes memastikan pengumpulan log yang mulus dari kontainer dan node, dan dukungannya untuk beberapa tujuan keluaran menawarkan fleksibilitas dalam penyimpanan dan analisis log. Fitur lanjutan seperti penguraian log dan pemfilteran memungkinkan Anda memproses dan merutekan log berdasarkan persyaratan tertentu. Sifat ringan dari Fluent Bit membuatnya sangat cocok untuk lingkungan kontainer.

- Datadog: Observabilitas tumpukan penuh

  Datadog menyediakan kemampuan pemantauan komprehensif dengan dukungan Kubernetes asli. Ini menawarkan pemantauan infrastruktur, pemantauan kinerja aplikasi (APM), manajemen log, dan analitik real-time. Anda dapat menggunakan penemuan layanan otomatis platform dan katalog

integrasi ekstensif untuk pemantauan Amazon EKS, dan kemampuan pembelajaran mesinnya untuk mendeteksi anomali dan memprediksi potensi masalah.

- Relik Baru: Pemantauan kinerja aplikasi

  New Relic menawarkan visibilitas ke kinerja aplikasi dan kesehatan infrastruktur. Integrasi Kubernetes menyediakan wawasan kontainer yang mendetail, penelusuran terdistribusi, dan dasbor khusus. Platform ini membantu Anda menghubungkan kinerja aplikasi dengan metrik infrastruktur, sehingga Anda dapat dengan cepat mengidentifikasi dan menyelesaikan masalah.

- Elastic Stack (ELK Stack): Analisis log dan pencarian

  ELK Stack menggabungkan Elasticsearch, Logstash, dan Kibana untuk menyediakan manajemen log dan kemampuan analisis. Ini menawarkan fungsionalitas pencarian lanjutan, alat visualisasi, dan fitur pembelajaran mesin. Anda dapat menggunakan tumpukan untuk menangani volume besar data log dari lingkungan Amazon EKS Anda.

Alat khusus

Anda dapat mencampur dan mencocokkan alat-alat berikut berdasarkan persyaratan pemantauan spesifik Anda, skala operasi, dan preferensi organisasi. Kuncinya adalah membuat tumpukan pemantauan yang memberikan visibilitas komprehensif sambil tetap dapat dikelola dan hemat biaya.

- kube-state-metrics (KSM): Pemantauan status Kubernetes

  Layanan add-on ini mendengarkan server API Kubernetes dan menghasilkan metrik tentang status objek. Ini memberikan wawasan tentang kesehatan penerapan, pod, dan sumber daya Kubernetes lainnya.

- Kubernetes Metrics Server: Metrik sumber daya

  Server metrik ini mengumpulkan metrik sumber daya dari kubelet dan mengeksposnya melalui API metrik Kubernetes. Ini menyediakan penskalaan otomatis pod horizontal dan CPU dasar dan metrik memori.

- Kubecost: Pemantauan biaya Kubernetes

  Alat seperti Kubecost memberikan analisis biaya rinci dan rekomendasi optimasi untuk kluster EKS. Mereka membantu Anda memahami dan mengoptimalkan pengeluaran cloud di berbagai ruang nama, penerapan, dan layanan.

Menerapkan ketersediaan tinggi untuk solusi pemantauan Amazon EKS

Strategi ketersediaan tinggi (HA) yang kuat untuk pemantauan Amazon EKS sangat penting untuk memastikan visibilitas berkelanjutan ke lingkungan Kubernetes Anda. Bagian ini membahas pendekatan komprehensif untuk menerapkan HA di berbagai aspek infrastruktur pemantauan Anda.

Redundansi dan skalabilitas arsitektur

Membangun sistem pemantauan yang sangat tersedia dimulai dengan desain arsitektur yang tepat. Komponen pemantauan harus didistribusikan di beberapa AWS Availability Zone untuk melindungi dari kegagalan zona. Ini termasuk menerapkan penskalaan horizontal untuk komponen pemantauan penting seperti server Prometheus, pengumpul log, dan manajer peringatan. Anda dapat menggunakan layanan AWS terkelola seperti Amazon Managed Service for Prometheus dan Amazon Managed Grafana untuk membantu mengurangi overhead operasional sekaligus memastikan ketersediaan yang tinggi. Konfigurasikan mekanisme failover otomatis untuk menjaga kontinuitas layanan selama kegagalan komponen, dengan pemeriksaan kesehatan dan prosedur pemulihan otomatis.

Strategi penyimpanan data yang tangguh

Ketahanan penyimpanan data sangat penting untuk menjaga keandalan sistem pemantauan. Menerapkan solusi penyimpanan terdistribusi memastikan bahwa data metrik dan log tetap dapat diakses bahkan jika node penyimpanan individu gagal. Ini termasuk mengonfigurasi replikasi data yang tepat di beberapa Availability Zone dan menggunakan backend penyimpanan yang berbeda untuk redundansi. Tetapkan prosedur pencadangan reguler untuk data historis, dengan proses pemulihan terdokumentasi untuk berbagai skenario kegagalan. Untuk database deret waktu seperti Prometheus, menerapkan solusi penyimpanan jarak jauh membantu memisahkan masalah penyimpanan dari pengumpulan data dan meningkatkan keandalan sistem secara keseluruhan.

Manajemen peringatan redundan

Manajemen peringatan memerlukan perhatian khusus dalam pengaturan HA. Menerapkan pengelola peringatan yang berlebihan memastikan bahwa pemberitahuan kritis mencapai penerima yang dituju bahkan selama kegagalan sistem. Konfigurasikan beberapa saluran notifikasi seperti email, SMS, Slack, dan PagerDuty untuk menyediakan jalur komunikasi alternatif. Gunakan mekanisme deduplikasi peringatan untuk mencegah badai peringatan selama kegagalan sistem sebagian, dan metode pemberitahuan mundur untuk memastikan bahwa peringatan kritis tidak pernah terlewatkan.

