Kerangka Kerja AWS Well-Architected

Pilar Keunggulan Operasional

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

Pilar Keunggulan Operasional: Kerangka Kerja AWS Well-Architected

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

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

Table of Contents

Abstrak dan pengantar 1
Pengantar 1
Keunggulan operasional 3
Prinsip desain 3
Definisi 5
Organisasi 6
Prioritas organisasi 9
OPS01-BP01 Mengevaluasi kebutuhan pelanggan eksternal 10
OPS01-BP02 Mengevaluasi kebutuhan nasabah internal 11
OPS01-BP03 Mengevaluasi persyaratan tata kelola 12
OPS01-BP04 Evaluasi persyaratan kepatuhan 15
OPS01-BP05 Mengevaluasi lanskap ancaman 18
OPS01-BP06 Mengevaluasi kompromi sambil mengelola manfaat dan risiko 20
Model operasi 24
Representasi model operasi 2 per 2 26
Hubungan dan kepemilikan 35
Budaya organisasi 56
OPS03-BP01 Memberikan sponsor eksekutif 56
OPS03-BP02 Anggota tim diberdayakan untuk bertindak ketika hasil dipertaruhkan 60
OPS03-BP03 Eskalasi didorong 63
OPS03-BP04 Komunikasi yang tepat waktu, jelas, dan dapat ditindaklanjuti 66
OPS03-BP05 Mendorong eksperimen 72
OPS03-BP06 Mendorong dan mendukung para anggota tim untuk mempertahankan dan mengembangkan tingkat keterampilan mereka 75
OPS03-BP07 Tim sumber daya dengan tepat 78
Persiapkan 82
Implementasikan observabilitas 84
OPS04-BP01 Identifikasi indikator kinerja utama 84
OPS04-BP02 Melaksanakan telemetri aplikasi 86
OPS04-BP03 Menerapkan telemetri pengalaman pengguna 90
OPS04-BP04 Mengimplementasikan telemetri dependensi 93
OPS04-BP05 Melaksanakan penelusuran terdistribusi 96
Desain operasi 99
OPS05-BP01 Menggunakan kontrol versi 100

OPS05-BP02 Menguji dan memvalidasi perubahan 101
OPS05-BP03 Menggunakan sistem manajemen konfigurasi 105
OPS05-BP04 Menggunakan sistem manajemen build dan deployment 108
OPS05-BP05 Melakukan manajemen patch 111
OPS05-BP06 Bagikan standar desain 114
OPS05-BP07 Mengimplementasikan praktik untuk meningkatkan kualitas kode 117
OPS05-BP08 Menggunakan beberapa lingkungan 120
OPS05-BP09 Lakukan perubahan yang sering, kecil, dan reversibel 123
OPS05-BP10 Mengotomatiskan integrasi dan deployment sepenuhnya 124
Memitigasi risiko deployment 126
OPS06-BP01 Rencana untuk perubahan yang gagal 126
OPS06-BP02 Uji penerapan 129
OPS06-BP03 Menggunakan strategi deployment yang aman 132
OPS06-BP04 Mengotomatiskan pengujian dan rollback 136
Kesiapan operasional dan manajemen perubahan 140
OPS07-BP01 Memastikan kemampuan personel 141
OPS07-BP02 Memastikan peninjauan yang konsisten terkait kesiapan operasional 143
OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur 147
OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah 151
OPS07-BP05 Membuat keputusan yang tepat untuk melakukan deployment sistem dan perubahan 156
OPS07-BP06 Membuat rencana dukungan untuk beban kerja produksi 158
Jalankan 162
Memanfaatkan observabilitas beban kerja 163
OPS08-BP01 Menganalisis metrik beban kerja 164
OPS08-BP02 Menganalisis log beban kerja 166
OPS08-BP03 Menganalisis jejak beban kerja 169
OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti 172
OPS08-BP05 Membuat dasbor 175
Memahami kesehatan operasional 178
OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik 179
OPS09-BP02 Mengomunikasikan status dan tren untuk memastikan visibilitas beroperasi .. 181
OPS09-BP03 Meninjau metrik-metrik operasi dan memprioritaskan perbaikan 183
Merespons peristiwa 185
OPS10-BP01 Menggunakan proses untuk manajemen peristiwa, insiden, dan masalah 186
OPS10-BP02 Menjalankan proses untuk setiap peringatan 191

OPS10-BP03 Memprioritaskan peristiwa operasional berdasarkan dampaknya terhadap bisnis ... 196
OPS10-BP04 Tetapkan jalur eskalasi ... 199
OPS10-BP05 Menentukan rencana komunikasi pelanggan untuk peristiwa yang berdampak pada layanan ... 202
OPS10-BP06 Mengomunikasikan status melalui dasbor ... 205
OPS10-BP07 Otomatiskan tanggapan terhadap acara ... 208
Kembangkan ... 211
Belajar, berdiskusi, dan melakukan perbaikan ... 212
OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan ... 212
OPS11-BP02 Lakukan analisis pasca-insiden ... 214
OPS11-BP03 Menerapkan loop umpan balik ... 217
OPS11-BP04 Melakukan manajemen pengetahuan ... 220
OPS11-BP05 Menetapkan pendorong untuk perbaikan ... 223
OPS11-BP06 Validasi wawasan ... 225
OPS11-BP07 Lakukan tinjauan metrik operasi ... 227
OPS11-BP08 Mendokumentasikan dan membagikan pelajaran yang didapatkan ... 229
OPS11-BP09 Alokasikan waktu untuk melakukan perbaikan ... 231
Kesimpulan ... 233
Kontributor ... 234
Sumber bacaan lebih lanjut ... 235
Revisi dokumen ... 236
Pemberitahuan ... 239
AWS Glosarium ... 240

Pilar Keunggulan Operasional - Kerangka Kerja AWS Well-Architected

Tanggal publikasi: 6 November 2024 (Revisi dokumen)

Laporan ini berfokus pada pilar keunggulan operasional Kerangka Kerja AWS Well-Architected. Laporan ini menyediakan panduan untuk membantu Anda menerapkan praktik terbaik dalam desain, pengiriman, dan pemeliharaan beban kerja AWS.

Pengantar

Kerangka Kerja AWS Well-Architected akan membantu Anda memahami keputusan yang Anda ambil saat membuat beban kerja di AWS. Dengan menggunakan Kerangka Kerja ini, Anda akan mengetahui praktik-praktik terbaik operasional dan berkaitan dengan arsitektur untuk mendesain dan mengoperasikan beban kerja yang andal, aman, efisien, hemat biaya, dan ramah lingkungan di cloud. Kerangka Kerja ini menyediakan cara untuk secara terus menerus menilai operasi dan arsitektur Anda berdasarkan praktik terbaik dan mengidentifikasi area yang perlu diperbaiki. Kami percaya bahwa memiliki beban kerja Well-Architected yang didesain dengan mempertimbangkan operasi sangat meningkatkan kemungkinan keberhasilan bisnis.

Enam pilar landasan kerangka kerja:

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

Artikel ini berfokus pada pilar keunggulan operasional dan cara menerapkannya sebagai fondasi solusi Well-Architected Anda. Keunggulan operasional sulit dicapai di lingkungan di mana operasi dianggap sebagai fungsi yang terpisah dan berbeda dari lini-lini bisnis dan tim pengembangan yang didukungnya. Dengan mengadopsi praktik-praktik dalam artikel ini, Anda dapat membangun arsitektur yang menyediakan wawasan tentang statusnya, diaktifkan untuk operasi dan respons peristiwa yang efektif dan efisien, dan dapat terus meningkatkan dan mendukung tujuan bisnis Anda.

Artikel ini dimaksudkan untuk orang-orang yang memiliki peran di bidang teknologi, seperti kepala pejabat teknologi (CTO), arsitek, developer, dan anggota tim operasi. Setelah membaca artikel ini, Anda akan memahami praktik terbaik AWS dan strategi yang digunakan ketika merancang arsitektur cloud untuk keunggulan operasional. Artikel ini tidak menyediakan detail implementasi atau pola arsitektur. Namun, artikel ini menyertakan referensi ke sumber daya tepat untuk informasi ini.

Keunggulan operasional

Keunggulan operasional (OE) merupakan komitmen untuk membangun perangkat lunak dengan benar sambil memberikan pengalaman pelanggan yang luar biasa secara konsisten. Pilar keunggulan operasional berisi praktik terbaik untuk mengatur tim Anda, mendesain beban kerja Anda, mengoperasikannya dalam skala besar, dan mengembangkannya seiring waktu.

Sasaran keunggulan operasional adalah untuk menyediakan fitur baru dan perbaikan bug kepada pelanggan dengan cepat dan andal. Organisasi yang berinvestasi dalam keunggulan operasional akan secara konsisten membuat pelanggan puas sambil membangun fitur baru, membuat perubahan, dan menangani kegagalan. Dalam prosesnya, keunggulan operasional akan mengarah ke integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) dengan membantu para developer mencapai hasil berkualitas tinggi secara konsisten.

Prinsip desain

Berikut ini adalah prinsip-prinsip desain untuk meraih keunggulan operasional di cloud:

- Membentuk tim-tim berdasarkan tujuan bisnis yang ingin dicapai: Kemampuan sebuah tim untuk mencapai tujuan bisnis berasal dari visi kepemimpinan, operasi yang efektif, dan model operasional yang selaras dengan bisnis. Pimpinan harus sepenuhnya berinvestasi dan berkomitmen untuk transformasi CloudOps dengan model operasi cloud yang sesuai yang memberi insentif kepada tim untuk beroperasi dengan cara yang paling efisien dan memenuhi hasil bisnis. Model operasi yang tepat menggunakan kemampuan personel, proses, dan teknologi untuk menskalakan, mengoptimalkan produktivitas, serta membedakan melalui ketangkasan, responsivitas, dan adaptasi. Visi jangka panjang organisasi diwujudkan menjadi tujuan yang dikomunikasikan di seluruh korporasi kepada pemangku kepentingan dan konsumen layanan cloud Anda. Tujuan dan KPI operasional diselaraskan di semua tingkat. Praktik ini menopang nilai jangka panjang yang diperoleh dari penerapan prinsip-prinsip desain berikut.
- Implementasikan observabilitas untuk wawasan yang dapat ditindaklanjuti: Dapatkan pemahaman komprehensif tentang perilaku beban kerja, performa, keandalan, biaya, dan kesehatan. Tetapkan indikator kinerja utama (KPI) dan manfaatkan telemetri observabilitas untuk membuat keputusan yang lebih tepat dan mengambil tindakan cepat ketika hasil bisnis berisiko. Tingkatkan performa, keandalan, dan biaya secara proaktif berdasarkan data observabilitas yang dapat ditindaklanjuti.
- Lakukan otomatisasi yang aman jika memungkinkan: Di cloud, Anda bisa menerapkan disiplin teknik yang sama yang Anda gunakan untuk kode aplikasi pada seluruh lingkungan Anda. Anda

dapat menentukan seluruh beban kerja dan operasinya (aplikasi, infrastruktur, konfigurasi, dan prosedur) sebagai kode, dan memperbaruinya. Anda kemudian dapat mengotomatiskan operasi beban kerja Anda dengan memulainya sebagai respons terhadap peristiwa. Di cloud, Anda dapat menggunakan keamanan otomatisasi dengan mengonfigurasi pagar pembatas, termasuk pengontrolan tingkat, ambang batas kesalahan, dan persetujuan. Melalui otomatisasi yang efektif, Anda dapat mencapai respons yang konsisten terhadap peristiwa, membatasi kesalahan manusia, dan mengurangi kerja keras operator.

- Buat perubahan yang sering, kecil, dan dapat dibatalkan: Rancang beban kerja dapat disesuaikan skalanya dan perijinan komponen-komponennya tidak terlalu bergantung satu sama lain (loosely coupled) untuk dapat diperbarui secara rutin. Teknik deployment otomatis bersama dengan perubahan yang lebih kecil dan bertahap mengurangi radius ledakan dan memungkinkan pembalikan lebih cepat ketika terjadi kegagalan. Hal ini meningkatkan kepercayaan diri untuk memberikan perubahan yang menguntungkan pada beban kerja Anda sekaligus mempertahankan kualitas dan beradaptasi dengan cepat terhadap perubahan kondisi pasar.
- Perbaiki prosedur operasi sesering mungkin: Saat beban kerja Anda berkembang, prosedur operasional yang mendukungnya juga harus menyesuaikan. Saat Anda menggunakan prosedur operasi, carilah peluang untuk meningkatkannya. Lakukan peninjauan rutin dan pastikan bahwa semua prosedur sudah berlaku efektif dan dipahami dengan baik oleh tim. Jika kesenjangan diidentifikasi, perbarui prosedur yang sesuai. Komunikasikan pembaruan prosedural kepada semua pemangku kepentingan dan tim. Ciptakan mekanisme yang menyenangkan dalam operasi Anda untuk berbagi praktik terbaik dan mengedukasi tim.
- Mengantisipasi kegagalan: Maksimalkan keberhasilan operasional dengan menciptakan skenario kegagalan untuk memahami profil risiko beban kerja dan dampaknya terhadap hasil bisnis Anda. Uji keefektifan prosedur dan respons tim Anda terhadap simulasi kegagalan ini. Ambil keputusan yang bijaksana untuk mengelola risiko terbuka yang diidentifikasi dalam pengujian Anda.
- Pelajari dari semua peristiwa dan metrik operasional: Ciptakan peningkatan melalui pelajaran yang dipetik dari semua peristiwa dan kegagalan operasional. Bagikan materi yang telah dipelajari kepada seluruh tim dan organisasi. Pembelajaran harus menyoroti data dan anekdot tentang bagaimana operasi berkontribusi pada hasil bisnis.
- Menggunakan layanan terkelola: Kurangi beban operasional dengan menggunakan layanan terkelola AWS jika memungkinkan. Bangun prosedur operasional seputar interaksi dengan layanan tersebut.

Definisi

Ada empat area praktik terbaik untuk keunggulan operasional di cloud:

- Organisasi
- Persiapkan
- Jalankan
- Kembangkan

Kepemimpinan organisasi Anda menentukan tujuan bisnis. Organisasi Anda harus memahami kebutuhan dan prioritas serta menggunakannya untuk mengatur dan melakukan pekerjaan guna mendukung pencapaian hasil bisnis. Beban kerja Anda harus memberikan informasi yang diperlukan untuk mendukungnya. Mengimplementasi layanan untuk mengaktifkan integrasi, deployment, dan penyediaan beban kerja Anda akan menciptakan peningkatan alur perubahan yang menguntungkan menuju produksi dengan mengotomatiskan proses yang repetitif.

Operasi beban kerja Anda dapat memiliki risiko tersendiri. Anda harus memahami risiko tersebut dan mengambil keputusan yang matang untuk memasuki produksi. Tim Anda harus mampu mendukung beban kerja Anda. Dengan metrik bisnis dan operasional yang didapatkan dari hasil bisnis yang diinginkan, Anda dapat memahami kondisi beban kerja, aktivitas operasi, serta respons Anda terhadap insiden. Prioritas Anda akan berubah sesuai dengan kebutuhan bisnis Anda dan perubahan lingkungan bisnis. Gunakan ini sebagai loop umpan balik untuk mendorong peningkatan secara berkelanjutan bagi organisasi Anda dan operasi beban kerja Anda.

Organisasi

Tim Anda harus memiliki pemahaman bersama tentang seluruh beban kerja Anda, peran mereka di dalamnya, dan sasaran bisnis bersama untuk menetapkan prioritas yang akan mendukung kesuksesan bisnis. Prioritas yang terdefinisi dengan baik akan memaksimalkan manfaat dari upaya Anda. Evaluasi kebutuhan internal dan eksternal pelanggan dengan melibatkan pemangku kepentingan utama, termasuk tim bisnis, pengembangan, dan operasional, untuk menentukan arah fokus upaya. Mengevaluasi kebutuhan pelanggan akan memastikan Anda memiliki pemahaman menyeluruh mengenai dukungan yang dibutuhkan untuk mencapai hasil bisnis. Pastikan Anda mengetahui pedoman atau kewajiban yang ditetapkan oleh tata kelola organisasi Anda serta faktor eksternal, seperti persyaratan kepatuhan peraturan dan standar industri, yang mungkin mewajibkan atau menekankan fokus tertentu. Validasikan bahwa Anda memiliki mekanisme untuk mengidentifikasi perubahan pada tata kelola internal dan persyaratan kepatuhan eksternal. Jika persyaratan ini belum teridentifikasi, pastikan Anda telah menerapkan uji kelayakan untuk penetapan tersebut. Tinjau prioritas Anda secara berkala agar dapat diperbarui sesuai perubahan kebutuhan.

Evaluasi ancaman pada bisnis (misalnya, risiko dan kewajiban hukum bisnis, serta ancaman keamanan informasi) dan pelihara informasi ini pada registri risiko. Evaluasi dampak risiko, serta kompromi di antara kepentingan yang bertentangan atau pendekatan alternatif. Misalnya, meningkatkan kecepatan fitur baru ke pasar dapat lebih diprioritaskan daripada optimalisasi biaya, atau Anda dapat memilih basis data relasional untuk data non-relasional guna menyederhanakan upaya migrasi sistem tanpa pemfaktoran ulang. Kelola manfaat dan risiko untuk membuat keputusan yang tepat ketika menentukan arah fokus upaya. Risiko atau pilihan tertentu mungkin dapat diterima sesaat, risiko terkait mungkin dapat dimitigasi, atau membiarkan risiko tetap ada mungkin menjadi tidak dapat diterima. Jika demikian, Anda akan mengambil tindakan untuk mengatasi risiko tersebut.

Tim Anda harus memahami peran mereka dalam mencapai hasil bisnis. Tim wajib memahami peran mereka dalam kesuksesan tim lain, peran tim lain dalam kesuksesan mereka, dan memiliki sasaran bersama. Memahami tanggung jawab, kepemilikan, bagaimana keputusan diambil, dan siapa yang memiliki otoritas untuk mengambil keputusan akan membantu memfokuskan upaya dan memaksimalkan manfaat dari tim Anda. Kebutuhan sebuah tim akan dibentuk oleh pelanggan yang mereka dukung, organisasi mereka, formasi tim, dan karakteristik beban kerja mereka. Tidak realistis berharap pada satu model operasional untuk mampu mendukung semua tim dan beban kerja mereka di organisasi Anda.

Pastikan ada pemilik yang teridentifikasi untuk setiap aplikasi, beban kerja, platform, dan komponen infrastruktur, serta ada pemilik yang teridentifikasi untuk setiap proses dan prosedur yang bertanggung jawab atas definisinya, dan pemilik yang bertanggung jawab atas kinerja mereka.

Memahami nilai bisnis setiap komponen, proses, dan prosedur, tentang alasan penyediaan sumber daya atau alasan dilakukannya aktivitas, serta alasan adanya kepemilikan tersebut akan menjadi dasar tindakan anggota tim Anda. Tentukan dengan jelas tanggung jawab anggota tim sehingga mereka bisa bertindak dengan benar dan memiliki mekanisme untuk mengidentifikasi tanggung jawab dan kepemilikan. Miliki mekanisme untuk meminta penambahan, perubahan, dan pengecualian sehingga Anda tidak membatasi inovasi. Tetapkan perjanjian antar-tim yang menjelaskan tentang bagaimana mereka bekerja sama untuk mendukung satu sama lain dan mendukung hasil bisnis Anda.

Sediakan dukungan untuk anggota tim Anda agar mereka bisa menjadi lebih efektif dalam mengambil tindakan dan mendukung hasil bisnis Anda. Kepemimpinan senior yang terlibat harus menetapkan ekspektasi dan mengukur kesuksesan. Pimpinan senior harus menjadi pendukung, penasihat, dan pendorong untuk mengadopsi praktik terbaik maupun perkembangan organisasi. Biarkan anggota tim mengambil tindakan ketika terdapat risiko pada hasil agar dampak dapat diminimalkan, serta dorong mereka untuk melakukan eskalasi kepada pengambil keputusan dan pemangku kepentingan ketika mereka yakin terdapat risiko agar dapat segera ditangani dan insiden dapat dicegah. Sediakan komunikasi yang tepat waktu, jelas, dan dapat ditindaklanjuti tentang risiko yang diketahui serta peristiwa yang direncanakan agar anggota tim dapat mengambil tindakan yang tepat waktu dan sesuai.

Dorong pelaksanaan eksperimen untuk meningkatkan proses pembelajaran dan menjaga minat serta keterlibatan anggota tim. Tim harus mengembangkan set keterampilan mereka untuk mengadopsi perkembangan teknologi, serta untuk mengimbangi perubahan permintaan dan tanggung jawab. Dukung dan dorong hal ini dengan menyediakan waktu terstruktur khusus untuk pembelajaran. Pastikan anggota tim Anda memiliki sumber daya, berupa alat bantu dan anggota tim, untuk meraih keberhasilan serta skala untuk mendukung hasil bisnis Anda. Manfaatkan keragaman lintas organisasi untuk mencari berbagai perspektif unik. Gunakan perspektif ini untuk meningkatkan inovasi, menantang asumsi Anda, dan mengurangi risiko bias konfirmasi. Kembangkan inklusi, keragaman, dan kemudahan akses dalam tim Anda untuk mendapatkan perspektif yang bermanfaat.

Jika ada peraturan eksternal atau persyaratan kepatuhan yang berlaku untuk organisasi Anda, Anda harus menggunakan sumber daya yang disediakan oleh Kepatuhan Cloud AWS untuk membantu Anda memberikan edukasi bagi tim Anda agar mereka dapat menentukan dampak pada prioritas Anda. Kerangka Kerja Well-Architected menekankan pembelajaran, pengukuran, dan peningkatan.

Kerangka kerja ini menyediakan pendekatan yang konsisten bagi Anda untuk mengevaluasi arsitektur, dan mengimplementasikan desain yang akan meningkatkan skalanya (menskalakan) seiring waktu. AWS menyediakan AWS Well-Architected Tool untuk membantu Anda meninjau pendekatan Anda sebelum pengembangan, status beban kerja Anda sebelum produksi, serta status beban kerja Anda dalam produksi. Anda dapat membandingkan beban kerja dengan praktik-praktik terbaik yang berkaitan dengan arsitektur AWS terbaru, memantau statusnya secara keseluruhan, dan mendapatkan wawasan tentang potensi risiko. AWS Trusted Advisor adalah alat bantu yang menyediakan akses ke rangkaian inti pemeriksaan yang merekomendasikan optimalisasi yang dapat membantu membentuk prioritas Anda. Pelanggan Dukungan Bisnis dan Korporasi menerima akses ke pemeriksaan tambahan yang berfokus pada keamanan, keandalan, kinerja, optimalisasi biaya, dan keberlanjutan yang dapat membantu membentuk prioritas mereka lebih lanjut.

AWS dapat membantu mengedukasi tim Anda tentang AWS beserta layanannya untuk meningkatkan pemahaman mereka tentang bagaimana pilihan mereka dapat menimbulkan dampak pada beban kerja. Gunakan sumber daya yang disediakan oleh AWS Dukungan (Pusat Pengetahuan AWS, Forum Diskusi AWS, dan Pusat AWS Dukungan) serta Dokumentasi AWS untuk memberikan edukasi bagi tim Anda. Hubungi AWS Dukungan melalui Pusat AWS Dukungan untuk mengajukan pertanyaan-pertanyaan AWS Anda. AWS juga membagikan praktik terbaik serta pola yang telah kami pelajari melalui operasi AWS di Amazon Builders' Library. Beragam informasi berguna lainnya dapat diakses melalui Blog AWS dan Podcast Resmi AWS. AWS Pelatihan dan Sertifikasi menyediakan beberapa pelatihan melalui kursus digital mandiri tentang dasar-dasar AWS. Anda juga dapat mengikuti pelatihan yang dibimbing instruktur untuk mendukung perkembangan keterampilan AWS tim Anda.

Gunakan alat bantu atau layanan yang memungkinkan Anda mengelola lingkungan di seluruh akun secara terpusat, seperti AWS Organizations, untuk membantu mengelola model operasi Anda. Layanan seperti AWS Control Tower akan memperluas kemampuan manajemen ini dengan memudahkan Anda menentukan cetak biru (yang mendukung model operasi Anda) untuk penyiapan akun, menerapkan tata kelola berkelanjutan menggunakan AWS Organizations, dan mengotomatiskan penyediaan akun baru. Penyedia Layanan Terkelola seperti AWS Managed Services, Partner AWS Managed Services, atau Penyedia Layanan Terkelola di Jaringan Partner AWS, menyediakan keahlian implementasi lingkungan cloud, dan mendukung persyaratan keamanan dan kepatuhan serta tujuan bisnis Anda. Menambahkan Layanan Terkelola ke model operasi Anda dapat menghemat waktu dan sumber daya Anda, serta memungkinkan Anda menjaga tim internal Anda untuk bisa tetap belajar dan fokus pada hasil strategis yang akan membedakan bisnis Anda, dan bukan mengembangkan keterampilan dan kemampuan baru.

Anda mungkin mendapati bahwa Anda ingin mengutamakan sebagian kecil dari prioritas Anda pada titik waktu tertentu. Gunakan pendekatan yang seimbang dalam jangka panjang untuk memastikan pengembangan kemampuan yang diperlukan dan manajemen risiko. Tinjau prioritas Anda secara rutin dan perbarui sesuai perubahan kebutuhan. Ketika tanggung jawab dan kepemilikan tidak ditetapkan atau tidak diketahui, Anda menanggung risiko karena tidak melakukan tindakan yang diperlukan secara tepat waktu serta risiko munculnya upaya yang berulang dan berpotensi bertentangan untuk menangani kebutuhan-kebutuhan tersebut. Budaya organisasi berdampak langsung pada retensi dan kepuasan kerja anggota tim. Dukung keterlibatan dan kemampuan anggota tim Anda untuk mencapai keberhasilan bisnis Anda. Eksperimen diperlukan untuk menciptakan inovasi dan mewujudkan ide. Ketahui bahwa hasil yang tidak sesuai harapan merupakan eksperimen yang berhasil, karena dengan begitu jalur yang tidak mengarahkan kepada keberhasilan dapat diidentifikasi.

Topik

- Prioritas organisasi
- Model operasi
- Budaya organisasi

Prioritas organisasi

Tim Anda perlu memiliki pemahaman bersama tentang seluruh beban kerja Anda, peran mereka di dalamnya, dan sasaran bisnis bersama untuk menetapkan prioritas yang akan mewujudkan kesuksesan bisnis. Prioritas yang terdefinisi dengan baik akan memaksimalkan manfaat dari upaya Anda. Tinjau prioritas Anda secara rutin sehingga dapat diperbarui sesuai perubahan kebutuhan organisasi.

Praktik terbaik

- OPS01-BP01 Mengevaluasi kebutuhan pelanggan eksternal
- OPS01-BP02 Mengevaluasi kebutuhan nasabah internal
- OPS01-BP03 Mengevaluasi persyaratan tata kelola
- OPS01-BP04 Evaluasi persyaratan kepatuhan
- OPS01-BP05 Mengevaluasi lanskap ancaman
- OPS01-BP06 Mengevaluasi kompromi sambil mengelola manfaat dan risiko

OPS01-BP01 Mengevaluasi kebutuhan pelanggan eksternal

Libatkan pemangku kepentingan utama, termasuk tim bisnis, pengembangan, dan operasional, untuk menentukan ke mana harus memfokuskan usaha terkait kebutuhan pelanggan eksternal. Hal ini memverifikasi bahwa Anda memiliki pemahaman menyeluruh mengenai dukungan operasi yang dibutuhkan untuk mencapai hasil bisnis yang diinginkan.

Hasil yang diinginkan:

- Anda bekerja dengan berpatokan pada hasil pelanggan.
- Anda memahami bagaimana praktik operasional Anda mendukung hasil dan tujuan bisnis.
- Anda melibatkan semua pihak yang relevan.
- Anda memiliki mekanisme untuk merekam kebutuhan pelanggan eksternal.

Anti-pola umum:

- Anda memutuskan untuk tidak menyediakan dukungan pelanggan di luar jam kerja, tetapi Anda belum meninjau riwayat data permintaan dukungan. Anda tidak tahu apakah hal ini akan memengaruhi pelanggan Anda.
- Anda mengembangkan fitur baru, tetapi belum melibatkan pelanggan untuk mencari tahu apakah hal tersebut diinginkan—jika diinginkan, dalam bentuk apa—dan belum menjalankan eksperimen untuk memvalidasi kebutuhan serta metode penyediaannya.

Manfaat penerapan praktik terbaik: Pelanggan yang kebutuhannya terpenuhi akan sangat berpotensi menjadi pelanggan tetap. Mengevaluasi dan memahami kebutuhan pelanggan eksternal akan menginformasikan cara Anda memprioritaskan usaha untuk memberikan nilai bisnis.

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

Panduan implementasi

Pahami kebutuhan bisnis: Kesuksesan bisnis terwujud dengan tujuan dan pemahaman bersama di seluruh pemangku kepentingan, termasuk tim bisnis, pengembangan, dan operasional.

Tinjau tujuan bisnis, kebutuhan, dan prioritas pelanggan eksternal: Libatkan para pemangku kepentingan utama, termasuk tim bisnis, pengembangan, dan operasional, untuk mendiskusikan tujuan, kebutuhan, dan prioritas pelanggan eksternal. Hal ini memastikan bahwa Anda memiliki

pemahaman menyeluruh mengenai dukungan operasional yang dibutuhkan untuk mencapai hasil bisnis dan pelanggan.

Tetapkan pemahaman bersama: Tetapkan pemahaman bersama terkait fungsi bisnis beban kerja, peran masing-masing tim dalam mengoperasikan beban kerja, dan bagaimana faktor-faktor ini mendukung tujuan bisnis bersama bagi seluruh pelanggan internal dan eksternal.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP03 Mengimplementasikan loop umpan balik

OPS01-BP02 Mengevaluasi kebutuhan nasabah internal

Libatkan pemangku kepentingan utama, termasuk tim bisnis, pengembangan, dan operasional, untuk menentukan ke mana harus memfokuskan usaha terkait kebutuhan pelanggan internal. Hal ini akan memastikan bahwa Anda memiliki pemahaman menyeluruh mengenai dukungan operasi yang dibutuhkan untuk mencapai hasil bisnis yang diinginkan.

Hasil yang diinginkan:

- Anda menggunakan prioritas yang ditetapkan untuk memfokuskan usaha peningkatan yang dapat memberikan dampak paling besar (misalnya, mengembangkan keterampilan tim, meningkatkan kinerja beban kerja, mengurangi biaya, mengotomatiskan runbook, atau meningkatkan pemantauan).
- Anda memperbarui prioritas Anda sesuai perubahan kebutuhan.

Anti-pola umum:

- Anda memutuskan untuk mengubah alokasi alamat IP untuk tim produk tanpa berkonsultasi dengan mereka agar manajemen jaringan menjadi lebih mudah. Anda tidak tahu dampak yang akan ditimbulkan kepada tim produk.
- Anda mengimplementasikan alat pengembangan baru tetapi belum melibatkan pelanggan internal untuk mencari tahu apakah alat itu dibutuhkan atau kompatibel dengan praktik yang sudah ada.
- Anda mengimplementasikan sistem pemantauan baru, tetapi belum menghubungi pelanggan internal untuk mencari tahu apakah mereka memiliki kebutuhan pemantauan atau pelaporan yang perlu dipertimbangkan.

Manfaat menerapkan praktik terbaik: Mengevaluasi dan memahami kebutuhan pelanggan internal akan menginformasikan cara Anda memprioritaskan usaha untuk memberikan nilai bisnis.

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

Panduan implementasi

- Pahami kebutuhan bisnis: Kesuksesan bisnis diciptakan dengan tujuan dan pemahaman bersama di seluruh pemangku kepentingan termasuk tim bisnis, pengembangan, dan operasional.
- Tinjau tujuan bisnis, kebutuhan, dan prioritas pelanggan internal: Libatkan pemangku kepentingan utama, termasuk tim bisnis, pengembangan, dan operasional, untuk mendiskusikan tujuan, kebutuhan, dan prioritas pelanggan internal. Hal ini memastikan bahwa Anda memiliki pemahaman menyeluruh mengenai dukungan operasional yang dibutuhkan untuk mencapai hasil bisnis dan pelanggan.
- Tetapkan pemahaman bersama: Tetapkan pemahaman bersama terkait fungsi bisnis beban kerja, peran masing-masing tim dalam mengoperasikan beban kerja, dan bagaimana faktor-faktor ini mendukung tujuan bisnis bersama bagi seluruh pelanggan internal dan eksternal.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP03 Menerapkan loop umpan balik

OPS01-BP03 Mengevaluasi persyaratan tata kelola

Tata kelola adalah serangkaian kebijakan, aturan, atau kerangka kerja yang digunakan perusahaan untuk mencapai sasaran bisnisnya. Persyaratan tata kelola dibuat dari dalam organisasi Anda. Persyaratan ini dapat memengaruhi jenis teknologi yang Anda pilih atau memengaruhi cara Anda mengoperasikan beban kerja Anda. Sertakan persyaratan tata kelola organisasi ke dalam beban kerja Anda. Konformitas adalah kemampuan untuk menunjukkan bahwa Anda telah mengimplementasikan persyaratan tata kelola.

Hasil yang diinginkan:

- Persyaratan tata kelola disertakan ke dalam operasi dan desain arsitektur beban kerja Anda.
- Anda dapat memberikan bukti bahwa Anda telah mengikuti persyaratan tata kelola.

- Persyaratan tata kelola ditinjau dan diperbarui secara teratur.

Anti-pola umum:

- Organisasi Anda memerintahkan agar akun root memiliki autentikasi multi-faktor. Anda gagal mengimplementasikan persyaratan ini dan akun root terancam bahaya.
- Selama desain beban kerja Anda, Anda memilih jenis instans yang tidak disetujui oleh departemen IT. Anda tidak dapat meluncurkan beban kerja Anda dan harus mendesain ulang.
- Anda diwajibkan memiliki rencana pemulihan bencana. Anda tidak membuat rencana pemulihan bencana dan beban kerja Anda mengalami pemadaman yang berdurasi lama.
- Tim Anda ingin menggunakan instans baru tetapi persyaratan tata kelola Anda belum diperbarui untuk memungkinkannya.