Menerapkan korelasi peringatan membantu mempertahankan konteks selama skenario failover dan mencegah pemberitahuan duplikat dari sistem yang berlebihan.

Load balancing dan penemuan layanan

Penyeimbangan beban yang tepat sangat penting untuk menjaga layanan pemantauan yang stabil. AWS Application Load Balancer mendistribusikan lalu lintas pemantauan masuk di beberapa titik akhir, dan pemeriksaan kesehatan memastikan bahwa lalu lintas hanya diarahkan ke instans yang sehat. Mekanisme penemuan layanan membantu komponen pemantauan secara otomatis beradaptasi dengan perubahan lingkungan, seperti penambahan node atau layanan baru. Terapkan agen pemantauan secara konsisten di semua node dengan menggunakan DaemonSets untuk memastikan cakupan komprehensif saat skala klaster.

Pertimbangan HA tambahan

Ketahanan jaringan:

- Menerapkan jalur jaringan yang berlebihan.
- Konfigurasikan desain subnet yang tepat di seluruh Availability Zone.
- Gunakan AWS Direct Connectdengan rute cadangan.
- Konfigurasikan grup keamanan yang sesuai dan daftar kontrol akses jaringan (jaringan ACLs).

Memantau monitor:

- Menyebarkan sistem pemantauan sekunder.
- Menerapkan pemantauan lintas wilayah.
- Konfigurasikan peringatan untuk sistem yang tidak responsif.
- Uji prosedur failover secara teratur.

Perencanaan kapasitas:

- Pantau tren penggunaan sumber daya.
- Menerapkan penskalaan prediktif.
- Uji kinerja secara teratur.

Manajemen data:

- Menerapkan kebijakan retensi data.
- Konfigurasikan agregasi metrik.
- Merencanakan manajemen siklus hidup data.
- Optimalkan penyimpanan secara teratur.

Prosedur pemulihan:

- Proses pemulihan dokumen.
- Uji pemulihan bencana secara teratur.
- Terapkan pemulihan otomatis jika memungkinkan.
- Identifikasi dan terapkan jalur eskalasi yang jelas.

Dengan menerapkan praktik ketersediaan tinggi ini, Anda dapat memastikan bahwa infrastruktur pemantauan Amazon EKS Anda tetap andal dan tangguh, dan Anda memiliki visibilitas berkelanjutan ke lingkungan Kubernetes Anda bahkan selama berbagai skenario kegagalan. Pengujian dan pembaruan rutin untuk konfigurasi HA ini memastikan bahwa konfigurasi tersebut tetap efektif seiring berkembangnya lingkungan.

Praktik terbaik untuk pemantauan di Amazon EKS

Pendekatan implementasi strategis

Strategi pemantauan Amazon EKS yang sukses dimulai dengan pendekatan implementasi bertahap yang terencana dengan baik.

- Mulailah dengan mengidentifikasi dan memantau metrik penting yang secara langsung memengaruhi operasi bisnis dan keandalan aplikasi Anda. Fondasi ini harus mencakup metrik infrastruktur penting, indikator kinerja aplikasi utama, dan metrik keamanan kritis. Secara bertahap memperluas cakupan pemantauan berdasarkan kebutuhan operasional dan pelajaran yang dipetik, dan pastikan bahwa setiap penambahan memberikan nilai yang berarti.
- Menerapkan proses penerapan otomatis dengan menggunakan alat infrastruktur sebagai kode (IAc) seperti Terraform atau CloudFormation untuk memastikan konsistensi dan pengulangan.
- Uji dan validasi sistem pemantauan untuk membantu menjaga keandalan dan akurasi.
- Perbaiki parameter pemantauan secara terus menerus sesuai dengan kebutuhan bisnis yang terus berkembang.

Manajemen data yang efektif

Manajemen data yang tepat sangat penting untuk mempertahankan solusi pemantauan yang efisien dan hemat biaya.

- Menerapkan kebijakan penyimpanan data yang jelas yang menyeimbangkan kebutuhan analisis historis dengan biaya penyimpanan.
- Konfigurasikan laju pengambilan sampel yang sesuai untuk berbagai jenis metrik: frekuensi yang lebih tinggi untuk metrik kritis dan frekuensi yang lebih rendah untuk yang kurang kritis.
- Gunakan agregasi metrik untuk mengurangi volume data sambil mempertahankan wawasan yang bermakna, terutama untuk analisis tren jangka panjang.
- Menerapkan penyimpanan log sistematis dan prosedur arsip untuk sistem logging terpusat (seperti CloudWatch Log) untuk mengelola biaya penyimpanan dan menjaga akses ke data penting tetap dapat diakses.

> **Note**
>
> Rotasi log tingkat kontainer ditangani secara otomatis oleh kubelet di Amazon EKS versi 1.21 atau yang lebih baru.

- Pertimbangkan untuk menerapkan hot-warm-cold arsitektur penyimpanan log untuk mengoptimalkan kecepatan akses dan efisiensi biaya.

Konfigurasi dan manajemen peringatan

Konfigurasi peringatan memerlukan pertimbangan yang cermat untuk mempertahankan efektivitas tanpa menyebabkan kelelahan waspada.

- Tentukan ambang batas yang jelas dan dapat ditindaklanjuti berdasarkan tujuan tingkat layanan (SLOs) dan pola kinerja historis.
- Menerapkan sistem tingkat keparahan peringatan berjenjang yang secara jelas membedakan antara masalah kritis yang memerlukan perhatian segera dan hal-hal yang kurang mendesak.
- Pastikan bahwa peringatan memberikan konteks yang cukup dan informasi yang dapat ditindaklanjuti untuk memfasilitasi penyelesaian masalah dengan cepat.
- Tetapkan prosedur eskalasi yang jelas dengan kepemilikan dan waktu respons yang ditentukan untuk tingkat keparahan peringatan yang berbeda.

- Tinjau dan perbaiki konfigurasi peringatan secara teratur untuk membantu menjaga relevansi dan efektivitasnya.