Manfaat menjalankan praktik terbaik ini:

- Mengikuti persyaratan tata kelola akan menyelaraskan beban kerja Anda dengan kebijakan lebih besar dalam organisasi.
- Persyaratan tata kelola mencerminkan standar industri dan praktik terbaik untuk organisasi Anda.

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

Panduan implementasi

Identifikasi persyaratan tata kelola melalui kerja sama dengan pemangku kepentingan dan organisasi tata kelola. Sertakan persyaratan tata kelola ke dalam beban kerja Anda. Dapat menunjukkan bukti bahwa Anda telah mengikuti persyaratan tata kelola.

Contoh pelanggan

Di AnyCompany Retail, tim operasi cloud bekerja dengan pemangku kepentingan di seluruh organisasi untuk mengembangkan persyaratan tata kelola. Misalnya, mereka melarang SSH akses ke EC2 instans Amazon. Jika tim memerlukan akses ke sistem, mereka harus menggunakan AWS Systems Manager Session Manager. Tim operasi cloud secara teratur memperbarui persyaratan tata kelola saat layanan baru tersedia.

Langkah-langkah implementasi

1. Identifikasi pemangku kepentingan untuk beban kerja Anda, termasuk semua tim tersentralisasi.

2. Bekerja sama dengan pemangku kepentingan untuk mengidentifikasi persyaratan tata kelola.
3. Setelah Anda membuat daftar, prioritaskan item untuk ditingkatkan, dan mulai implementasikan ke dalam beban kerja Anda.
   a. Gunakan layanan seperti AWS Configuntuk membuat governance-as-code dan memvalidasi bahwa persyaratan tata kelola diikuti.
   b. Jika Anda menggunakan AWS Organizations, Anda dapat memanfaatkan Kebijakan Kontrol Layanan untuk menerapkan persyaratan tata kelola.
4. Berikan dokumentasi yang memvalidasi implementasinya.

Tingkat upaya untuk rencana implementasi: Sedang. Mengimplementasikan persyaratan tata kelola yang tidak ada dapat mengakibatkan beban kerja Anda harus dikerjakan ulang.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP04 Evaluasi persyaratan kepatuhan - Kepatuhan itu seperti tata kelola tetapi kepatuhan berasal dari luar organisasi.

Dokumen terkait:

- AWS Panduan Lingkungan Cloud Manajemen dan Tata Kelola
- Praktik Terbaik untuk Kebijakan Kontrol AWS Organizations Layanan di Lingkungan Multi-Akun
- Tata Kelola dalam AWS Cloud: Keseimbangan yang Tepat Antara Kelincahan dan Keamanan
- Apa itu Tata Kelola, Risiko, dan Kepatuhan (GRC)?

Video terkait:

- AWS Manajemen dan Tata Kelola: Konfigurasi, Kepatuhan, dan Audit - AWS Pembicaraan Teknologi Online
- AWS re: Inforce 2019: Tata Kelola untuk Zaman Awan (-R1) DEM12
- AWS re:invent 2020: Mencapai kepatuhan sebagai menggunakan kode AWS Config
- AWS Re:invent 2020: Tata kelola tangkas di AWS GovCloud (US)

Contoh terkait:

- AWS Config Sampel Paket Kesesuaian

Layanan terkait:

- AWS Config
- AWS Organizations - Kebijakan Pengendalian Layanan

OPS01-BP04 Evaluasi persyaratan kepatuhan

Persyaratan kepatuhan peraturan, industri, dan internal merupakan pendorong penting dalam menentukan prioritas organisasi Anda. Kerangka kerja kepatuhan Anda dapat menghalangi Anda untuk menggunakan teknologi atau lokasi geografi tertentu. Terapkan uji tuntas jika tidak ada kerangka kerja kepatuhan eksternal yang diidentifikasi. Buatlah audit atau laporan yang memvalidasi kepatuhan.

Jika Anda mengiklankan bahwa produk Anda memenuhi standar kepatuhan tertentu, maka Anda harus memiliki proses internal untuk memastikan kepatuhan yang berkelanjutan. Contoh standar kepatuhan antara lain PCI DSS, FedRAMP, dan HIPAA. Standar kepatuhan yang berlaku akan ditentukan oleh berbagai faktor, seperti jenis data yang disimpan atau dikirim oleh solusi, serta wilayah geografis mana yang didukung oleh solusi.

Hasil yang diinginkan:

- Persyaratan kepatuhan berdasarkan regulasi, industri, dan internal disertakan ke dalam pemilihan arsitektur.
- Anda dapat memvalidasi kepatuhan dan membuat laporan audit.

Anti-pola umum:

- Bagian dari beban kerja Anda termasuk dalam kerangka kerja Standar Keamanan Data Industri Kartu Pembayaran (Payment Card Industry Data Security Standard, PCI-DSS) tetapi beban kerja Anda menyimpan data kartu kredit yang tidak dienkripsi.
- Arsitek dan pengembang perangkat lunak Anda tidak mengetahui kerangka kerja kepatuhan yang harus ditaati oleh organisasi Anda.
- Audit tahunan terhadap Kontrol Sistem dan Organisasi (SOC2) Tipe II akan segera diadakan dan Anda tidak dapat memverifikasi bahwa kontrol tersebut sudah ada atau tidak.

Manfaat menjalankan praktik terbaik ini:

- Mengevaluasi dan memahami persyaratan kepatuhan yang berlaku untuk beban kerja Anda akan menginformasikan bagaimana Anda memprioritaskan upaya-upaya Anda untuk memberikan nilai bisnis.
- Anda memilih teknologi dan lokasi yang tepat yang selaras dengan kerangka kerja kepatuhan Anda.
- Mendesain beban kerja Anda agar dapat diaudit akan membantu Anda membuktikan bahwa Anda menaati kerangka kerja kepatuhan.

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

Panduan implementasi

Mengimplementasikan praktik terbaik ini berarti Anda menyertakan persyaratan kepatuhan ke dalam proses desain arsitektur Anda. Anggota tim Anda mengetahui kerangka kerja kepatuhan yang diperlukan. Anda memvalidasi bahwa kepatuhan selaras dengan kerangka kerja.

Contoh pelanggan

AnyCompany Retail menyimpan informasi kartu kredit bagi para pelanggan. Para pengembang di tim penyimpanan kartu memahami bahwa mereka harus mematuhi kerangka kerja PCI-DSS. Mereka telah mengambil langkah-langkah yang diperlukan untuk memverifikasi bahwa informasi kartu kredit disimpan dan diakses dengan aman sesuai dengan kerangka kerja PCI-DSS. Setiap tahun mereka bekerja sama dengan tim keamanan mereka untuk melakukan validasi kepatuhan.

Langkah-langkah implementasi

1. Bekerjasamalah dengan tim tata kelola dan kepatuhan Anda untuk menentukan kerangka kerja kepatuhan industri, peraturan, atau internal apa yang harus ditaati oleh beban kerja Anda. Sertakan kerangka kerja kepatuhan ke dalam beban kerja Anda.
   a. Lakukan validasi kepatuhan berkelanjutan terhadap sumber daya AWS dengan mengunakan layanan seperti AWS Compute Optimizer dan AWS Security Hub CSPM.
2. Didik anggota tim Anda mengenai persyaratan kepatuhan sehingga mereka dapat mengoperasikan dan mengubah beban kerja sesuai dengan persyaratan kepatuhan. Persyaratan kepatuhan harus disertakan dalam pilihan-pilihan berkaitan dengan arsitektur dan teknologi.

3. Tergantung pada kerangka kerja kepatuhan yang diterapkan, Anda mungkin diharuskan untuk membuat laporan kepatuhan atau audit. Bekerjasamalah dengan organisasi Anda untuk melakukan otomatisasi terhadap proses ini sebanyak mungkin.
   a. Gunakan layanan-layanan seperti AWS Audit Manager untuk memvalidasi kepatuhan dan menghasilkan laporan audit.
   b. Anda dapat mengunduh dokumen-dokumen keamanan dan kepatuhan AWS dengan AWS Artifact.

Tingkat upaya untuk rencana implementasi: Sedang. Mengimplementasikan kerangka kerja kepatuhan bisa jadi sulit dilakukan. Membuat laporan audit atau dokumen kepatuhan menambahkan kompleksitas tambahan.

Sumber daya

Praktik-praktik terbaik terkait:

- SEC01-BP03 Melakukan identifikasi dan validasi tujuan kontrol - Tujuan pengendalian keamanan adalah bagian penting dari kepatuhan secara keseluruhan.
- SEC01-BP06 Melakukan otomatisasi terhadap pengujian dan validasi kontrol keamanan di pipeline - Sebagai bagian dari pipeline Anda, validasi kontrol keamanan. Anda juga dapat membuat dokumentasi kepatuhan untuk perubahan-perubahan baru.
- SEC07-BP02 Tentukan kontrol perlindungan data - Banyak kerangka kerja kepatuhan memiliki penanganan data dan kebijakan yang berbasis penyimpanan.
- SEC10-BP03 Siapkan kemampuan forensik - Kemampuan forensik terkadang dapat digunakan dalam kepatuhan audit.

Dokumen terkait:

- Pusat Kepatuhan AWS
- Sumber Daya Kepatuhan AWS
- Laporan Resmi Risiko dan Kepatuhan AWS
- Model Tanggung Jawab Bersama AWS
- Layanan AWS dalam lingkup sesuai program kepatuhan

Video terkait:

- AWS re:Invent 2020: Capai kepatuhan sebagai kode dengan menggunakan AWS Compute Optimizer
- AWS re:Invent 2021 - Kepatuhan, jaminan, dan audit cloud
- AWS Summit ATL 2022 - Menerapkan kepatuhan, jaminan, dan audit pada AWS (COP202)

Contoh terkait:

- PCI DSS dan Praktik Terbaik Keamanan Dasar AWS pada AWS

Layanan terkait:

- AWS Artifact
- AWS Audit Manager
- AWS Compute Optimizer
- AWS Security Hub CSPM

OPS01-BP05 Mengevaluasi lanskap ancaman

Evaluasi ancaman pada bisnis (misalnya, persaingan, risiko dan kewajiban bisnis, risiko operasional, serta ancaman keamanan informasi) dan pelihara informasi yang ada di registri risiko. Sertakan dampak risiko ketika menentukan ke mana upaya harus difokuskan.

Kerangka Kerja Well-Architected menekankan pembelajaran, pengukuran, dan peningkatan. Ini memberikan pendekatan yang konsisten bagi Anda untuk mengevaluasi arsitektur, dan menerapkan desain yang akan skala dari waktu ke waktu. AWS menyediakan AWS Well-Architected Tooluntuk membantu Anda meninjau pendekatan Anda sebelum pengembangan, keadaan beban kerja Anda sebelum produksi, dan keadaan beban kerja Anda dalam produksi. Anda dapat membandingkannya dengan praktik terbaik AWS arsitektur terbaru, memantau status keseluruhan beban kerja Anda, dan mendapatkan wawasan tentang potensi risiko.

AWS pelanggan memenuhi syarat untuk Tinjauan Well-Architected yang dipandu dari beban kerja kritis misi mereka untuk mengukur arsitektur mereka terhadap praktik terbaik. AWS Pelanggan Dukungan Perusahaan memenuhi syarat untuk Peninjauan Operasi, yang dirancang untuk membantu mereka mengidentifikasi kesenjangan yang ada dalam pendekatan mereka untuk beroperasi di cloud.

Interaksi lintas tim pada tinjauan ini akan membantu Anda dalam membangun pemahaman bersama tentang beban kerja Anda serta bagaimana peran tim akan membantu meraih keberhasilan. Kebutuhan yang diidentifikasi melalui tinjauan tersebut dapat membantu membentuk prioritas Anda.

AWS Trusted Advisor adalah sebuah alat yang menyediakan akses ke set inti pemeriksaan yang menyarankan optimalisasi yang dapat membantu membentuk prioritas Anda. Pelanggan Dukungan Bisnis dan Perusahaan menerima akses ke pemeriksaan tambahan yang berfokus pada keamanan, keandalan, kinerja, dan optimisasi biaya yang dapat membantu membentuk prioritas mereka lebih lanjut.

Hasil yang diinginkan:

- Anda secara teratur meninjau dan bertindak berdasarkan Well-Architected dan output Trusted Advisor
- Anda mengetahui status patch terbaru layanan Anda
- Anda memahami risiko dan dampak ancaman yang diketahui dan bertindak sebagaimana mestinya
- Anda mengimplementasikan mitigasi sesuai keperluan
- Anda mengomunikasikan tindakan dan konteks

Anti-pola umum:

- Anda menggunakan pustaka perangkat lunak versi lama dalam produk Anda. Anda tidak mengetahui bahwa ada pembaruan keamanan pustaka untuk masalah-masalah yang mungkin memiliki dampak yang tidak diinginkan pada beban kerja Anda.
- Kompetitor Anda baru saja merilis sebuah versi produk mereka yang dapat mengatasi keluhan pelanggan Anda tentang produk Anda. Anda belum memprioritaskan penanganan masalah-masalah yang sudah diketahui ini.
- Pembuat peraturan telah menyasar perusahaan-perusahaan yang tidak mematuhi persyaratan kepatuhan hukum seperti Anda. Anda belum memprioritaskan penanganan persyaratan kepatuhan Anda yang belum terpenuhi.

Manfaat menerapkan praktik terbaik ini: Anda dapat mengidentifikasi dan memahami ancaman terhadap organisasi dan beban kerja Anda, hal ini akan membantu Anda menentukan ancaman mana yang harus ditangani, tingkat prioritasnya, serta sumber daya yang diperlukan untuk melakukannya.

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

Panduan implementasi

- Evaluasi lanskap ancaman: Evaluasi ancaman terhadap bisnis (misalnya kompetisi, risiko dan kewajiban bisnis, risiko operasional, dan ancaman keamanan informasi), sehingga Anda dapat menyertakan dampaknya ketika menentukan ke mana upaya perlu difokuskan.
  - Buletin Keamanan Terkini AWS
  - AWS Trusted Advisor
- Pelihara model ancaman: Buat dan pelihara model ancaman yang mengidentifikasi potensi ancaman, mitigasi terencana dan sedang diterapkan, serta prioritasnya. Tinjau kemungkinan ancaman yang berwujud insiden, biaya untuk melakukan pemulihan dari insiden tersebut serta perkiraan bahaya yang ditimbulkan, dan biaya untuk mencegah terjadinya insiden tersebut. Revisi proritas seiring perubahan konten model ancaman.

Sumber daya

Praktik terbaik terkait:

- SEC01-BP07 Mengidentifikasi ancaman dan memprioritaskan mitigasi menggunakan model ancaman

Dokumen terkait:

- Kepatuhan AWS Cloud
- Buletin Keamanan Terkini AWS
- AWS Trusted Advisor

Video terkait:

- AWS re: Inforce 2023 - Alat untuk membantu Anda meningkatkan pemodelan ancaman Anda

OPS01-BP06 Mengevaluasi kompromi sambil mengelola manfaat dan risiko

Kepentingan yang saling berbenturan dari berbagai pihak dapat menyulitkan Anda dalam membuat prioritas upaya, pembangunan kemampuan, dan pemberian hasil yang selaras dengan strategi bisnis. Misalnya, Anda mungkin diminta untuk memprioritaskan peningkatan kecepatan masuk pasar untuk fitur-fitur baru daripada optimalisasi biaya infrastruktur IT. Hal ini dapat menjadikan dua pihak

yang berkepentingan mengalami konflik satu sama lain. Dalam situasi seperti ini, keputusan perlu dibawa ke otoritas yang lebih tinggi untuk menyelesaikan konflik. Data diperlukan agar tidak ada keterikatan emosional dalam proses pengambilan keputusan.

Tantangan yang sama dapat terjadi pada tingkatan taktis. Misalnya, ketika ada pilihan antara menggunakan teknologi basis data relasional atau non-relasional, hal itu dapat berdampak signifikan pada pengoperasian aplikasi. Sangat penting bagi Anda untuk memahami hasil yang dapat diprediksi dari berbagai pilihan.

AWS dapat membantu mengedukasi tim Anda tentang AWS beserta layanannya untuk meningkatkan pemahaman mereka tentang bagaimana pilihan mereka dapat menimbulkan dampak pada beban kerja. Gunakan sumber daya yang disediakan oleh Dukungan (Pusat Pengetahuan AWS, Forum Diskusi AWS, dan Pusat Dukungan) serta Dokumentasi AWS untuk memberikan edukasi bagi tim Anda. Untuk pertanyaan lebih lanjut, hubungi Dukungan.

AWS juga membagikan pola dan praktik terbaik di Amazon Builders' Library. Beragam informasi berguna lainnya dapat diakses melalui Blog AWS dan Siniar Resmi AWS.

Hasil yang diharapkan: Anda memiliki kerangka kerja pengambilan keputusan yang jelas untuk memudahkan Anda mengambil keputusan penting di setiap level dalam organisasi pengiriman cloud Anda. Kerangka kerja ini mencakup fitur-fitur seperti pencatatan risiko, peran yang ditentukan untuk wewenang pengambilan keputusan, dan model yang ditentukan untuk masing-masing tingkat keputusan yang dapat diambil. Kerangka kerja ini juga menetapkan di awal bagaimana konflik akan diselesaikan, data apa yang perlu disajikan, dan bagaimana opsi diprioritaskan, sehingga setelah keputusan diambil, Anda dapat menjalankannya tanpa jeda. Kerangka pengambilan keputusan ini mencakup pendekatan terstandardisasi dalam meninjau dan menimbang manfaat-manfaat serta risiko dari setiap keputusan yang diambil untuk memahami kompromi. Ini mungkin mencakup faktor-faktor eksternal, seperti misalnya, kepatuhan terhadap persyaratan kepatuhan berdasarkan regulasi.