Optimalisasi sumber daya

Pemantauan berkelanjutan pemanfaatan sumber daya sangat penting untuk mempertahankan operasi yang hemat biaya.

- Menerapkan pemantauan sumber daya yang komprehensif di semua komponen klaster, termasuk node, pod, dan volume persisten.
- Konfigurasikan penskalaan otomatis berdasarkan pola penggunaan aktual dan persyaratan kinerja untuk memastikan pemanfaatan sumber daya yang efisien sekaligus mempertahankan kinerja.
- Gunakan tag alokasi biaya untuk melacak konsumsi sumber daya oleh tim, aplikasi, atau lingkungan yang berbeda.
- Analisis metrik efisiensi sumber daya secara teratur untuk mengidentifikasi peluang pengoptimalan dan menerapkan peningkatan.
- Pertimbangkan untuk menerapkan alat manajemen biaya untuk melacak dan mengoptimalkan pengeluaran cloud.

Keamanan

Pertimbangan keamanan harus menjadi bagian integral dari strategi pemantauan Anda.

- Menerapkan prinsip akses hak istimewa terkecil untuk semua komponen pemantauan untuk memastikan bahwa pengguna dan layanan hanya memiliki izin yang mereka butuhkan.
- Aktifkan pencatatan audit komprehensif untuk melacak semua akses dan perubahan pada sistem pemantauan.
- Melakukan tinjauan keamanan reguler terhadap konfigurasi pemantauan dan pola akses untuk mengidentifikasi potensi kerentanan.
- Menerapkan enkripsi untuk data pemantauan sensitif baik dalam perjalanan maupun saat istirahat.
- Integrasikan pemantauan keamanan dengan sistem informasi keamanan dan manajemen acara (SIEM) yang ada untuk visibilitas keamanan yang komprehensif.

Pertimbangan pemantauan lanjutan di Amazon EKS

Optimalisasi kinerja:

- Optimalkan interval pengumpulan metrik.
- Konfigurasikan pola kueri yang efisien.
- Menerapkan pra-agregasi metrik.
- Gunakan solusi penyimpanan yang tepat.

Kepatuhan dan tata kelola:

- Pertahankan jejak audit.
- Menerapkan pemantauan kepatuhan.
- Berikan pelaporan kepatuhan secara teratur.
- Prosedur pemantauan dokumen.

Pemulihan bencana:

- Cadangkan konfigurasi pemantauan secara teratur.
- Prosedur pemulihan dokumen.
- Uji proses pemulihan.

Perbaikan berkelanjutan:

- Pantau sesi peninjauan secara teratur.
- Optimalkan siklus kinerja.
- Perbarui pemantauan berdasarkan insiden.
- Menggabungkan umpan balik pengguna.

Praktik terbaik ini menyediakan kerangka kerja untuk menerapkan dan memelihara solusi pemantauan yang efektif untuk lingkungan Amazon EKS. Tinjau dan perbarui praktik ini secara teratur agar tetap selaras dengan kebutuhan organisasi dan standar industri Anda. Pemantauan bukanlah pengaturan satu kali — ini adalah proses berkelanjutan yang membutuhkan perhatian dan penyempurnaan secara teratur.

Menelusuri di Amazon EKS

Penelusuran adalah komponen penting dari observabilitas aplikasi di Amazon EKS. Penelusuran memberikan visibilitas terperinci ke dalam alur permintaan dan interaksi layanan dengan mengumpulkan, memproses, dan memvisualisasikan jalur permintaan saat mereka melakukan perjalanan melalui berbagai layanan mikro yang digunakan pada kluster EKS. Kemampuan ini membantu Anda memahami perilaku sistem, mengidentifikasi kemacetan, dan memecahkan masalah secara efektif di lingkungan Amazon EKS Anda. Penelusuran yang efektif menghilangkan kompleksitas debugging sistem terdistribusi dengan memberikan end-to-end visibilitas ke dalam alur permintaan. Hal ini memungkinkan untuk melacak transaksi lintas batas layanan dan mengidentifikasi masalah kinerja atau kegagalan dalam beban kerja Amazon EKS.

Implementasi penelusuran keseluruhan di Amazon EKS memungkinkan Anda memahami perilaku sistem, mengoptimalkan kinerja, dan mempertahankan keandalan aplikasi kontainer Anda. Pada akhirnya, kemampuan penelusuran meningkatkan visibilitas operasional dan pemeliharaan sistem di lingkungan Amazon EKS.

AWS X-Ray memainkan peran penting dalam melacak data tentang aplikasi Anda. Penelusuran melibatkan pemantauan berbagai aspek interaksi layanan, termasuk yang berikut:

- Jalur permintaan dan dependensi memberikan wawasan penting tentang perilaku sistem terdistribusi Anda. Mereka melacak perjalanan lengkap permintaan saat mereka melintasi berbagai layanan mikro dan komponen. Memetakan dependensi layanan membantu Anda memahami pola komunikasi dan mengidentifikasi jalur penting dalam arsitektur aplikasi Anda. Untuk detail implementasi, lihat Menggunakan peta jejak AWS X-Ray layanan dalam dokumentasi X-Ray.
- Latensi dan kemacetan layanan adalah metrik penting untuk mempertahankan kinerja sistem yang optimal. Dengan mengukur dan menganalisis waktu respons antar layanan, Anda dapat mengidentifikasi masalah kinerja secara efektif. Data ini memungkinkan Anda untuk menentukan layanan atau operasi tertentu yang menyebabkan penundaan dalam rantai permintaan dan memungkinkan upaya pengoptimalan yang ditargetkan. Untuk mempelajari lebih lanjut tentang analisis latensi, lihat Berinteraksi dengan konsol Analytics di dokumentasi X-Ray.
- Pola propagasi kesalahan membantu Anda memahami keandalan sistem dan toleransi kesalahan. Dengan memahami bagaimana kegagalan mengalir melalui sistem dengan melacak jalur kesalahan di seluruh layanan, Anda dapat merancang aplikasi Anda dengan lebih baik. Visibilitas ini membantu Anda mengidentifikasi akar penyebab kesalahan dan dampaknya pada layanan dependen, yang mengarah ke sistem yang lebih tangguh. Untuk detail implementasi, lihat Jejak dalam dokumentasi X-Ray.