Anti-pola umum:

- Investor Anda meminta Anda untuk mendemonstrasikan kepatuhan terhadap Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS). Anda tidak mempertimbangkan kompromi antara memenuhi permintaan mereka dan melanjutkan upaya-upaya pengembangan yang Anda lakukan saat ini. Alih-alih, Anda melanjutkan upaya-upaya pengembangan tanpa menunjukkan kepatuhan. Investor Anda menghentikan dukungan untuk perusahaan Anda karena mengkhawatirkan keamanan platform Anda serta investasi yang mereka tanamkan.

- Anda telah memutuskan untuk menyertakan pustaka yang ditemukan oleh salah satu developer Anda di internet. Anda belum mengevaluasi risiko adopsi pustaka ini dari sumber tak dikenal dan Anda tidak tahu jika pustaka ini memiliki kerentanan atau kode berbahaya.
- Pembenaran bisnis asli untuk migrasi Anda didasarkan pada modernisasi 60% beban kerja aplikasi Anda. Namun demikian, karena hambatan teknis, akhirnya diputuskan untuk melakukan modernisasi terhadap hanya 20% beban kerja tersebut, yang mengakibatkan berkurangnya manfaat yang direncanakan dalam jangka panjang, bertambahnya beban operasional bagi tim infrastruktur untuk mendukung sistem warisan secara manual, dan ketergantungan yang lebih besar pada pengembangan keterampilan baru di dalam tim infrastruktur yang tidak merencanakan perubahan ini.

Manfaat menerapkan praktik terbaik ini: Sepenuhnya menyelaraskan dan mendukung prioritas bisnis tingkat dewan, memahami risiko untuk mencapai kesuksesan, membuat keputusan yang tepat berdasarkan informasi, dan bertindak dengan tepat ketika ada risiko yang menghambat peluang untuk mencapai kesuksesan. Memahami implikasi dan konsekuensi keputusan yang Anda ambil akan membantu Anda dalam menyusun prioritas opsi dan menghadirkan kesepakatan para pemimpin dengan lebih cepat, yang akan mengarahkan Anda pada hasil bisnis yang lebih baik. Mengidentifikasi manfaat-manfaat yang bisa dihadirkan oleh pilihan Anda dan menyadari risiko yang ditimbulkannya terhadap organisasi akan membantu Anda mengambil keputusan dengan berlandaskan data, bukan opini.

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

Panduan implementasi

Manajemen manfaat dan risiko harus ditentukan oleh badan pengatur yang mendorong persyaratan untuk pengambilan keputusan utama. Anda ingin keputusan diambil dan diprioritaskan berdasarkan bagaimana keputusan tersebut menguntungkan organisasi, dengan memahami risiko-risiko yang mungkin dihadapi. Informasi yang akurat sangat penting untuk mengambil keputusan organisasi. Hal ini harus didasarkan pada pengukuran yang solid dan ditentukan oleh praktik industri umum mengenai analisis manfaat biaya. Untuk mengambil keputusan semacam ini, bangun keseimbangan antara otoritas tersentralisasi dan terdesentralisasi. Kompromi akan selalu ada, dan penting untuk memahami bagaimana setiap pilihan akan memengaruhi strategi yang ditentukan dan hasil bisnis yang diinginkan.

Langkah-langkah implementasi

1. Susun praktik pengukuran manfaat dalam sebuah kerangka kerja tata kelola cloud yang menyeluruh.
   a. Seimbangkan kontrol pengambilan keputusan tersentralisasi dengan otoritas terdesentralisasi untuk beberapa keputusan.
   b. Pahami bahwa proses pengambilan keputusan yang memberatkan yang diterapkan pada setiap keputusan akan memperlambat Anda.
   c. Sertakan faktor-faktor eksternal ke dalam proses pengambilan keputusan Anda (seperti persyaratan kepatuhan).
2. Tetapkan kerangka kerja pengambilan keputusan yang telah disepakati untuk berbagai tingkat keputusan, yang menyertakan orang yang diminta untuk menengahi keputusan yang mengalami benturan kepentingan.
   a. Pusatkan pengambilan keputusan satu arah yang tidak dapat dibatalkan.
   b. Izinkan pengambilan keputusan dua arah oleh pemimpin organisasi tingkat bawah.
3. Pahami dan kelola manfaat serta risiko. Seimbangkan manfaat keputusan sesuai risiko yang terlibat.
   a. Identifikasi manfaat: Identifikasi manfaat berdasarkan tujuan, kebutuhan, dan prioritas bisnis. Contohnya antara lain dampak kasus bisnis, waktu masuk pasar, keamanan, keandalan, performa, dan biaya.
   b. Identifikasi risiko: Identifikasi risiko berdasarkan tujuan, kebutuhan, dan prioritas bisnis. Contohnya antara lain waktu masuk pasar, keamanan, keandalan, performa, dan biaya.
   c. Evaluasi manfaat dibandingkan risiko dan ambil keputusan yang bijaksana: Tentukan dampak manfaat dan risiko berdasarkan tujuan, kebutuhan, dan prioritas pemangku kepentingan utama Anda, termasuk bagian bisnis, pengembangan, dan operasi. Lakukan evaluasi terhadap nilai dari manfaat yang didapatkan dibanding dengan probabilitas terjadinya risiko dan kerugian yang ditimbulkan oleh dampaknya. Sebagai contoh, menekankan pada kecepatan masuk pasar dan bukannya keandalan dapat memberikan keunggulan dalam persaingan. Tetapi, hal ini dapat mengakibatkan berkurangnya waktu aktif jika ada masalah keandalan yang terjadi.
4. Secara terprogram, berlakukan keputusan utama yang mengotomatiskan kesesuaian Anda terhadap persyaratan-persyaratan kepatuhan.
5. Manfaatkan kerangka kerja dan kemampuan industri yang sudah diketahui, seperti Value Stream Analysis dan LEAN, untuk membuat garis acuan performa saat ini, metrik bisnis, dan menentukan iterasi progres menuju peningkatan terhadap metrik-metrik ini.

Tingkat upaya untuk rencana implementasi: Sedang cenderung Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP05 Mengevaluasi lanskap ancaman

Dokumen terkait:

- Elemen-elemen Amazon, Hari 1 Budaya | Buat keputusan berkualitas tinggi dan berkecepatan tinggi
- Tata Kelola Cloud
- Manajemen dan Tata Kelola Lingkungan Cloud
- Tata Kelola di Cloud dan di Era Digital: Bagian Satu & Dua

Video terkait:

- Siniar | Jeff Bezos | Tentang cara membuat keputusan

Contoh terkait:

- Buat keputusan berdasarkan informasi dengan menggunakan data (DevOps Sagas)
- Menggunakan pemetaan aliran nilai pengembangan untuk mengidentifikasi kendala pada hasil DevOps

Model operasi

Pada bagian ini, kami telah menyediakan cara untuk memahami model operasi tempat Anda bekerja, bagaimana model itu dapat divisualisasikan, dan bagaimana, pada tingkat tim, Anda harus berevolusi untuk mengekstrak nilai maksimum dari investasi Anda yang ditanamkan dalam layanan cloud. Dengan demikian, Anda dapat meningkatkan praktik operasional Anda, membangun tim dan beban kerja yang gesit, dan berkontribusi positif terhadap hasil bisnis Anda.

Sudah menjadi hal umum bagi tim Anda untuk hadir dalam beberapa lapisan organisasi, dan lapisan tersebut memiliki cara kerja yang sekarang ada. Dengan partisipasi Anda bersama tim Anda dalam mencapai hasil bisnis artinya Anda harus memahami di mana tim Anda berada di lapisan tersebut,

posisi tim yang berinteraksi dengan Anda, dan bagaimana cara kerja mereka. Berikutnya, tim harus memahami peran mereka dalam kesuksesan tim lain, mengetahui peran tim lain dalam kesuksesan mereka, dan memiliki sasaran bersama.

Lapisan-lapisan ini membentuk model operasi keseluruhan organisasi. Bagaimana organisasi berfungsi untuk memberikan hasil bisnis akan tergantung pada banyak faktor, seperti jenis, industri, geografi, ukuran, dan tingkat otonomi. Namun demikian, bisa dikelompokkan dalam tiga kategori besar:

- Tersentralisasi
- Terdesentralisasi
- Terfederasi

Topologi tingkat organisasi ini dijelaskan dalam Mengatur untuk kesuksesan.

Tim dan beban kerja Anda hadir dalam model operasi organisasi Anda. Namun demikian, satu model operasi saja tentunya tidak cukup untuk mendukung semua tim dan beban kerja mereka. Oleh karena itu, tim Anda juga membutuhkan model operasinya sendiri. Cara kerja ini dibentuk oleh organisasi Anda, departemen Anda, susunan tim Anda, dan karakteristik beban kerja itu sendiri.

Sebagian besar organisasi yang berpindah ke cloud melakukannya sebagai bagian dari program transformasi perusahaan yang berupaya untuk membuka cara kerja baru (model operasi) untuk mendukung tujuan strategis jangka panjang. Perjalanan ini bukanlah sebuah titik dalam lini masa latihan, tetapi merupakan sebuah proses yang membutuhkan evolusi berkelanjutan dan kemajuan bertahap menuju tujuan strategis. Hal ini akan memungkinkan pemilik beban kerja untuk terus beradaptasi dengan model operasi yang berkembang dengan hanya mengalami sedikit gangguan.

Amazon sering digunakan sebagai contoh bagaimana organisasi besar mampu berinovasi dalam skala besar dengan memberdayakan tim untuk tetap dekat dengan pelanggan, meluncurkan produk dan layanan inovatif dengan cepat, dan memanfaatkan arsitektur teknis yang mendukung kecepatan dan kelincahan. Ini mengharuskan kami untuk merestrukturisasi bagaimana tim kami diatur, dan sekarang dikenal sebagai tim dua pizza. Tim dua pizza memiliki semua sumber daya yang tepat yang tertanam di dalamnya (teknik, pengujian, manajemen produk dan program, dan operasi) untuk memiliki dan menjalankan beban kerja secara menyeluruh (end-to-end).

Kami menyarankan Anda untuk berupaya mewujudkan model operasi ini sebagai cara yang terbukti bagi tim beban kerja untuk bisa bergerak dengan cepat dan berkontribusi pada hasil bisnis secara keseluruhan dengan cara yang paling baik dalam melayani pelanggan mereka.

Organisasi-organisasi yang ingin meniru keberhasilan ini mungkin perlu menyesuaikan model operasi mereka di sepanjang perjalanan transformasi mereka. Di tingkat organisasi dan tim, hal ini membutuhkan pertimbangan, perencanaan, dan komunikasi. Bagian berikut menyediakan cara yang bisa dilakukan untuk memvisualisasikan model operasi tingkat tim ini dan bagaimana mengembangkannya dengan prinsip Anda membangunnya, Anda menjalankannya.

Representasi model operasi 2 per 2

Representasi model operasi 2 per 2 merupakan ilustrasi untuk membantu Anda memahami hubungan antar-tim dalam lingkungan. Diagram ini berfokus pada tindakan dan pelaksananya serta hubungan antar-tim, selain itu kami juga akan mendiskusikan tata kelola dan pembuatan keputusan yang berkaitan dengan contoh ini.

Tim Anda dapat memiliki tanggung jawab dalam berbagai model bergantung pada beban kerja yang mereka dukung. Anda mungkin ingin mencoba area disiplin yang lebih terspesialisasi daripada yang dijelaskan. Variasi model ini dapat menjadi tak terbatas seiring Anda memisahkan atau menggabungkan aktivitas, atau menyebarkan tim dan memberikan detail spesifik.

Anda dapat mengidentifikasi bahwa Anda memiliki kemampuan yang tumpang tindih atau tidak diketahui untuk seluruh tim, yang dapat memberikan keuntungan tambahan, atau mendorong efisiensi. Anda juga dapat mengidentifikasi kebutuhan yang belum terpenuhi dalam organisasi dan dapat merencanakan penanganannya.

Saat mengevaluasi perubahan organisasi, pelajari kompensasi antara model, posisi tim individu Anda dalam model (sekarang dan setelah perubahan), bagaimana hubungan dan kemampuan tim akan berubah, dan apakah manfaat sesuai dengan dampak pada organisasi.

Anda bisa sukses menggunakan masing-masing dari empat model operasi berikut. Beberapa model lebih sesuai untuk kasus penggunaan tertentu atau pada titik tertentu dalam pengembangan. Beberapa model dapat memberikan manfaat lebih dari model yang digunakan dalam lingkungan.

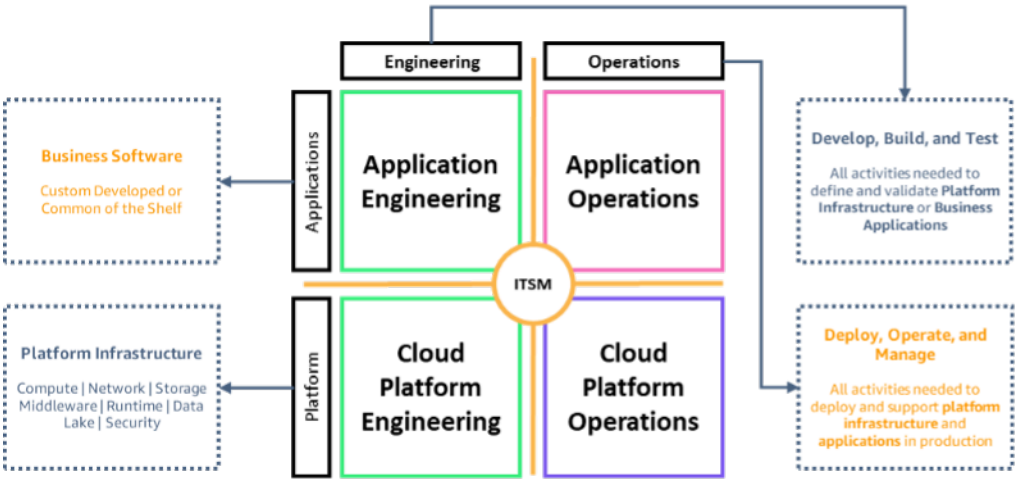
Topik

- Model operasi yang terpisah sepenuhnya
- DevOps dengan penyedia layanan terkelola cloud
- Operasi cloud dan pemberdayaan platform (COPE)
- DevOps Terdistribusi
- DevOps Terdesentralisasi
- Mengembangkan model operasi Anda

Model operasi yang terpisah sepenuhnya

Dalam diagram berikut, Aplikasi dan Infrastruktur berada pada sumbu vertikal. Aplikasi merujuk pada beban kerja yang digunakan dalam hasil bisnis dan dapat dikembangkan secara kustom atau perangkat lunak yang dibeli. Platform merujuk pada infrastruktur fisik serta virtual dan perangkat lunak lainnya yang mendukung beban kerja tersebut.

Rekayasa dan Operasi berada pada sumbu horizontal. Rekayasa merujuk pada pengembangan, pembangunan, dan pengujian aplikasi serta infrastruktur. Operasi adalah deployment, pembaruan, dan dukungan yang masih berlangsung untuk aplikasi dan infrastruktur.



Model tradisional

Secara historis, organisasi-organisasi biasanya menganut kerangka kerja seperti ITIL atau standar seperti ISO dan membentuk kegiatan operasional mereka sesuai standar-standar itu, yang sering kali menghasilkan topologi yang sepenuhnya terpisah. Dalam model ini, aktivitas dalam setiap kuadran dilakukan oleh tim yang berbeda. Pekerjaan diteruskan antar tim melalui mekanisme seperti permintaan pekerjaan, antrean, tiket, atau dengan menggunakan sistem manajemen layanan IT (ITSM).

Transisi tugas kepada atau antar-tim dapat meningkatkan kompleksitas, serta menyebabkan bottleneck dan penundaan. Permintaan mungkin ditunda hingga menjadi prioritas. Kecacatan yang terlambat diidentifikasi dapat memerlukan banyak upaya untuk pengerjaan ulang dan mungkin harus melewati tim yang sama beserta fungsinya satu kali lagi. Jika ada insiden yang memerlukan tindakan tim rekayasawan, respons mereka akan ditunda oleh aktivitas transfer.

Ada risiko ketidaksesuaian yang lebih tinggi saat bisnis, pengembangan, dan tim operasi dikelola di sekitar aktivitas atau fungsi yang sedang dijalankan. Hal ini dapat membuat tim berfokus pada

tanggung jawab spesifik mereka, bukan pada mencapai hasil bisnis. Tim dapat terspesialisasi secara sempit, terisolasi secara fisik, atau terisolasi secara logis, menghambat komunikasi dan kolaborasi.

DevOps dengan penyedia layanan terkelola cloud

DevOps dengan model penyedia layanan yang dikelola cloud akan mengikuti metodologi Anda membangunnya, Anda menjalankannya untuk tim aplikasi. Namun demikian, organisasi Anda mungkin belum memiliki keterampilan, atau anggota tim, untuk mendukung rekayasa platform khusus dan tim operasi, atau Anda mungkin tidak ada dalam posisi ingin meluangkan waktu dan tenaga untuk melakukannya.

Jika tidak, Anda mungkin ingin memiliki tim platform yang berkonsentrasi pada pengembangan kemampuan yang akan memberikan bisnis Anda diferensiasi, tetapi Anda ingin melakukan outsourcing pada operasi harian yang tidak mendatangkan keuntungan kompetitif dan diferensiasi.

Penyedia Layanan Terkelola seperti AWS Managed Services, atau Penyedia AWS Partner Network, menyediakan keahlian untuk mengimplementasikan lingkungan cloud, dan mendukung persyaratan keamanan dan kepatuhan serta tujuan bisnis Anda.

DevOps dengan penyedia layanan terkelola cloud

Untuk variasi ini, kami akan menetapkan agar tata kelola dipusatkan dan dikelola oleh tim platform, dengan pembuatan akun dan kebijakan yang dikelola dengan AWS Organizations dan AWS Control Tower.

Dalam model ini, mekanisme harus diubah agar dapat dijalankan dengan penyedia layanan tersebut. Hal ini tidak mengatasi bottleneck dan penundaan yang disebabkan oleh transisi tugas antar-tim, termasuk penyedia layanan, atau kemungkinan pengerjaan ulang terkait dengan keterlambatan identifikasi kecacatan.

Anda mendapatkan manfaat dari standar penyedia, praktik terbaik, proses, dan keahlian. Anda juga mendapatkan manfaat dari pengembangan berkelanjutan penawaran layanan.

Menambahkan layanan terkelola ke model operasi Anda dapat menghemat waktu dan sumber daya Anda, serta memungkinkan Anda menjaga tim internal Anda untuk tetap belajar dan fokus pada hasil strategis yang akan membedakan bisnis Anda, dan bukan mengembangkan keterampilan dan kemampuan baru. Hal ini juga dapat memberikan waktu bagi Anda untuk membangun dan mematangkan kemampuan platform Anda sendiri tanpa memperlambat program migrasi cloud Anda.

Operasi cloud dan pemberdayaan platform (COPE)

Model cloud operations and platform enablement (COPE) ini berupaya membangun metodologi Anda membangunnya, Anda menjalankannya dengan mendukung tim aplikasi untuk melakukan aktivitas rekayasa dan operasi untuk beban kerja mereka, mengadopsi budaya DevOps.

Tim aplikasi Anda mungkin bertugas untuk melakukan migrasi, mengadopsi cloud, atau melakukan modernisasi beban kerja, dan tidak memiliki kemampuan yang ada untuk mendukung arsitektur cloud serta operasi cloud secara memadai. Kurangnya kemampuan tim aplikasi dan keakraban ini cenderung akan memperlambat kelincahan organisasi Anda dan berdampak pada hasil bisnis.

Untuk mengatasi masalah ini, Anda dapat menggunakan keahlian operasional yang ada dari dalam organisasi Anda untuk mendukung tim aplikasi yang sedang berproses mengadopsi operasi cloud. Ini dapat berupa tim ahli tim virtual khusus yang di beranggotakan peserta-peserta pilihan dari seluruh organisasi. Namun demikian, tujuannya tetap sama, yaitu memberikan dukungan operasional yang membangun kemampuan tim beban kerja, menggunakan prinsip otomatisasi cloud first, menghilangkan upaya-upaya yang menuntut tetapi tidak memberikan diferensiasi, menyediakan pola standar, dan mempromosikan otonomi. Tujuannya adalah untuk membangun kematangan yang cukup di seluruh kemampuan cloud dan menurunkan hambatan tanggung jawab operasional sehingga tim aplikasi tidak lagi membutuhkan dukungan tambahan.

Model COPE berkonsentrasi pada tingkat beban kerja. Jika pendekatan ini diperlukan di beberapa tim sekaligus, dan jika Anda melakukan proyek migrasi multi-tahun yang kompleks dan dalam skala besar, atau jika Anda sedang membangun platform untuk mendukung inisiatif ini, Anda harus mempertimbangkan untuk menggunakan Pusat Keunggulan Cloud (CCoE). Ini adalah sebuah mekanisme yang banyak ditemukan berhasil ketika dilakukan dalam upaya untuk mempercepat migrasi mereka ke cloud dan secara luas mengubah organisasi mereka.

Operasi Cloud dan Pemberdayaan Platform (COPE)

Tim rekayasa platform Anda membangun lapisan tipis kemampuan platform bersama inti, yang didasarkan pada standar yang telah ditentukan sebelumnya yang harus diadopsi oleh tim aplikasi dan disediakan oleh tim COPE. Tim rekayasa platform mengodekan arsitektur referensi dan pola yang disediakan untuk tim aplikasi melalui mekanisme layanan mandiri. Dengan menggunakan layanan seperti AWS Service Catalog, tim aplikasi dapat melakukan deployment arsitektur referensi, pola, layanan, dan konfigurasi yang disetujui, serta secara default mematuhi tata kelola terpusat dan standar keamanan.

Tim rekayasa platform juga menyediakan set layanan yang terstandardisasi (misalnya, alat pengembangan, alat observabilitas, alat pencadangan dan pemulihan, serta jaringan) kepada tim aplikasi.

Tim COPE mengelola dan mendukung layanan-layanan terstandardisasi dan memberikan bantuan kepada tim aplikasi yang membangun kehadiran cloud mereka berdasarkan arsitektur dan pola yang dijadikan referensi. Tim COPE bekerja dengan tim aplikasi untuk membantu mereka membangun operasi dasar. Selama proses ini, tim aplikasi akan mengemban tanggung jawab yang semakin besar atas sistem dan sumber daya mereka dari waktu ke waktu. Tim COPE mendorong peningkatan berkelanjutan bersama dengan tim rekayasa platform serta bertindak sebagai pendukung untuk tim aplikasi.

Tim aplikasi mendapatkan bantuan dalam menyiapkan lingkungan, pipeline CI/CD, manajemen perubahan, observabilitas dan pemantauan, serta menetapkan insiden dan proses manajemen peristiwa dengan tim COPE yang terintegrasi sebagaimana yang diperlukan. Tim COPE bekerja sama dengan tim aplikasi dalam menjalankan aktivitas operasi ini, menghapus keterlibatan tim COPE dari waktu ke waktu saat tim aplikasi mengambil alih kepemilikan.

Tim aplikasi memperoleh manfaat dari keterampilan tim COPE dan pelajaran yang dipetik oleh organisasi. Mereka dilindungi oleh pagar pembatas yang didirikan melalui tata kelola tersentralisasi. Tim aplikasi melakukan pembangunan berdasarkan kesuksesan yang diakui dan mendapatkan manfaat dari pengembangan berkelanjutan dari standar organisasi yang telah diadopsi. Mereka mendapatkan wawasan ke operasi beban kerja melalui proses penetapan observabilitas dan pemantauan dan dapat memahami dampak perubahan yang dibuat untuk beban kerja dengan lebih baik.

Tim COPE juga dapat mempertahankan akses yang diperlukan untuk mendukung aktivitas operasi, memberikan tampilan operasi perusahaan yang mencakup tim aplikasi, dan untuk menyediakan dukungan manajemen insiden yang sangat penting. Tim COPE mempertahankan tanggung jawab untuk aktivitas yang dianggap sebagai pekerjaan berat yang tidak mendatangkan keuntungan kompetitif, yang mereka penuhi melalui solusi standar yang dapat didukung dalam skala besar. Mereka juga terus mengelola aktivitas operasi terprogram dan otomatis yang dipahami dengan baik untuk tim aplikasi sehingga mereka dapat fokus menjadikan aplikasi mereka kompetitif.

Anda mendapatkan keuntungan dari standar, praktik terbaik, proses, dan keahlian organisasi yang diperoleh dari keberhasilan tim Anda. Anda menetapkan mekanisme untuk mereplikasi pola yang berhasil ini untuk tim baru yang melakukan adopsi atau modernisasi di cloud. Model ini menekankan pada kemampuan tim COPE untuk membantu penetapan tim aplikasi, serta mentransisikan pengetahuan dan artefak. Hal ini akan mengurangi beban operasional tim aplikasi dengan risiko

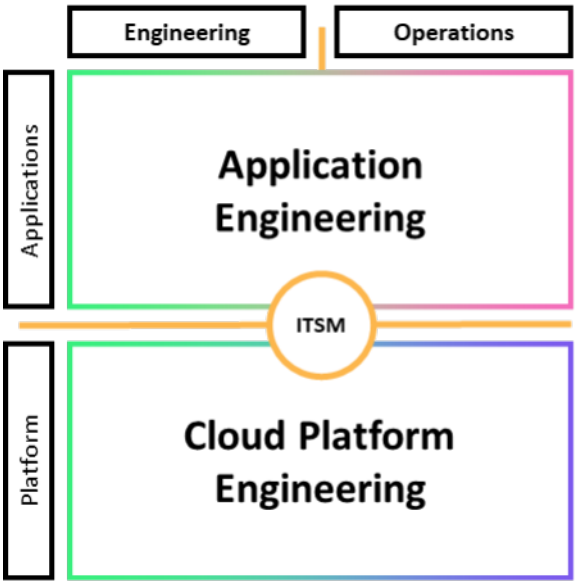
bahwa tim aplikasi bisa gagal menjadi independen secara umum. Hal ini juga akan membangun hubungan antara CCoE, COPE, dan tim aplikasi dalam membuat loop umpan balik guna mendukung perkembangan dan inovasi lebih lanjut.

Menetapkan tim rekayasa platform dan COPE, sekaligus menentukan standar di seluruh organisasi, dapat memfasilitasi adopsi cloud dan mendukung upaya modernisasi yang dilakukan. Dengan memberikan dukungan tambahan dari tim COPE yang bertindak sebagai konsultan dan mitra untuk tim aplikasi, Anda dapat menghilangkan hambatan tingkat beban kerja yang memperlambat adopsi tim aplikasi dari kemampuan cloud yang bermanfaat.

DevOps Terdistribusi

Model DevOps terdistribusi memisahkan (atau mendistribusikan) operasi rekayasa aplikasi dan tanggung jawab operasi rekayasa infrastruktur di seluruh tim teknik, mengikuti metodologi COPE.

Rekayasawan aplikasi Anda melakukan rekayasa dan operasi beban kerja mereka. Dengan cara yang sama, rekayasawan infrastruktur melakukan rekayasa dan operasi pada platform yang digunakan untuk mendukung tim aplikasi.



DevOps Terdistribusi

Untuk contoh ini, kami memperlakukan tata kelola sebagai tata kelola terpusat di tempat lain dalam organisasi. Standar sudah didistribusikan, disediakan, atau dibagikan kepada tim aplikasi.

Gunakan alat atau layanan yang memungkinkan Anda mengelola lingkungan di seluruh akun Anda secara tersentralisasi, seperti AWS Organizations. Layanan seperti AWS Control Tower akan memperluas kemampuan manajemen ini dengan membantu Anda menentukan cetak biru (yang

mendukung model operasi Anda) untuk penyiapan akun, menerapkan tata kelola berkelanjutan menggunakan AWS Organizations, dan mengotomatiskan penyediaan akun baru.

You build it you run it bukan berarti tim aplikasi bertanggung jawab atas tumpukan penuh (full-stack), tool chain, dan platform.

Tim rekayasawan platform menyediakan set layanan yang terstandardisasi (misalnya, alat pengembangan, alat pemantauan, alat pencadangan dan pemulihan, serta jaringan) kepada tim aplikasi. Tim platform juga dapat menyediakan ke layanan penyedia cloud yang disetujui, konfigurasi tertentu, atau keduanya, kepada tim aplikasi.

Mekanisme yang menyediakan kemampuan mandiri untuk men-deploy layanan dan konfigurasi yang disetujui, seperti Service Catalog, dapat membantu membatasi penundaan yang terkait dengan permintaan pemenuhan, dan sekaligus menegakkan tata kelola.

Tim platform mengaktifkan visibilitas tumpukan penuh (full stack) agar tim aplikasi dapat membedakan antara masalah dengan komponen aplikasi dan layanan serta komponen infrastruktur yang digunakan aplikasi. Tim platform juga dapat menyediakan bantuan untuk mengonfigurasi layanan ini dan panduan tentang cara meningkatkan operasi sebuah tim aplikasi.

Seperti yang didiskusikan sebelumnya, tersedianya mekanisme sangat penting untuk meminta penambahan, perubahan, dan pengecualian terhadap standar guna mendukung aktivitas dan inovasi aplikasi.

Model DevOps terdistribusi memberikan loop umpan balik yang kuat kepada tim aplikasi. Operasi harian beban kerja meningkatkan kontak dengan pelanggan, baik melalui interaksi langsung atau tidak langsung lewat permintaan fitur dan dukungan. Visibilitas yang lebih tinggi ini memungkinkan tim aplikasi untuk menangani masalah dengan cepat. Keterlibatan yang mendalam dan hubungan yang lebih dekat memberikan wawasan atas kebutuhan pelanggan dan menciptakan inovasi yang lebih cepat.

Semua ini juga berlaku untuk tim platform yang mendukung tim aplikasi, karena tim platform harus melihat tim aplikasi ini sebagai pelanggan mereka.

Standar yang diadopsi mungkin belum disetujui untuk penggunaan, mengurangi jumlah peninjauan yang diperlukan untuk memasuki produksi. Menggunakan standar yang didukung dan diuji serta disediakan oleh tim platform dapat mengurangi frekuensi masalah dengan layanan tersebut. Pengadopsian standar akan membantu tim aplikasi untuk fokus menjadikan beban kerja mereka kompetitif.

DevOps Terdesentralisasi

Model DevOps terdesentralisasi adalah variasi dari metodologi Anda membangunnya, Anda menjalankannya di mana operasi sebagian besar berada di bawah kepemilikan tim beban kerja.

Rekayasawan aplikasi Anda melakukan rekayasa dan operasi-operasi beban kerja mereka. Dengan cara yang sama, rekayasawan infrastruktur melakukan rekayasa dan operasi-operasi pada platform yang digunakan untuk mendukung tim aplikasi.

DevOps Terdesentralisasi

Untuk contoh ini, kami membuat tata kelola menjadi terdesentralisasi. Standar masih didistribusikan, disediakan, atau dibagikan kepada tim aplikasi oleh tim platform, tetapi tim aplikasi bebas untuk merekayasa dan mengoperasikan kemampuan platform baru guna mendukung beban kerja.

Dalam model ini, tim aplikasi mengalami lebih sedikit kendala, tetapi hal itu disertai dengan peningkatan tanggung jawab yang signifikan. Keterampilan tambahan (dan anggota tim potensial) harus ada untuk mendukung kemampuan platform tambahan. Risiko pengerjaan ulang yang signifikan akan meningkat jika keahlian tidak memadai dan kecacatan tidak teridentifikasi lebih awal.

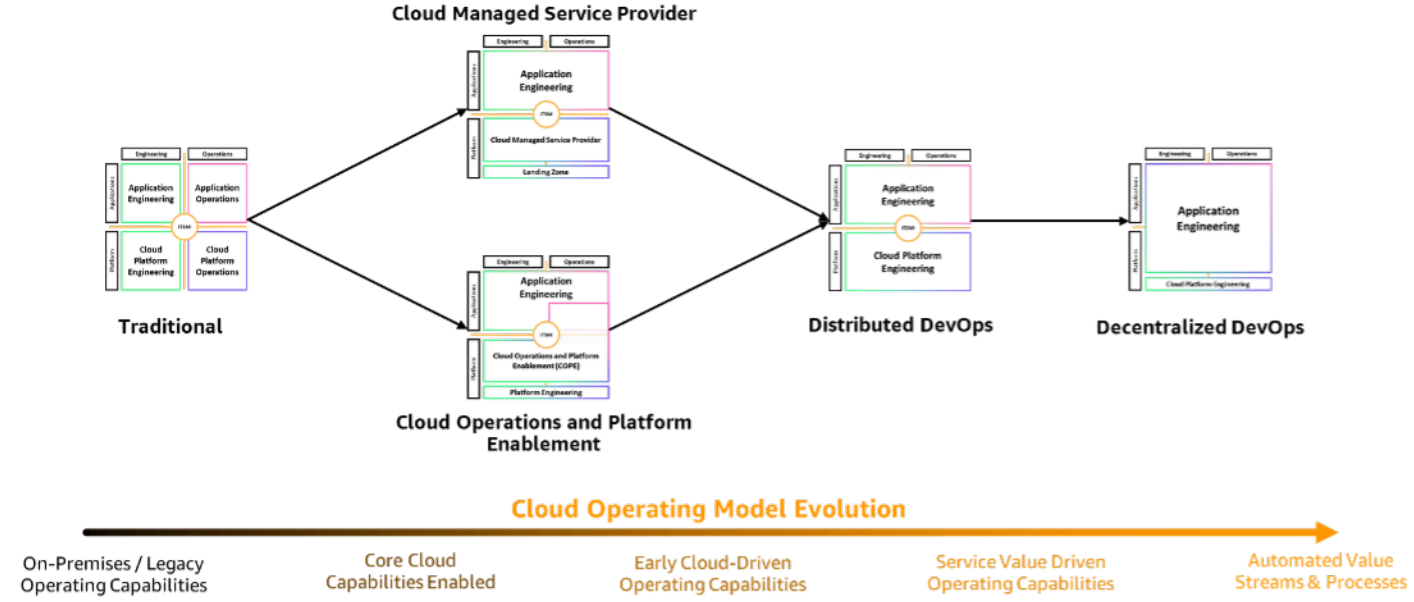
Terapkan kebijakan yang tidak didelegasikan secara khusus ke tim aplikasi. Gunakan alat atau layanan yang memungkinkan Anda untuk mengelola lingkungan di seluruh akun Anda secara tersentralisasi, seperti AWS Organizations. Layanan seperti AWS Control Tower akan memperluas kemampuan manajemen ini dengan membantu Anda menentukan cetak biru (yang mendukung model operasi Anda) untuk penyiapan akun, menerapkan tata kelola berkelanjutan menggunakan AWS Organizations, dan mengotomatiskan penyediaan akun baru.