- Pemanfaatan sumber daya di seluruh layanan memberikan wawasan tentang efisiensi sistem dan optimalisasi biaya. Anda dapat memantau CPU, memori, dan pola penggunaan jaringan yang berkorelasi dengan data jejak untuk memahami permintaan sumber daya. Data ini membantu Anda menganalisis tren konsumsi sumber daya untuk mengoptimalkan kinerja dan biaya layanan di seluruh klaster EKS Anda. Untuk penyiapan pemantauan, lihat Memantau performa klaster Anda dan melihat log di dokumentasi Amazon EKS.
- Alur transaksi pengguna akhir sangat penting untuk memahami dan meningkatkan pengalaman pengguna. Dengan melacak interaksi pengguna lengkap dari frontend ke layanan backend, Anda dapat memastikan kinerja aplikasi yang optimal. Anda dapat mengukur dan mengoptimalkan waktu end-to-end respons untuk perjalanan pengguna yang kritis, yang secara langsung memengaruhi kepuasan pelanggan. Untuk menerapkan pemantauan pengguna akhir, gunakan AWS X-Ray SDK untuk bahasa pemrograman Anda.
- Interaksi API gateway membentuk garis depan kinerja dan keamanan aplikasi Anda. Anda dapat memantau pola permintaan dan kinerja di titik masuk API untuk memastikan pengiriman layanan yang optimal. Visibilitas ini membantu Anda melacak autentikasi, otorisasi, dan dampak pembatasan tarif pada alur permintaan, untuk menjaga persyaratan keamanan dan kinerja. Pelajari selengkapnya tentang penelusuran API di Amazon API Gateway dengan dokumentasi X-Ray.

Penelusuran efektif di Amazon EKS melampaui pengumpulan rentang dan jejak. Ini membutuhkan strategi yang terstruktur dengan baik yang menyeimbangkan kebutuhan observabilitas dengan kinerja sistem. Strategi ini harus fokus pada:

- Menerapkan tarif pengambilan sampel yang sesuai: Konfigurasikan aturan pengambilan sampel berdasarkan pola lalu lintas dan prioritas bisnis untuk mengoptimalkan biaya sambil mempertahankan visibilitas transaksi penting. Untuk mempelajari lebih lanjut, lihat Mengonfigurasi aturan pengambilan sampel dalam dokumentasi X-Ray.
- Mendefinisikan jalur dan layanan penting untuk dilacak: Mengidentifikasi dan memprioritaskan layanan penting dan perjalanan pengguna yang memerlukan penelusuran terperinci untuk memastikan pemantauan kinerja yang optimal. Untuk informasi selengkapnya, lihat Mengirim metrik dan melacak data dengan Operator ADOT di dokumentasi Amazon EKS.
- Menetapkan kebijakan retensi data yang tepat: Siapkan aturan manajemen siklus hidup data untuk menyeimbangkan kebutuhan observabilitas dengan biaya penyimpanan dan persyaratan kepatuhan. Untuk melihat kebijakan CloudWatch penyimpanan, lihat Bekerja dengan grup log dan aliran log di dokumentasi CloudWatch Log.

- Menyiapkan alat visualisasi dan analisis yang efektif: Terapkan dan konfigurasikan alat visualisasi seperti konsol AWS X-Ray Analytics atau Grafana yang Dikelola Amazon untuk menganalisis data jejak secara efektif. Untuk informasi selengkapnya, lihat Berinteraksi dengan konsol Analytics di dokumentasi X-Ray.

Di bagian ini:

- Alat penelusuran untuk Amazon EKS
- Praktik terbaik untuk melacak di Amazon EKS

Alat penelusuran untuk Amazon EKS

Amazon EKS mendukung beberapa AWS opsi pihak ketiga untuk menerapkan penelusuran terdistribusi.

Layanan AWS

- AWS X-Ray: Platform penelusuran terdistribusi tingkat lanjut

  X-Ray sepenuhnya dikelola Layanan AWS yang menyediakan kemampuan end-to-end penelusuran. Ini secara otomatis instrumen Layanan AWS dan menyediakan peta layanan terperinci dan analitik untuk aplikasi Anda yang berjalan di Amazon EKS. X-Ray terintegrasi dengan yang lain Layanan AWS CloudWatch, termasuk Amazon, dan menawarkan korelasi otomatis jejak dengan Layanan AWS panggilan.
- AWS Distro untuk OpenTelemetry: Kerangka observabilitas terpadu

  Distro for OpenTelemetry adalah distribusi yang aman, siap produksi, dan AWS didukung untuk aplikasi cloud-native. OpenTelemetry Ini menawarkan kemampuan instrumentasi vendor-netral sambil mempertahankan Layanan AWS integrasi asli, yang membuatnya ideal untuk lingkungan cloud hybrid. Distro untuk OpenTelemetry mendukung beberapa backend observabilitas dan menyediakan integrasi tanpa batas dengan layanan pemantauan. AWS

Solusi sumber terbuka

- OpenTelemetry: Kerangka observabilitas sumber terbuka

OpenTelemetry menyediakan kerangka observabilitas standar dengan pustaka instrumentasi komprehensif yang mendukung beberapa bahasa pemrograman. Opsi backend yang fleksibel dan pendekatan netral vendor membuatnya ideal untuk beban kerja yang membutuhkan konsistensi di berbagai lingkungan. Ekosistem kerangka kerja yang luas memastikan kompatibilitas yang luas dengan berbagai solusi pemantauan.