Tim aplikasi akan diuntungkan dengan memiliki mekanisme untuk meminta penambahan dan perubahan standar. Mereka dapat memberikan kontribusi berupa standar baru yang dapat memberikan manfaat bagi tim aplikasi lain. Tim platform dapat memutuskan bahwa memberikan dukungan langsung untuk kemampuan tambahan ini merupakan dukungan yang efektif untuk hasil bisnis.

Model ini meminimalkan kendala inovasi dengan keterampilan yang signifikan dan persyaratan anggota tim. Ini mengatasi banyak hambatan dan penundaan yang disebabkan oleh transisi tugas antar tim sambil tetap mempromosikan pengembangan hubungan yang efektif antara tim dan pelanggan.

Mengembangkan model operasi Anda

Model-model yang disediakan secara progresif terus bergerak menuju semakin banyaknya otonomi yang dimilikinya pada tingkat beban kerja, sesuai dengan prinsip tim dua pizza. Anda harus memahami bahwa perjalanan ini adalah perjalanan dari pendekatan tradisional ke pendekatan DevOps terdesentralisasi (sebagai dasar untuk evolusi berkelanjutan ke model tim dua pizza), dan ini kemungkinan akan memakan waktu dan membutuhkan kematangan pembangunan di sejumlah kemampuan. Oleh karena itu, kami telah memberikan contoh bagaimana Anda dapat bertransisi antar model saat tim dan organisasi Anda bergerak di sepanjang perjalanan transformasi perusahaan. Dalam setiap perubahan atau model, Anda berkembang menuju tim yang lebih otonom, tetapi masih selaras secara organisasi.



Evolusi model operasi cloud

Saat melakukan evaluasi tentang bagaimana tim Anda dapat mendukung evolusi organisasi, pelajari kompensasi antara model, posisi tim individu Anda dalam model (sekarang dan setelah evolusi), bagaimana hubungan dan kemampuan tim akan berubah, dan apakah manfaat-manfaat yang didapatkan sesuai dengan dampak pada organisasi. Perlu diingat bahwa perubahan tidak pernah linier. Beberapa model lebih sesuai untuk kasus penggunaan atau titik tertentu dalam perjalanan, dan beberapa model ini mungkin memberikan keunggulan dibandingkan dengan yang ada di lingkungan Anda.

Hubungan dan kepemilikan

Model operasi menentukan hubungan antar-tim dan mendukung kepemilikan serta tanggung jawab yang dapat diidentifikasi.

Praktik terbaik

- OPS02-BP01 Sumber daya telah mengidentifikasi pemilik
- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi
- OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas performanya
- OPS02-BP04 Mekanisme tersedia untuk mengelola tanggung jawab dan kepemilikan
- OPS02-BP05 Mekanisme tersedia untuk meminta penambahan, perubahan, dan pengecualian
- OPS02-BP06 Tanggung jawab antara tim telah dinegosiasikan atau ditetapkan sebelumnya

OPS02-BP01 Sumber daya telah mengidentifikasi pemilik

Sumber daya untuk beban kerja Anda harus memiliki pemilik yang teridentifikasi untuk pengontrolan perubahan, penyelesaian masalah, dan fungsi-fungsi lainnya. Pemilik ditetapkan untuk beban kerja, akun, infrastruktur, platform, dan aplikasi. Kepemilikan dicatat menggunakan alat seperti daftar sentral atau metadata yang dilampirkan ke sumber daya. Nilai bisnis komponen menginformasikan proses dan prosedur yang diterapkan kepadanya.

Hasil yang diinginkan:

- Sumber daya telah mengidentifikasi pemilik dengan menggunakan metadata atau daftar sentral.
- Anggota tim dapat mengidentifikasi siapa pemilik sumber daya.
- Akun memiliki satu pemilik apabila mungkin.

Anti-pola umum:

- Kontak alternatif untuk Akun AWS Anda tidak diisi.
- Sumber daya tidak memiliki tag yang mengidentifikasi tim mana yang memilikinya.
- Anda memiliki ITSM antrian tanpa pemetaan email.
- Dua tim sama-sama merupakan pemilik bagian penting dari infrastruktur.

Manfaat menjalankan praktik terbaik ini:

- Kontrol perubahan untuk sumber daya akan menjadi mudah dilakukan dengan ditetapkannya kepemilikan.
- Anda dapat melibatkan pemilik yang benar ketika menyelesaikan masalah.

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

Panduan implementasi

Tentukan pentingnya kepemilikan untuk kasus penggunaan sumber daya di lingkungan Anda. Kepemilikan dapat berarti siapa yang mengawasi perubahan pada sumber daya, yang mendukung sumber daya selama penyelesaian masalah, atau siapa yang bertanggung jawab terhadapnya secara finansial. Sebutkan dan catat pemilik untuk sumber daya, termasuk nama, informasi kontak, organisasi, serta tim.

Contoh pelanggan

AnyCompany Ritel mendefinisikan kepemilikan sebagai tim atau individu yang memiliki perubahan dan dukungan untuk sumber daya. Mereka memanfaatkan AWS Organizations untuk mengelola mereka Akun AWS. Kontak akun alternatif dikonfigurasi menggunakan kotak masuk grup. Setiap ITSM antrian memetakan ke alias email. Tag mengidentifikasi siapa yang memiliki AWS sumber daya. Untuk infrastruktur dan platform-platform lainnya, mereka memiliki halaman wiki yang mengidentifikasi kepemilikan dan informasi kontak.

Langkah-langkah implementasi

1. Mulai dengan menentukan kepemilikan untuk organisasi Anda. Kepemilikan dapat menyiratkan siapa yang memiliki risiko untuk sumber daya, siapa yang memiliki perubahan pada sumber daya, atau siapa yang mendukung sumber daya ketika melakukan penyelesaian masalah. Kepemilikan juga dapat menyiratkan kepemilikan sumber daya secara finansial atau administratif.
2. Gunakan AWS Organizations untuk mengelola akun. Anda dapat mengelola kontak alternatif untuk akun Anda secara terpusat.
   a. Penggunaan alamat email dan nomor telepon milik perusahaan untuk informasi kontak akan membantu Anda untuk mengaksesnya meskipun orang yang memilikinya sudah tidak bekerja di organisasi Anda. Misalnya, buatlah daftar distribusi email terpisah untuk penagihan, operasional, dan keamanan lalu konfigurasikan ketiganya sebagai kontak Penagihan, Keamanan, dan Operasional di masing-masing Akun AWS yang aktif. Beberapa orang akan

menerima AWS pemberitahuan dan dapat merespons, bahkan jika seseorang sedang berlibur, mengubah peran, atau meninggalkan perusahaan.

   b. Jika akun tidak dikelola oleh AWS Organizations, kontak akun alternatif dapat membantu AWS menghubungi personel yang tepat jika diperlukan. Konfigurasikan kontak alternatif akun sehingga menunjuk ke sebuah grup, bukan ke individu perseorangan.

3. Gunakan tag untuk mengidentifikasi pemilik AWS sumber daya. Anda dapat menentukan pemilik maupun informasi kontak mereka dalam tag terpisah.
   a. Anda dapat menggunakan aturan AWS Config untuk menegaskan bahwa sumber daya memiliki tag kepemilikan yang diperlukan.
   b. Untuk panduan mendalam tentang cara membangun strategi pemberian tag untuk organisasi Anda, silakan lihat laporan resmi mengenai Praktik Terbaik Pemberian Tag AWS.
4. Gunakan Amazon Q Business, sebuah asisten percakapan yang menggunakan AI generatif untuk meningkatkan produktivitas tenaga kerja, menjawab pertanyaan, dan menyelesaikan tugas berdasarkan informasi dalam sistem perusahaan Anda.
   a. Hubungkan Amazon Q Business ke sumber data perusahaan Anda. Amazon Q Business menawarkan konektor bawaan ke lebih dari 40 sumber data yang didukung, termasuk Amazon Simple Storage Service (Amazon S3), SharePoint Microsoft, Salesforce, dan Atlassian Confluence. Untuk informasi selengkapnya, silakan lihat Konektor Amazon Q.
5. Untuk sumber daya, platform, dan infrastruktur lainnya, buatlah dokumentasi yang mengidentifikasi kepemilikan. Dokumentasi ini harus dapat diakses oleh semua anggota tim.

Tingkat upaya untuk rencana implementasi: Rendah. Manfaatkan informasi kontak akun dan tag untuk menetapkan kepemilikan AWS sumber daya. Untuk sumber daya lain, Anda dapat menggunakan sesuatu yang sederhana seperti tabel di wiki untuk mencatat kepemilikan dan informasi kontak, atau menggunakan ITSM alat untuk memetakan kepemilikan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP02 Proses dan prosedur telah mengidentifikasi pemilik
- OPS02-BP04 Mekanisme ada untuk mengelola tanggung jawab dan kepemilikan

Dokumen terkait:

- Manajemen Akun AWS - Memperbarui informasi kontak

- AWS Organizations - Memperbarui kontak alternatif di organisasi Anda
- Laporan resmi Praktik Terbaik Pemberian Tag AWS
- Bangun aplikasi AI generatif perusahaan pribadi dan aman dengan Amazon Q Business and AWS IAM Identity Center
- Amazon Q Business, sekarang tersedia secara umum dan dapat membantu Anda meningkatkan produktivitas tenaga kerja dengan AI generatif
- AWS Cloud Blog Operasi & Migrasi - Menerapkan kontrol penandaan otomatis dan terpusat dengan dan AWS ConfigAWS Organizations
- AWS Blog Keamanan - Perpanjang kait pra-komit Anda dengan AWS CloudFormation Guard
- AWS DevOps Blog - Mengintegrasikan AWS CloudFormation Guard ke dalam pipa CI/CD

Lokakarya terkait:

- Lokakarya - Pemberian Tag AWS

Contoh terkait:

- Aturan AWS Config - Amazon EC2 dengan tag yang diperlukan dan nilai yang valid

Layanan terkait:

- Aturan AWS Config - tag yang dibutuhkan
- AWS Organizations

OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi

Pahami siapa pemegang kepemilikan atas definisi dari masing-masing proses dan prosedur, alasan prosedur dan proses tertentu digunakan, serta alasan adanya kepemilikan tersebut. Dengan memahami alasan untuk menggunakan proses dan prosedur tertentu, peluang pengembangan dapat lebih mudah diidentifikasi.

Hasil yang diinginkan: Organisasi Anda memiliki serangkaian proses dan prosedur yang terdefinisi dengan baik dan terpelihara untuk tugas-tugas operasional. Proses dan prosedur-prosedur tersebut disimpan di lokasi terpusat dan tersedia untuk anggota tim Anda. Proses dan prosedur-prosedur sering diperbarui, dengan kepemilikan yang ditetapkan dengan jelas. Jika memungkinkan, skrip, templat, dan dokumen otomatisasi diimplementasikan sebagai kode.

Anti-pola umum:

- Proses tidak didokumentasikan. Mungkin terdapat skrip yang terfragmentasi di stasiun kerja operator yang terisolasi.
- Pengetahuan tentang cara menggunakan skrip dipegang oleh beberapa individu perorangan atau secara informal sebagai pengetahuan tim.
- Proses warisan sudah harus diperbarui, tetapi kepemilikan pembaruan masih tidak jelas, dan penulis aslinya sudah bukan bagian dari organisasi.
- Proses dan skrip tidak dapat ditemukan, sehingga tidak tersedia saat diperlukan (misalnya, dalam merespons sebuah insiden).

Manfaat menjalankan praktik terbaik ini:

- Proses dan prosedur-prosedur akan meningkatkan upaya Anda untuk mengoperasikan beban kerja Anda.
- Anggota tim baru menjadi lebih efektif dengan lebih cepat.
- Mengurangi waktu mitigasi insiden.
- Anggota tim (dan tim) yang berbeda dapat menggunakan proses dan prosedur yang sama secara konsisten.
- Tim dapat menskalakan proses mereka dengan proses yang dapat diulang.
- Proses dan prosedur standar membantu mengurangi dampak pengalihan tanggung jawab beban kerja antar-tim.

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

Panduan implementasi

- Proses dan prosedur-prosedur memiliki pemilik yang jelas untuk bertanggung jawab atas penetapannya.
  - Identifikasi aktivitas operasi yang dijalankan untuk mendukung beban kerja Anda. Buatlah dokumentasi dari aktivitas ini di lokasi yang mudah ditemukan.
  - Identifikasi secara khusus individu atau tim yang bertanggung jawab atas spesifikasi dari sebuah aktivitas. Mereka bertanggung jawab untuk memverifikasi bahwa aktivitas dapat dijalankan dengan sukses oleh anggota tim yang memiliki keterampilan memadai serta memiliki izin, akses, serta alat yang sesuai. Jika terdapat masalah saat menjalankan aktivitas tersebut, maka anggota

tim yang menjalankannya bertanggung jawab untuk memberikan umpan balik mendetail yang diperlukan agar aktivitas tersebut dapat ditingkatkan.

- Rekam kepemilikan dalam metadata artefak aktivitas melalui layanan seperti AWS Systems Manager, melalui dokumen, dan AWS Lambda. Rekam kepemilikan sumber daya menggunakan grup sumber daya atau tag, dengan menentukan informasi kontak dan kepemilikan. Gunakan AWS Organizations untuk membuat kebijakan penandaan dan merekam informasi kontak serta kepemilikan.

- Seiring waktu, prosedur ini harus dikembangkan agar dapat dijalankan sebagai kode, sehingga mengurangi kebutuhan akan campur tangan manusia.
  - Misalnya, pertimbangkan fungsi AWS Lambda, templat CloudFormation, atau dokumen AWS Systems Manager Automation.
  - Jalankan kontrol versi di repositori yang sesuai.
  - Sertakan tanda sumber daya yang sesuai sehingga pemilik dan dokumentasi dapat diidentifikasi dengan mudah.

Contoh pelanggan

AnyCompany Retail mendefinisikan kepemilikan sebagai tim atau individu perseorangan yang memiliki proses untuk suatu aplikasi atau kelompok aplikasi (yang memiliki teknologi dan praktik arsitektur yang sama). Awalnya, proses dan prosedur didokumentasikan dalam bentuk panduan langkah demi langkah di dalam sistem manajemen dokumen, yang dapat ditemukan menggunakan tag pada Akun AWS yang meng-host aplikasi dan di kelompok sumber daya tertentu yang ada di dalam akun. Mereka memanfaatkan AWS Organizations untuk mengelola Akun AWS mereka. Seiring waktu, proses-proses tersebut dikonversi menjadi kode, dan sumber daya didefinisikan dengan menggunakan infrastruktur sebagai kode (seperti templat CloudFormation atau AWS Cloud Development Kit (AWS CDK)). Proses operasional menjadi dokumen otomatisasi di dalam AWS Systems Manager atau fungsi AWS Lambda, yang dapat diaktifkan sebagai tugas terjadwal, sebagai respons terhadap peristiwa-peristiwa seperti alarm CloudWatch AWS atau peristiwa AWS EventBridge, atau diaktifkan berdasarkan permintaan di dalam platform manajemen layanan TI (ITSM). Semua proses memiliki tanda untuk mengidentifikasi kepemilikan. Dokumentasi untuk otomatisasi dan proses dipertahankan di halaman wiki yang dihasilkan oleh repositori kode untuk proses tersebut.

Langkah-langkah implementasi

1. Dokumentasikan proses dan prosedur yang ada.

   a. Tinjau dan terus perbarui proses dan prosedur tersebut.
   b. Identifikasi pemilik untuk setiap proses atau prosedur.
   c. Tempatkan mereka di bawah kontrol versi.
   d. Jika memungkinkan, bagikan proses dan prosedur-prosedur itu di seluruh beban kerja dan lingkungan yang berbagi desain arsitektur.
2. Buat mekanisme untuk umpan balik dan perbaikan.
   a. Tentukan kebijakan untuk frekuensi peninjauan proses.
   b. Tentukan proses untuk peninjau dan pemberi persetujuan.
   c. Implementasikan permasalahan-permasalahan atau antrean tiket untuk umpan balik yang akan diberikan dan dilacak.
   d. Jika memungkinkan, proses dan prosedur harus memiliki klasifikasi risiko dan persetujuan di awal dari dewan persetujuan perubahan (CAB).
3. Verifikasi bahwa proses dan prosedur dapat diakses dan ditemukan oleh orang-orang yang perlu menjalankannya.
   a. Gunakan tanda untuk menunjukkan di mana proses dan prosedur dapat diakses untuk beban kerja.
   b. Gunakan pesan kesalahan dan peristiwa yang dapat dipahami untuk menunjukkan proses atau prosedur yang sesuai untuk mengatasi sebuah permasalahan.
   c. Gunakan wiki dan manajemen dokumen, dan jadikan proses dan prosedur dapat dicari secara konsisten di seluruh organisasi.
4. Gunakan Amazon Q Business, sebuah asisten percakapan yang menggunakan AI generatif untuk meningkatkan produktivitas tenaga kerja, menjawab pertanyaan, dan menyelesaikan tugas berdasarkan informasi dalam sistem perusahaan Anda.
   a. Hubungkan Amazon Q Business ke sumber data perusahaan Anda. Amazon Q Business menawarkan konektor-konektor bawaan ke lebih dari 40 sumber data yang didukung, termasuk Amazon S3, Microsoft SharePoint, Salesforce, dan Atlassian Confluence. Untuk informasi selengkapnya, silakan lihat Konektor Amazon Q.
5. Lakukan otomatisasi jika perlu.
   a. Otomatisasi harus dikembangkan ketika layanan dan teknologi menyediakan API.
   b. Berikan edukasi secara memadai tentang proses. Kembangkan kisah dan persyaratan pengguna untuk mengotomatiskan proses-proses tersebut.
   c. Ukur penggunaan proses dan prosedur Anda dengan sukses, dan laporkan masalah atau ajukan tiket untuk mendukung perbaikan berulang.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP01 Sumber daya memiliki pemilik teridentifikasi
- OPS02-BP04 Mekanisme tersedia untuk mengelola tanggung jawab dan kepemilikan
- OPS11-BP04 Menjalankan manajemen pengetahuan

Dokumen terkait:

- AWS Laporan resmi - Pengantar DevOps di AWS
- Laporan resmi AWS - Praktik Terbaik Penandaan Sumber Daya AWS
- Laporan resmi AWS - Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun
- Blog Operasi dan Migrasi AWS Cloud - Menggunakan Amazon Q Business untuk merampingkan operasi Anda
- AWS Cloud Blog Operasi & Migrasi - Membangun Praktik Otomatisasi Cloud untuk Keunggulan Operasional: Praktik Terbaik dari AWS Managed Services
- Blog Operasi & Migrasi AWS Cloud - Menerapkan kontrol penandaan otomatis dan tersentralisasi dengan AWS Config dan AWS Organizations
- Blog Keamanan AWS - Perpanjang hook pra-commit Anda dengan AWS CloudFormation Guard
- Blog DevOps AWS - Mengintegrasikan AWS CloudFormation Guard ke dalam pipeline CI/CD

Lokakarya terkait:

- AWS Lokakarya Keunggulan Operasional Well-Architected
- Lokakarya - Penandaan AWS

Video terkait:

- Cara melakukan otomatisasi Operasi IT di AWS
- AWS re:Invent 2020 - Melakukan otomatisasi atas apa pun dengan AWS Systems Manager
- AWS re:Inforce 2022 - Melakukan otomatisasi manajemen dan kepatuhan patch dengan menggunakan AWS (NIS306)

- Dukungans You - Memahami Lebih Dalam AWS Systems Manager

Layanan terkait:

- AWS Systems Manager - Otomatisasi
- AWS Service Management Connector

OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas performanya

Pahami siapa yang bertanggung jawab untuk menjalankan aktivitas tertentu terhadap beban kerja yang ditentukan serta alasan adanya tanggung jawab tersebut. Memahami siapa yang bertanggung jawab untuk menjalankan aktivitas dapat memberikan informasi tentang siapa yang akan melakukan aktivitas tersebut, memvalidasi hasilnya, serta memberikan umpan balik kepada pemilik aktivitas.

Hasil yang diinginkan:

Organisasi Anda secara jelas menetapkan tanggung jawab untuk menjalankan aktivitas tertentu pada beban kerja yang ditentukan dan merespons peristiwa-peristiwa yang dihasilkan oleh beban kerja tersebut. Organisasi mendokumentasikan kepemilikan proses dan pemenuhan dan membuat informasi ini dapat ditemukan. Anda meninjau dan memperbarui tanggung jawab ketika ada perubahan yang terjadi pada organisasi, dan tim melacak serta mengukur performa aktivitas identifikasi kekurangan dan inefisiensi. Anda mengimplementasikan mekanisme umpan balik untuk melacak kekurangan dan perbaikan serta mendukung perbaikan berulang.

Anti-pola umum:

- Anda tidak mendokumentasikan tanggung jawab.
- Terdapat skrip yang terfragmentasi di stasiun kerja operator yang terisolasi. Hanya sedikit orang saja yang tahu cara menggunakannya atau secara informal menyebutnya sebagai pengetahuan tim.
- Proses warisan sudah harus diperbarui, tetapi tidak ada yang tahu siapa yang memiliki proses tersebut, dan penulis aslinya sudah tidak lagi menjadi bagian dari organisasi.
- Proses dan skrip tidak dapat ditemukan, dan tidak tersedia saat diperlukan (misalnya, saat merespons insiden).

Manfaat menjalankan praktik terbaik ini:

- Anda memahami siapa yang bertanggung jawab untuk menjalankan sebuah aktivitas, siapa yang harus mendapatkan notifikasi saat diperlukan tindakan, dan siapa yang melakukan tindakan, memvalidasi hasilnya, serta memberikan umpan balik kepada pemilik aktivitas tersebut.
- Proses dan prosedur-prosedur akan meningkatkan upaya Anda untuk mengoperasikan beban kerja Anda.
- Anggota tim baru menjadi lebih efektif dengan lebih cepat.
- Anda mengurangi waktu yang dibutuhkan untuk memitigasi insiden.
- Tim yang berbeda menggunakan proses dan prosedur yang sama untuk melakukan tugas-tugas secara konsisten.
- Tim dapat menskalakan proses mereka dengan proses yang dapat diulang.
- Proses dan prosedur standar membantu memitigasi dampak pengalihan tanggung jawab beban kerja antar-tim.

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

Panduan implementasi

Untuk mulai menentukan tanggung jawab, mulailah dengan dokumentasi yang sudah ada sekarang, seperti matriks tanggung jawab, proses dan prosedur, peran dan tanggung jawab, serta alat-alat dan otomatisasi. Tinjau dan lakukan diskusi mengenai tanggung jawab untuk proses yang terdokumentasi. Lakukan peninjauan bersama tim untuk mengidentifikasi ketidakselarasan antara tanggung jawab dokumen dan proses. Diskusikan layanan-layanan yang ditawarkan dengan pelanggan internal tim tersebut untuk mengidentifikasi perbedaan ekspektasi di antara tim.

Analisis dan atasi perbedaan. Identifikasi setiap peluang perbaikan, dan cari aktivitas padat sumber daya yang sering diminta, yang biasanya merupakan kandidat kuat untuk perbaikan. Jelajahi praktik terbaik, pola, dan panduan preskriptif untuk menyederhanakan dan melakukan standardisasi perbaikan. Dokumentasikan peluang-peluang perbaikan, dan lacak perbaikan hingga selesai.

Seiring waktu, prosedur-prosedur ini harus dikembangkan agar dapat dijalankan sebagai kode, sehingga akan mengurangi kebutuhan akan campur tangan manusia. Misalnya, prosedur-prosedur itu dapat dimulai dalam bentuk fungsi AWS Lambda, templat CloudFormation, atau dokumen Otomatisasi AWS Systems Manager. Pastikan bahwa semua prosedur ini memiliki kontrol versi di repositori yang sesuai, dan sertakan tag sumber daya yang sesuai sehingga tim dapat mengidentifikasi pemilik dan dokumentasi dengan mudah. Buatlah dokumentasi tanggung jawab untuk melaksanakan aktivitas, kemudian pantau otomatisasi untuk inisiasi dan operasi yang berhasil, serta kinerja hasil yang diinginkan.

Contoh pelanggan

AnyCompany Retail mendefinisikan kepemilikan sebagai tim atau individu perseorangan yang memiliki proses untuk suatu aplikasi atau kelompok aplikasi yang memiliki teknologi dan praktik arsitektur yang sama. Awalnya, perusahaan ini mendokumentasikan proses dan prosedur dalam bentuk panduan langkah demi langkah di dalam sistem manajemen dokumen. Mereka membuat prosedur tersebut dapat ditemukan dengan menggunakan tag pada Akun AWS yang meng-host aplikasi dan pada kelompok sumber daya tertentu di dalam akun, dengan menggunakan AWS Organizations untuk mengelola Akun AWS mereka. Seiring waktu, AnyCompany Retail mengonversi proses-proses tersebut menjadi kode dan mendefinisikan sumber daya dengan menggunakan infrastruktur sebagai kode (melalui layanan seperti CloudFormation atau templat AWS Cloud Development Kit (AWS CDK)). Proses operasional menjadi dokumen otomatisasi di dalam fungsi AWS System Manager atau fungsi AWS Lambda, yang dapat dimulai sebagai tugas terjadwal untuk merespons peristiwa seperti alarm Amazon CloudWatch atau peristiwa Amazon EventBridge, atau berdasarkan permintaan di dalam platform manajemen layanan IT (ITSM). Semua proses memiliki tag untuk mengidentifikasi pemiliknya. Tim mengelola dokumentasi untuk otomatisasi dan proses di dalam halaman wiki yang dihasilkan oleh repositori kode untuk proses tersebut.

Langkah-langkah implementasi

1. Dokumentasikan proses dan prosedur yang ada.
    a. Tinjau dan pastikan dokumentasi tersebut mutakhir.
    b. Pastikan bahwa setiap proses atau prosedur mempunyai pemilik.
    c. Tempatkan prosedur di bawah kontrol versi.
    d. Jika memungkinkan, bagikan proses dan prosedur-prosedur itu di seluruh beban kerja dan lingkungan yang berbagi desain arsitektur.
2. Buat mekanisme untuk umpan balik dan perbaikan.
    a. Tentukan kebijakan untuk frekuensi peninjauan proses.
    b. Tentukan proses untuk peninjau dan pemberi persetujuan.
    c. Implementasikan permasalahan-permasalahan atau antrean tiket untuk memberikan dan melacak umpan balik.
    d. Jika memungkinkan, sediakan klasifikasi risiko dan persetujuan awal untuk proses dan prosedur dari dewan persetujuan perubahan (CAB).
3. Buat proses dan prosedur dapat diakses dan ditemukan oleh pengguna yang perlu menjalankannya.

   a. Gunakan tanda untuk menunjukkan di mana proses dan prosedur dapat diakses untuk beban kerja.
   b. Gunakan pesan kesalahan dan peristiwa yang dapat dipahami untuk menunjukkan proses atau prosedur yang sesuai untuk mengatasi masalah.
   c. Gunakan wiki atau manajemen dokumen agar proses dan prosedur dapat dicari secara konsisten di seluruh organisasi.
4. Lakukan otomatisasi jika perlu.
   a. Kembangkan otomatisasi ketika layanan dan teknologi menyediakan API.
   b. Pastikan bahwa proses tersebut dapat dipahami dengan baik, dan kembangkan kisah serta persyaratan pengguna untuk mengotomatiskan proses tersebut.
   c. Ukur penggunaan proses dan prosedur yang sukses, dengan pelacakan masalah untuk mendukung perbaikan berulang.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP01 Sumber daya memiliki pemilik teridentifikasi
- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi
- OPS02-BP04 Mekanisme tersedia untuk mengelola tanggung jawab dan kepemilikan
- OPS02-BP05 Mekanisme tersedia untuk mengidentifikasi tanggung jawab dan kepemilikan
- OPS11-BP04 Menjalankan manajemen pengetahuan

Dokumen terkait:

- Laporan Resmi AWS | Pengantar DevOps di AWS
- Laporan resmi AWS | Praktik Terbaik Pemberian Tag Sumber Daya AWS
- Laporan resmi AWS | Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun
- Blog Operasi & Migrasi AWS Cloud | Membangun Praktik Otomatisasi Cloud untuk Keunggulan Operasional: Praktik Terbaik dari AWS Managed Services
- Lokakarya - Pemberian Tag AWS
- Konektor Manajemen Layanan AWS

Video terkait:

- Pusat Pengetahuan Langsung AWS | Pemberian Tag pada Sumber Daya AWS
- AWS re:Invent 2020 | Melakukan otomatisasi atas apa pun dengan AWS Systems Manager
- AWS re:Inforce 2022 | Melakukan otomatisasi manajemen dan kepatuhan patch dengan menggunakan AWS (NIS306)
- AWS Dukungans You | Diving Deep into AWS Systems Manager

OPS02-BP04 Mekanisme tersedia untuk mengelola tanggung jawab dan kepemilikan

Pahami tanggung jawab yang dimiliki oleh peran Anda dan bagaimana Anda berkontribusi terhadap hasil bisnis, karena pemahaman ini akan mendasari penentuan prioritas tugas Anda dan mengapa peran Anda itu penting. Hal ini akan membantu anggota tim untuk mengenali kebutuhan dan merespons dengan tepat. Ketika anggota tim mengetahui peran mereka, mereka dapat membangun kepemilikan, mengidentifikasi peluang perbaikan, dan memahami cara untuk memengaruhi atau membuat perubahan yang sesuai.

Kadang-kadang, sebuah tanggung jawab mungkin tidak memiliki pemilik yang jelas. Dalam situasi seperti ini, rancang sebuah mekanisme untuk mengatasi kesenjangan ini. Buat jalur eskalasi yang ditentukan dengan baik kepada seseorang yang memiliki wewenang untuk menetapkan kepemilikan atau rencana untuk memenuhi kebutuhan tersebut.

Hasil yang diinginkan: Tim-tim yang ada dalam organisasi Anda memiliki tanggung jawab yang sudah ditentukan dengan jelas yang mencakup bagaimana kaitan mereka dengan sumber daya, tindakan yang harus dilakukan, proses, dan prosedur. Tanggung jawab ini selaras dengan tanggung jawab dan sasaran tim, serta tanggung jawab yang dimiliki tim lain. Anda mendokumentasikan rute eskalasi dengan cara yang konsisten dan dapat ditemukan dan memasukkan keputusan tersebut ke dalam artefak dokumentasi, misalnya matriks tanggung jawab, definisi tim, atau halaman wiki.

Anti-pola umum:

- Tanggung jawab tim yang bersifat ambigu atau tidak ditentukan dengan baik.
- Tim tidak menyelaraskan peran dengan tanggung jawab.
- Tim tidak menyelaraskan tujuan dan sasarannya dengan tanggung jawabnya, sehingga kesuksesan menjadi sulit diukur.
- Tanggung jawab anggota tim tidak selaras dengan tim dan organisasi yang lebih luas.

- Tim Anda tidak memperbarui tanggung jawab mereka, sehingga tanggung jawab tidak konsisten dengan tugas yang dilakukan oleh tim.
- Jalur eskalasi untuk menentukan tanggung jawab tidak ditetapkan atau tidak jelas.
- Jalur eskalasi tidak memiliki pemilik utas tunggal untuk memastikan pemberian respons yang cepat.
- Peran, tanggung jawab, dan jalur eskalasi tidak dapat ditemukan, dan tidak tersedia saat diperlukan (misalnya, saat merespons insiden).

Manfaat menjalankan praktik terbaik ini:

- Ketika Anda memahami siapa yang memegang tanggung jawab atau kepemilikan, Anda dapat menghubungi tim atau anggota tim yang tepat untuk melakukan permintaan atau pengalihann tugas.
- Untuk mengurangi risiko tidak adanya tindakan dan kebutuhan yang tidak tertangani, Anda telah mengidentifikasi seseorang yang memiliki wewenang untuk menetapkan tanggung jawab atau kepemilikan.
- Ketika Anda mendefinisikan cakupan suatu tanggung jawab dengan jelas, anggota tim Anda mendapatkan otonomi dan kepemilikan.
- Tanggung jawab Anda akan mendasari keputusan yang Anda ambil, tindakan yang akan Anda lakukan, dan penyerahan aktivitas Anda ke pemiliknya yang benar.
- Tanggung jawab yang ditinggalkan dapat dengan mudah diidentifikasi karena Anda memiliki pemahaman yang jelas tentang hal-hal yang berada di luar tanggung jawab tim Anda, sehingga membantu Anda dalam melakukan eskalasi untuk meminta klarifikasi.
- Tim menghindari kebingungan dan ketegangan, dan mereka dapat mengelola beban kerja serta sumber daya dengan lebih memadai.

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

Panduan implementasi

Identifikasi peran dan tanggung jawab anggota tim, dan pastikan bahwa mereka memahami apa yang diharapkan dari peran mereka. Buat informasi ini dapat ditemukan sehingga anggota-anggota organisasi Anda dapat mengidentifikasi siapa yang perlu mereka hubungi saat ada kebutuhan khusus, baik berupa tim atau perorangan. Ketika organisasi berusaha memanfaatkan peluang untuk memigrasi dan memodernisasi AWS, peran dan tanggung jawab juga dapat berubah. Jaga agar

tim Anda dan para anggotanya tetap menyadari tanggung jawab mereka, dan latih mereka dengan semestinya untuk melaksanakan tugas selama perubahan ini.