- Jaeger: Platform penelusuran terdistribusi sumber terbuka

  Jaeger menawarkan kemampuan penelusuran yang komprehensif dengan propagasi konteks terdistribusi waktu nyata. Ini memberikan analisis akar penyebab dan optimalisasi kinerja melalui visualisasi ketergantungan layanan terperinci. Arsitektur Jaeger dirancang untuk skalabilitas tinggi dan mendukung berbagai backend penyimpanan, yang membuatnya cocok untuk penyebaran Amazon EKS skala besar. Lihat Jaeger untuk pengaturan EKS
- Grafana Tempo: Penelusuran terdistribusi

  Tempo adalah solusi Grafana Labs yang menyediakan penyimpanan jejak skala tinggi dan integrasi tanpa batas dengan metrik Prometheus. Model retensi jejak yang hemat biaya dan integrasi asli dengan Grafana membuatnya cocok untuk organisasi yang sudah menggunakan Grafana untuk visualisasi. Arsitektur Tempo dirancang khusus untuk lingkungan cloud-native seperti Amazon EKS.

Praktik terbaik untuk melacak di Amazon EKS

Bagian ini menyediakan daftar lengkap praktik dan teknik terbaik untuk membuat sistem penelusuran efektif yang meningkatkan observabilitas dan pemecahan masalah untuk aplikasi berbasis Kubernetes Anda di Amazon EKS.

- Pengambilan sampel strategis: Konfigurasikan tingkat pengambilan sampel yang berbeda berdasarkan pola lalu lintas aplikasi Anda dan pentingnya layanan yang Anda gunakan. Menerapkan laju pengambilan sampel yang lebih tinggi untuk jalur kritis sekaligus mengurangi pengambilan sampel untuk rute volume tinggi dan kurang kritis untuk mengoptimalkan biaya. Untuk panduan, lihat Mengonfigurasi aturan pengambilan sampel dalam dokumentasi. AWS X-Ray
- Pengaturan instrumentasi: Gunakan alat instrumentasi otomatis seperti X-Ray SDK atau AWS Distro untuk OpenTelemetry kolektor untuk meminimalkan upaya instrumentasi manual. Pertahankan konvensi penamaan yang konsisten dan propagasi konteks di seluruh layanan untuk korelasi jejak yang lebih baik. Untuk informasi selengkapnya, lihat Distro untuk dokumentasi OpenTelemetry kolektor.

- Manajemen data: Menerapkan periode retensi dan strategi kompresi yang tepat untuk menyeimbangkan biaya penyimpanan dengan kebutuhan observabilitas Anda. Tetapkan kontrol privasi data yang jelas dan prosedur pencadangan untuk melindungi data jejak sensitif. Untuk informasi selengkapnya, lihat Mengubah penyimpanan data CloudWatch log di Log dalam dokumentasi CloudWatch Log.
- Optimalisasi kinerja: Memantau dan mengoptimalkan penelusuran overhead untuk meminimalkan dampak pada kinerja aplikasi. Gunakan buffering yang efisien dan pemrosesan asinkron untuk mengurangi dampak latensi. Untuk informasi selengkapnya, lihat Mengonfigurasi AWS X-Ray daemon dalam dokumentasi X-Ray.
- Kontrol keamanan: Menerapkan kontrol akses yang tepat dan langkah-langkah perlindungan data dengan menggunakan peran dan kebijakan IAM. Audit keamanan reguler dan tinjauan kepatuhan membantu memastikan bahwa data jejak tetap aman. Untuk informasi selengkapnya, lihat Keamanan AWS X-Ray dalam dokumentasi X-Ray.
- Pemantauan dan peringatan: Siapkan pemantauan komprehensif untuk kesehatan pengumpulan jejak dan konfigurasikan peringatan untuk masalah pengumpulan. Lacak laju pengambilan sampel dan metrik kinerja sistem untuk memastikan pengoperasian yang optimal. Untuk informasi selengkapnya, lihat Wawasan Kontainer dalam CloudWatch dokumentasi.
- Ketersediaan tinggi: Terapkan kolektor redundan di seluruh Availability Zone dan konfigurasikan mekanisme failover yang tepat. Pengujian reguler pengaturan ketersediaan tinggi memastikan pengumpulan jejak yang andal. Untuk informasi selengkapnya, lihat Menggunakan AWS Distro untuk OpenTelemetry sebagai kolektor di dokumentasi Amazon Managed Service for Prometheus.

Dengan mengikuti praktik terbaik ini, Anda dapat membuat sistem penelusuran yang kuat, efisien, dan efektif untuk lingkungan Amazon EKS Anda. Ini akan membantu memastikan observabilitas yang komprehensif, pemecahan masalah yang efisien, dan kinerja optimal aplikasi berbasis Kubernetes Anda.

Peringatan di Amazon EKS

Peringatan adalah komponen penting dalam mengelola dan memelihara aplikasi yang berjalan di Amazon EKS. Ini berfungsi sebagai sistem peringatan dini yang memberi tahu operator dan pengembang tentang potensi masalah, anomali, atau penurunan kinerja sebelum mereka meningkat menjadi masalah serius yang dapat memengaruhi ketersediaan layanan atau pengalaman pengguna. Peringatan melibatkan pemantauan berbagai aspek klaster Kubernetes, termasuk:

- Infrastruktur kesehatan
- Kinerja aplikasi
- Metrik kontainer
- Metrik bisnis khusus

Peringatan yang efektif di Amazon EKS lebih dari sekadar mengatur notifikasi. Ini membutuhkan well-thought-out strategi yang menyeimbangkan kebutuhan akan informasi tepat waktu dengan potensi kelelahan waspada. Strategi ini harus:

- Tentukan ambang batas dan kondisi yang bermakna.
- Prioritaskan peringatan berdasarkan tingkat keparahan dan dampak.
- Menerapkan prosedur routing dan eskalasi yang tepat.
- Integrasikan dengan manajemen insiden dan alat komunikasi.

Di bagian ini:

- Alat peringatan untuk Amazon EKS
- Praktik terbaik untuk peringatan di Amazon EKS

Alat peringatan untuk Amazon EKS

Amazon EKS mendukung beberapa AWS opsi pihak ketiga untuk menerapkan peringatan. Saat Anda memilih alat untuk peringatan Amazon EKS, pertimbangkan faktor-faktor seperti kemampuan integrasi, skalabilitas, kemudahan penggunaan, biaya, dan fitur spesifik yang sesuai dengan persyaratan pemantauan dan peringatan Anda. Banyak organisasi menggunakan kombinasi alat ini untuk membuat solusi pemantauan dan peringatan yang komprehensif untuk lingkungan Amazon EKS mereka.

- Amazon CloudWatch: Layanan AWS untuk pemantauan dan observabilitas

  CloudWatch menyediakan metrik, log, dan alarm untuk kluster EKS, dan terintegrasi dengan baik dengan yang lain. Layanan AWS
- Prometheus: Alat pemantauan dan peringatan open source untuk Kubernetes

  Prometheus menyediakan bahasa kueri yang kuat (PromQL) untuk menentukan kondisi peringatan.
- Alertmanager: Pendamping Prometheus untuk menangani peringatan

  Alertmanager menyediakan deduplikasi, pengelompokan, dan perutean peringatan. Ini mendukung berbagai saluran notifikasi, termasuk email, Slack, dan PagerDuty.
- Grafana: Platform open source untuk pemantauan dan observabilitas

  Grafana memberikan kemampuan visualisasi dan peringatan. Ini dapat berintegrasi dengan berbagai sumber data, termasuk Prometheus dan. CloudWatch
- Elastic Stack (ELK Stack): Kombinasi Elasticsearch, Logstash, dan Kibana

  Alat ini berguna untuk agregasi log, analisis, dan peringatan. Hal ini dapat diperluas dengan fitur observabilitas Elastic.
- Solusi pihak ketiga

  Ada banyak alat yang tersedia di pasaran, termasuk Datadog, New Relic, Sysdig, Dynatrace, Zabbix, Nagios, Splunk, IBM Instana, dan. AppDynamics

Praktik terbaik untuk peringatan di Amazon EKS

Bagian ini menjelaskan praktik terbaik untuk membuat sistem peringatan yang kuat yang meningkatkan keandalan dan kinerja aplikasi berbasis Kubernetes Anda di Amazon EKS.

Tentukan ambang batas peringatan yang jelas:

- Tetapkan ambang batas yang berarti berdasarkan data historis dan persyaratan bisnis.
- Gunakan ambang dinamis jika sesuai untuk memperhitungkan beban kerja yang bervariasi.

Menerapkan prioritas peringatan:

- Kategorikan peringatan berdasarkan tingkat keparahan (misalnya, kritis, tinggi, sedang, rendah).

- Sejajarkan prioritas peringatan dengan dampak bisnis.

Hindari kelelahan waspada:

- Kurangi kebisingan dengan menghilangkan peringatan berlebihan atau bernilai rendah.
- Korelasikan peringatan dengan masalah terkait grup.

Gunakan peringatan multi-tahap:

- Terapkan ambang peringatan sebelum level kritis tercapai.
- Gunakan saluran notifikasi yang berbeda untuk tingkat keparahan peringatan yang berbeda.

Menerapkan perutean peringatan yang tepat:

- Pastikan peringatan dikirim ke tim atau individu yang tepat.
- Gunakan jadwal panggilan dan rotasi untuk cakupan sepanjang hari, setiap hari.

Memanfaatkan metrik asli Kubernetes-:

- Pantau komponen inti Kubernetes (node, pod, layanan).
- Gunakan kube-state-metrics (KSM) untuk metrik objek Kubernetes tambahan.

Memantau infrastruktur dan aplikasi:

- Siapkan peringatan untuk kesehatan klaster, status node, dan pemanfaatan sumber daya.
- Menerapkan peringatan khusus aplikasi seperti tingkat kesalahan dan latensi.

Gunakan Prometheus dan Alertmanager:

- Gunakan Prometheus untuk pengumpulan metrik dan promQL untuk menentukan kondisi peringatan.
- Gunakan Alertmanager untuk perutean peringatan dan deduplikasi.

Integrasikan dengan Amazon CloudWatch:

- Gunakan CloudWatchWawasan Kontainer untuk metrik khusus Amazon EKS.
- Siapkan CloudWatchalarm untuk metrik AWS sumber daya penting.

Menerapkan lansiran kaya konteks:

- Sertakan informasi yang relevan dalam pesan peringatan, seperti nama cluster, namespace, dan detail pod.
- Berikan tautan ke dasbor atau runbook yang relevan dalam peringatan.

Gunakan deteksi anomali:

- Menerapkan deteksi anomali berbasis pembelajaran mesin untuk pola yang kompleks.
- Gunakan layanan seperti deteksi CloudWatch anomali atau alat pihak ketiga.

Menerapkan penindasan dan pembungkaman peringatan:

- Izinkan penindasan sementara dari masalah yang diketahui.
- Menerapkan jendela pemeliharaan untuk mengurangi kebisingan selama downtime yang direncanakan.

Pantau kinerja peringatan:

- Lacak metrik seperti frekuensi peringatan, waktu resolusi, dan tingkat positif palsu.
- Tinjau dan perbaiki aturan peringatan secara teratur berdasarkan metrik ini.

Menerapkan prosedur eskalasi:

- Tentukan jalur eskalasi yang jelas untuk peringatan yang belum terselesaikan.
- Gunakan alat seperti PagerDuty atau Otsgenie untuk eskalasi otomatis.

Uji sistem peringatan secara teratur:

- Lakukan tes berkala pada saluran peringatan Anda.
- Sertakan pengujian peringatan dalam latihan pemulihan bencana.

Gunakan templat untuk konsistensi peringatan:

- Buat templat peringatan standar untuk skenario umum.
- Pastikan pemformatan dan informasi yang konsisten di semua peringatan.

Menerapkan pembatasan tingkat:

- Cegah badai peringatan dengan menerapkan pembatasan laju pada peringatan yang sering dipicu.

Gunakan metrik khusus:

- Menerapkan metrik khusus untuk pemantauan khusus aplikasi.
- Gunakan API metrik kustom Kubernetes untuk penskalaan otomatis berdasarkan metrik ini.

Menerapkan integrasi logging:

- Korelasikan peringatan dengan log yang relevan untuk pemecahan masalah yang lebih cepat.
- Gunakan alat seperti Grafana Loki atau ELK Stack bersama dengan sistem peringatan Anda.

Pertimbangkan peringatan biaya:

- Siapkan peringatan untuk lonjakan tak terduga dalam penggunaan atau biaya sumber daya.
- Gunakan AWS Budgetsatau alat manajemen biaya pihak ketiga.

Gunakan penelusuran terdistribusi:

- Integrasikan alat penelusuran terdistribusi seperti Jaeger atau. AWS X-Ray
- Siapkan peringatan untuk pola atau latensi jejak abnormal.

Runbook peringatan dokumen:

- Buat runbook yang jelas dan dapat ditindaklanjuti untuk setiap jenis peringatan.
- Sertakan langkah-langkah pemecahan masalah dan prosedur eskalasi di runbook.

Dengan mengikuti praktik terbaik ini, Anda dapat membuat sistem peringatan yang kuat, efisien, dan efektif untuk lingkungan Amazon EKS Anda. Ini akan membantu memastikan ketersediaan tinggi, penyelesaian masalah yang cepat, dan kinerja optimal aplikasi berbasis Kubernetes Anda.

Langkah berikutnya

Panduan ini menyediakan kerangka kerja komprehensif untuk menerapkan observabilitas yang kuat di lingkungan Amazon EKS, dengan fokus pada pengumpulan metrik, infrastruktur logging, penelusuran terdistribusi, dan pengoptimalan biaya. Dengan memahami dan menerapkan komponen inti ini, Anda dapat membangun lingkungan kontainer yang sangat dapat diamati, dapat dipelihara, dan hemat biaya yang memberikan wawasan mendalam tentang perilaku aplikasi dan infrastruktur. Integrasi Layanan AWS seperti Amazon CloudWatch Container Insights dan AWS X-Ray, dikombinasikan dengan solusi sumber terbuka seperti Prometheus dan, menciptakan fondasi yang kuat untuk memantau dan memecahkan masalah OpenTelemetry aplikasi kontainer.

Keberhasilan implementasi bergantung pada pendekatan bertahap, dimulai dengan pengumpulan metrik inti dan secara bertahap berkembang ke kemampuan pencatatan dan penelusuran terdistribusi yang komprehensif. Kami menyarankan Anda memulai dengan menilai kemampuan pemantauan Anda saat ini, mengidentifikasi kesenjangan, dan memilih kombinasi perkakas yang sesuai yang sesuai dengan persyaratan operasional dan keahlian tim Anda. Pendekatan metodis ini memastikan bahwa setiap komponen tumpukan observabilitas diimplementasikan dan diintegrasikan dengan benar, sementara tim mengembangkan keterampilan dan proses yang diperlukan untuk menggunakan alat ini secara efektif.

Keberlanjutan jangka panjang dari observabilitas Amazon EKS bergantung pada optimalisasi biaya, sumber daya, dan proses secara teratur. Anda harus terus meninjau dan menyesuaikan infrastruktur observabilitas Anda, termasuk kebijakan retensi data, laju pengambilan sampel, dan alokasi sumber daya, untuk menjaga keseimbangan yang tepat antara pemantauan komprehensif dan efisiensi operasional. Pendekatan berulang untuk perbaikan ini, dikombinasikan dengan pelatihan tim yang sedang berlangsung dan pembaruan dokumentasi, memungkinkan organisasi Anda untuk mempertahankan observabilitas yang efektif sambil mendukung pertumbuhan bisnis dan beradaptasi dengan arsitektur aplikasi yang berkembang.

Sumber daya

AWS dokumentasi

- Panduan Praktik Terbaik Amazon EKS
- CloudWatchWawasan Kontainer Amazon
- Layanan Terkelola Amazon untuk Prometheus
- Amazon Managed Grafana
- AWS Distro untuk OpenTelemetry dan AWS X-Ray
- OpenSearch Layanan Amazon

AWS posting blog

- Amazon EKS meningkatkan observabilitas bidang kontrol Kubernetes
- Mengotomatiskan pengumpulan metrik di Amazon EKS dengan Layanan Terkelola Amazon untuk pencakar terkelola Prometheus
- Otomatiskan pemantauan untuk klaster Amazon EKS Anda menggunakan CloudWatch Container Insights
- Meningkatkan observabilitas dengan solusi pemantauan terkelola untuk Amazon EKS

Sumber daya lainnya

- Dokumentasi OpenTelemetry
- Dokumentasi Prometheus
- Dokumentasi Bit Lancar
- Monitoring, Logging, dan Debugging dalam dokumentasi Kubernetes

Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
Update	Kami memperbarui Bab Logging in Amazon EKS.	Maret 17, 2026
Publikasi awal	—	April 10, 2025

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

A2A () Agent-to-Agent

Protokol stateful untuk kolaborasi agen-ke-agen yang mendukung delegasi tugas dan transfer negara.

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

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

migrasi aktif-pasif

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

Agen

Sistem AI yang dapat secara mandiri bernalar, merencanakan, dan mengambil tindakan menggunakan alat untuk mencapai tujuan.

Agen Ops

Praktik operasional untuk membangun, menguji, menyebarkan, dan menjalankan agen AI dalam produksi dalam skala besar.

fungsi agregat

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

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani

sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat situs web AWS CAF dan whitepaper AWS CAF.

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

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

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

blue/green penyebaran

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

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

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

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

rekayasa kekacauan

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

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

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

Pengembang Warga

Pengguna bisnis yang membuat aplikasi AI menggunakan platform tanpa code/low kode tanpa keterampilan teknis khusus.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Cloud Center of Excellence (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat posting CCoE di Blog Strategi AWS Cloud Perusahaan.

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori dikhususkan untuk satu bagian fungsionalitas. Satu CI/CD pipa dapat menggunakan beberapa repositori.

cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat Membangun perimeter data pada AWS.

prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

DDL

Lihat bahasa definisi database.

ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

pertahanan-mendalam

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, pendekatan defense-in-depth mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat Layanan yang berfungsi dengan AWS Organizations AWS Organizations dokumentasi.

deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat bahasa manipulasi database.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

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

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

EDI

Lihat pertukaran data elektronik.

komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan komputasi awan, komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat Apa itu Pertukaran Data Elektronik.

enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

endianness

Urutan byte disimpan dalam memori komputer. Big-endian sistem menyimpan byte paling signifikan terlebih dahulu. Little-endian sistem menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat layanan titik akhir di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.

- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang dapat diakses oleh pengguna akhir. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.
- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat panduan implementasi program.

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat cabang.

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

Menyediakan LLM dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Few-shot prompt bisa efektif untuk tugas-tugas yang membutuhkan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga bidikan nol.

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui pengambilan data perubahan untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

FM

Lihat model pondasi.

model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FM mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat Apa itu Model Foundation.

Gerbang FM

Perantara terpusat yang mengontrol dan menormalkan akses ke model pondasi. Juga dikenal sebagai gateway LLM.

G

AI generatif

Subset model AI yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat Apa itu AI Generatif.

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat Membatasi distribusi geografis konten Anda dalam dokumentasi. CloudFront

Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan alur kerja berbasis batang adalah pendekatan modern yang lebih disukai.

gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur, gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (OU). Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

pagar pembatas (AI)

Mekanisme keamanan yang menyaring, memvalidasi, dan membatasi input dan output agen untuk membantu memastikan perilaku AI yang bertanggung jawab dan aman.

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. AWS menyediakan AWS SCT yang membantu dengan konversi skema.

ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran mesin. Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

manusia-dalam-lingkaran (HiTL)

Pola alur kerja di mana eksekusi agen berhenti untuk peninjauan dan persetujuan manusia pada titik keputusan kritis.

migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

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

masuk (masuknya) VPC

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

akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

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

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi selengkapnya, lihat Membangun strategi transformasi digital Internet of Things (IIoT) industri.

inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPC (dalam hal yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat Apa itu IoT?

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan. AWS

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan integrasi operasi.

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan.

model bahasa besar (LLM)

Model AI pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat Apa itu LLM.

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga endianness.

LLM

Lihat model bahasa besar.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat Program Percepatan Migrasi.

MCP

Lihat Protokol Konteks Model.

Protokol Konteks Model (MCP)

Protokol stateless untuk komunikasi agen -to- alat.

Server MCP

Layanan yang mengekspos satu atau lebih alat melalui Protokol Konteks Model.

mekanisme

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

akun anggota

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

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

Protokol komunikasi mesin-ke-mesin (M2M) yang ringan, berdasarkan pola publish/subscribe, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan API ringan. Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat Menerapkan layanan mikro di AWS.

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk

mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik terbaik dan pelajaran yang dipetik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari strategi AWS migrasi.

pabrik migrasi

Cross-functional tim yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat diskusi tentang pabrik migrasi dan panduan Pabrik Migrasi Cloud di kumpulan konten ini.

metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 AWS dengan Layanan Migrasi Aplikasi.

Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat pembelajaran mesin.

modernisasi

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

penilaian kesiapan modernisasi

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

aplikasi monolitik (monolit)

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

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

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

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

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

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu () OPC-UA

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

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

jejak organisasi

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

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

P

batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat Batas izin dalam dokumentasi IAM.

Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

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

persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

Kondisi kueri yang mengembalikan `true` atau`false`, biasanya terletak di `WHERE` klausa.

predikat pushdown

Teknik pengoptimalan kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

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

principal

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau beberapa VPC. Untuk informasi selengkapnya, lihat Bekerja dengan zona yang dihosting pribadi di dokumentasi Route 53.

kontrol proaktif

Kontrol keamanan yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat panduan referensi Kontrol

dalam AWS Control Tower dokumentasi dan lihat Kontrol proaktif dalam Menerapkan kontrol keamanan pada AWS.

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

Menggunakan output dari satu prompt LLM sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat Retrieval Augmented Generation.

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

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

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

refactor

Lihat 7 Rs.

Region

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

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

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

Retrieval Augmented Generation (RAG)

Teknologi AI generatif di mana LLM mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat Apa itu RAG.

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

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

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke Konsol Manajemen AWS atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

SCADA

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana

yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

Bayangan AI

Aplikasi AI yang tidak sah dibuat atau digunakan di luar saluran yang diatur dalam suatu organisasi.

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

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

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

model split-and-seed

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

SPOF

Lihat satu titik kegagalan.

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke LLM untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

T

tag

Key-value pasangan yang bertindak sebagai metadata untuk mengatur sumber daya Anda AWS . Tag membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai sumber daya AWS.

variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

alat

Fungsi atau API yang dapat dipanggil agen untuk melakukan operasi di sistem eksternal.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan VPC dan jaringan lokal Anda. Untuk informasi selengkapnya, lihat Apa itu gateway transit dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Organizations dengan AWS layanan lain dalam AWS Organizations dokumentasi.

penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian:

ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat lingkungan.

V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPC yang memungkinkan Anda merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Apa itu peering VPC di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

data hangat

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

fungsi jendela

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

beban kerja

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

aliran kerja

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

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat AWS Kerangka Kualifikasi Beban Kerja.

tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap tidak dapat diubah.

Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan zero-day.

kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

bisikan zero-shot

Memberikan LLM dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa bidikan yang diminta.

aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.