Tentukan peran atau tim yang harus menerima eskalasi untuk mengidentifikasi tanggung jawab dan kepemilikan. Tim ini dapat berinteraksi dengan berbagai pemangku kepentingan untuk mengambil suatu keputusan. Namun, mereka harus memiliki wewenang manajemen proses pengambilan keputusan.

Sediakan mekanisme yang dapat diakses bagi anggota organisasi untuk menemukan dan mengidentifikasi kepemilikan dan tanggung jawab. Mekanisme ini memberi tahu mereka siapa yang harus dihubungi saat ada kebutuhan khusus.

Contoh pelanggan

AnyCompany Retail baru-baru ini menyelesaikan migrasi beban kerja dari lingkungan on-premise ke zona landasan mereka di AWS dengan pendekatan angkat dan geser. Mereka melakukan peninjauan operasi untuk merenungkan cara mereka menyelesaikan tugas-tugas operasional umum dan memastikan bahwa matriks tanggung jawab mereka yang ada sekarang sudah sesuai dengan operasi yang ada di lingkungan baru. Ketika mereka bermigrasi dari on-premise ke AWS, mereka mengurangi tanggung jawab tim infrastruktur yang berkaitan dengan perangkat keras dan infrastruktur fisik. Langkah ini juga mengungkap adanya peluang baru untuk mengembangkan model operasi untuk beban kerja mereka.

Di saat mereka mengidentifikasi, menangani, dan mendokumentasikan sebagian besar tanggung jawab, mereka juga menetapkan rute eskalasi untuk tanggung jawab apa pun yang terlewatkan atau untuk tanggung jawab yang mungkin perlu diubah sesuai perkembangan praktik operasi. Untuk mengeksplorasi peluang-peluang baru untuk melakukan standardisasi dan meningkatkan efisiensi di seluruh beban kerja Anda, berikan akses ke alat-alat operasi seperti AWS Systems Manager dan alat-alat keamanan seperti AWS Security Hub CSPM dan Amazon GuardDuty. AnyCompany Retail menyusun sebuah tinjauan tanggung jawab dan strategi berdasarkan perbaikan yang ingin mereka tangani terlebih dahulu. Ketika perusahaan ini mengadopsi cara-cara kerja dan pola teknologi baru, mereka memperbarui matriks tanggung jawab mereka agar sesuai dengan itu semua.

Langkah-langkah implementasi

1. Mulailah dengan dokumentasi yang sudah ada. Beberapa dokumen sumber yang umum antara lain:
   a. Matriks pertanggungjawaban atau matriks bertanggung jawab, bisa dipertanggungjawabkan, dibuat dengan konsultasi, dan berdasarkan informasi (RACI)

   b. Definisi tim atau halaman wiki
   c. Definisi dan penawaran layanan
   d. Deskripsi peran atau pekerjaan
2. Tinjau dan lakukan diskusi tentang tanggung jawab yang didokumentasikan:
   a. Lakukan peninjauan bersama tim untuk mengidentifikasi ketidakselarasan yang terjadi antara tanggung jawab yang terdokumentasi dan tanggung jawab yang umumnya dijalankan oleh tim.
   b. Diskusikan layanan-layanan potensial yang ditawarkan oleh pelanggan internal untuk mengidentifikasi adanya perbedaan ekspektasi di antara tim.
3. Lakukan analisis dan atasi perbedaan.
4. Identifikasi peluang perbaikan.
   a. Identifikasi permintaan padat sumber daya yang sering kali mendapat permintaan, yang biasanya merupakan kandidat kuat untuk perbaikan.
   b. Cari praktik terbaik, pahami pola, dan ikuti panduan preskriptif, serta lakukan penyederhanaan dan standardisasi perbaikan.
   c. Dokumentasikan peluang-peluang perbaikan, dan lacak hingga perbaikan tersebut selesai.
5. Jika tim belum memiliki tanggung jawab untuk mengelola dan melacak penugasan tanggung jawab, identifikasi seseorang yang ada di dalam tim untuk memegang tanggung jawab ini.
6. Tentukan proses bagi tim untuk meminta klarifikasi tanggung jawab.
   a. Tinjau prosesnya, dan pastikan bahwa proses tersebut jelas dan mudah digunakan.
   b. Pastikan seseorang memiliki dan melacak proses eskalasi hingga selesai.
   c. Buat metrik operasional untuk mengukur efektivitas.
   d. Ciptakan sebuah mekanisme umpan balik untuk memastikan bahwa tim dapat menyoroti peluang-peluang perbaikan.
   e. Implementasikan sebuah mekanisme untuk peninjauan berkala.
7. Buatlah dokumentasi di sebuah lokasi yang dapat ditemukan dan dapat diakses.
   a. Wiki atau portal dokumentasi adalah pilihan umum.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP06 Mengevaluasi kompromi
- OPS03-BP02 Anggota tim diberdayakan untuk bertindak ketika hasil dipertaruhkan
- OPS03-BP03 Tim didorong untuk membawa masalah ke tingkat yang lebih tinggi
- OPS03-BP07 Bekali tim dengan sumber daya dengan sesuai
- OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik
- OPS09-BP03 Meninjau metrik-metrik operasi dan memprioritaskan perbaikan
- OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan

Dokumen terkait:

- Laporan resmi AWS - Pengantar DevOps di AWS
- Laporan Resmi AWS - Kerangka Kerja Adopsi AWS Cloud: Perspektif Operasi
- Keunggulan Operasional Kerangka Kerja AWS Well-Architected - Topologi model operasi Tingkat beban kerja
- Panduan Preskriptif AWS - Membangun Model Operasi Cloud Anda
- Panduan Preskriptif AWS - Membuat matriks RACI atau RASCI untuk model operasi cloud
- Blog Operasi & Migrasi AWS Cloud - Memberikan Nilai Bisnis dengan Tim Platform Cloud
- Blog Operasi & Migrasi AWS Cloud - Mengapa Model Operasi Cloud?
- DevOps Blog AWS - Bagaimana organisasi memodernisasi untuk operasi cloud

Video terkait:

- Summit Online AWS - Model Operasi Cloud untuk Transformasi yang Dipercepat
- re:Invent 2023 AWS - Keamanan cloud yang tahan masa depan: Model operasi baru

OPS02-BP05 Mekanisme tersedia untuk meminta penambahan, perubahan, dan pengecualian

Anda dapat mengajukan permintaan kepada pemilik proses, prosedur, dan sumber daya. Permintaan tersebut mencakup penambahan, perubahan, dan pengecualian. Permintaan ini diajukan melalui sebuah proses manajemen perubahan. Buatlah keputusan-keputusan yang matang berdasarkan informasi untuk menyetujui permintaan apabila memungkinkan dan dianggap tepat setelah dilakukan evaluasi manfaat dan risiko.

Hasil yang diinginkan:

- Anda dapat mengajukan permintaan untuk mengubah proses, prosedur, dan sumber daya berdasarkan kepemilikan yang ditetapkan.
- Perubahan harus dibuat dengan penuh pertimbangan, dengan memikirkan manfaat dan risikonya.

Anti-pola umum:

- Anda harus memperbarui cara Anda melakukan deployment terhadap aplikasi Anda, tetapi perubahan proses deployment tidak dapat diminta dari tim operasi.
- Rencana pemulihan bencana harus diperbarui, tetapi tidak ada pemilik yang diidentifikasi untuk dapat dimintai perubahan.

Manfaat menjalankan praktik terbaik ini:

- Proses, prosedur, dan sumber daya dapat berubah seiring dengan terjadinya perubahan persyaratan.
- Pemilik dapat mengambil keputusan yang matang berdasarkan informasi ketika harus membuat perubahan.
- Perubahan harus dibuat dengan cara yang penuh pertimbangan.

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

Panduan implementasi

Untuk mengimplementasikan praktik terbaik ini, Anda harus dapat membuat permintaan perubahan proses, prosedur, dan sumber daya. Proses manajemen perubahan bisa jadi hal yang ringan. Buatlah dokumentasi proses manajemen perubahan.

Contoh pelanggan

AnyCompany Retail menggunakan sebuah matriks penetapan tanggung jawab (RACI) untuk mengidentifikasi siapa yang memiliki perubahan untuk proses, prosedur, dan sumber daya. Mereka memiliki proses manajemen perubahan terdokumentasi yang ringan dan mudah diikuti. Menggunakan matriks RACI dan proses, siapa pun dapat menyampaikan permintaan perubahan.

Langkah-langkah implementasi

1. Identifikasi proses, prosedur, dan sumber daya untuk beban kerja Anda dan pemilik untuk masing-masing. Buatlah dokumentasi tentang itu semua dalam sistem manajemen pengetahuan Anda.
   a. Jika Anda belum menerapkan OPS02-BP01 Sumber daya telah mengidentifikasi pemilik, OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi, atau OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas performanya, mulailah dengan yang pertama.
2. Bekerjasamalah dengan para pemangku kepentingan yang ada di organisasi Anda untuk mengembangkan sebuah proses manajemen perubahan. Proses tersebut harus meliputi penambahan, perubahan, dan pengecualian untuk sumber daya, proses, dan prosedur.
   a. Anda dapat menggunakan Manajer Perubahan AWS Systems Manager sebagai sebuah platform manajemen perubahan untuk sumber daya beban kerja.
3. Buatlah dokumentasi proses manajemen perubahan dalam sistem manajemen pengetahuan Anda.

Tingkat upaya untuk rencana implementasi: Sedang. Mengembangkan sebuah proses manajemen perubahan memerlukan penyelarasan dengan banyak pemangku kepentingan yang ada di seluruh organisasi Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP01 Sumber daya telah mengidentifikasi pemilik - Sumber daya membutuhkan pemilik yang teridentifikasi sebelum Anda membangun sebuah proses manajemen perubahan.
- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi - Proses membutuhkan pemilik yang teridentifikasi sebelum Anda membangun sebuah proses manajemen perubahan.
- OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas performanya - Kegiatan Operasi membutuhkan pemilik yang teridentifikasi sebelum Anda membangun sebuah proses manajemen perubahan.

Dokumen terkait:

- Panduan Preskriptif AWS - Playbook landasan dasar untuk migrasi besar AWS: Membuat matriks RACI
- Manajemen Perubahan dalam Laporan Resmi Cloud

Layanan terkait:

- Manajer Perubahan AWS Systems Manager

OPS02-BP06 Tanggung jawab antara tim telah dinegosiasikan atau ditetapkan sebelumnya

Miliki perjanjian yang telah ditetapkan atau dinegosiasikan antara tim yang menjelaskan bagaimana mereka akan bekerja sama dan saling mendukung satu sama lain (contohnya, waktu respons, tujuan tingkat layanan, atau perjanjian tingkat layanan). Saluran komunikasi antar-tim didokumentasikan. Memahami dampak dari pekerjaan tim terhadap hasil bisnis, dan hasil dari tim dan organisasi yang lain akan membuat Anda tahu tentang penentuan prioritas tugas mereka dan membantu mereka merespons dengan tepat.

Ketika ada tanggung jawab dan kepemilikan yang tidak ditetapkan atau tidak diketahui, maka Anda akan menanggung risiko tidak menangani aktivitas yang diperlukan secara tepat waktu serta risiko munculnya upaya yang berulang dan kemungkinan bertentangan untuk menangani kebutuhan-kebutuhan tersebut.

Hasil yang diinginkan:

- Perjanjian bekerja atau mendukung antar-tim sudah disetujui dan didokumentasikan.
- Tim-tim yang mendukung atau bekerja dengan satu sama lain memiliki ekspektasi respons dan saluran komunikasi yang telah ditetapkan sebelumnya.

Anti-pola umum:

- Ada sebuah masalah yang terjadi dalam produksi dan dua tim terpisah mulai menyelesaikan masalahnya sendiri-sendiri. Upaya terpisah mereka memperpanjang masa henti produksi.
- Tim operasi membutuhkan bantuan dari tim pengembangan, tetapi tidak ada waktu respons yang disepakati. Permintaannya tetap berada dalam timbunan yang belum dikerjakan (backlog).

Manfaat menjalankan praktik terbaik ini:

- Tim mengetahui cara berinteraksi dan mendukung satu sama lain.
- Ekspektasi untuk tingkat responsivitas sudah diketahui.
- Saluran komunikasi sudah ditetapkan dengan jelas.

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

Panduan implementasi

Dengan mengimplementasikan praktik terbaik ini, artinya bahwa sudah tidak ada lagi ambiguitas tentang bagaimana tim bekerja dengan satu sama lain. Perjanjian resmi mengatur tentang bagaimana tim bekerja sama atau mendukung satu sama lain. Saluran komunikasi antar-tim sudah didokumentasikan.

Contoh pelanggan

Tim SRE AnyCompany Retail memiliki perjanjian tingkat layanan dengan tim pengembangan mereka. Setiap kali tim pengembangan mengajukan sebuah permintaan dalam sistem tiket mereka, mereka dapat mengantisipasi bahwa respons akan diterima dalam waktu lima belas menit. Jika ada penghentian kerja di lokasi, tim SRE akan memimpin pelaksanaan investigasi dengan dukungan dari tim pengembangan.

Langkah-langkah implementasi

1. Melalui kerja sama dengan para pemangku kepentingan yang ada di seluruh organisasi Anda, buatlah perjanjian antara tim berdasarkan proses dan prosedur.
   a. Jika sebuah proses atau prosedur dimiliki bersama antara dua tim, kembangkan sebuah runbook tentang cara tim akan bekerja sama.
   b. Jika ada ketergantungan antara tim, buat dan sepakati SLA respons untuk permintaan.
2. Buatlah dokumentasi tanggung jawab dalam sistem manajemen pengetahuan Anda.

Tingkat upaya untuk rencana implementasi: Sedang. Jika belum ada perjanjian yang dibuat antara tim, mungkin akan diperlukan upaya agar para pemangku kepentingan di seluruh organisasi Anda bisa sepakat.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi - Kepemilikan proses harus diidentifikasi sebelum menetapkan perjanjian antar tim.
- OPS02-BP03 Aktivitas operasi memiliki pemilik teridentifikasi yang bertanggung jawab atas performanya - Kepemilikan kegiatan operasi harus diidentifikasi sebelum menetapkan perjanjian antar tim.

Dokumen terkait:

- Wawasan Eksekutif AWS - Memberdayakan Inovasi dengan Tim Dua Pizza
- Pengantar DevOps di AWS - Tim Dua Pizza

Budaya organisasi

Berikan dukungan kepada anggota tim Anda sehingga mereka dapat menjadi lebih efektif dalam mengambil tindakan dan mendukung hasil bisnis Anda.

Praktik terbaik

- OPS03-BP01 Memberikan sponsor eksekutif
- OPS03-BP02 Anggota tim diberdayakan untuk bertindak ketika hasil dipertaruhkan
- OPS03-BP03 Eskalasi didorong
- OPS03-BP04 Komunikasi yang tepat waktu, jelas, dan dapat ditindaklanjuti
- OPS03-BP05 Mendorong eksperimen
- OPS03-BP06 Mendorong dan mendukung para anggota tim untuk mempertahankan dan mengembangkan tingkat keterampilan mereka
- OPS03-BP07 Tim sumber daya dengan tepat

OPS03-BP01 Memberikan sponsor eksekutif

Pada tingkat tertinggi, pimpinan senior bertindak sebagai sponsor eksekutif untuk menetapkan ekspektasi dan arah untuk hasil organisasi dengan jelas, termasuk mengevaluasi keberhasilannya. Sponsor mendukung dan mendorong penggunaan praktik-praktik terbaik serta perkembangan organisasi.

Hasil yang diinginkan: Organisasi-organisasi yang berusaha untuk mengadopsi, mentransformasi, dan mengoptimalkan operasi cloud mereka menetapkan garis kepemimpinan dan akuntabilitas yang jelas untuk hasil yang diinginkan. Organisasi memahami setiap kemampuan yang dibutuhkan oleh organisasi untuk mencapai hasil baru dan menetapkan kepemilikan kepada tim fungsional untuk pengembangan. Pimpinan secara aktif menetapkan arah ini, menetapkan kepemilikan, mengambil pertanggungjawaban, dan menentukan pekerjaan. Hasilnya, individu yang ada di seluruh organisasi dapat termobilisasi, merasa terinspirasi, dan secara aktif bekerja menuju tujuan yang diinginkan.

Anti-pola umum:

- Terdapat sebuah mandat bagi pemilik beban kerja untuk memigrasikan beban kerja AWS tanpa sponsor yang jelas dan membuat rencana untuk operasi cloud. Hal ini mengakibatkan tim berkolaborasi untuk melakukan peningkatan dan mematangkan kemampuan operasional tanpa ada kesadaran. Kurangnya standar praktik terbaik operasional membuat tim menjadi kewalahan (seperti kerja keras operator, kondisi selalu siaga, dan utang teknis), yang membatasi inovasi.
- Sasaran organisasi baru telah ditetapkan untuk mengadopsi sebuah teknologi baru tanpa memberikan sponsor dan strategi kepemimpinan. Tim menafsirkan tujuan-tujuan secara berbeda, yang menyebabkan kebingungan di antara mereka tentang ke mana upaya harus difokuskan, alasan mengapa tujuan-tujuan itu penting, dan bagaimana dampaknya harus diukur. Akibatnya, organisasi kehilangan momentum dalam mengadopsi teknologi.

Manfaat menerapkan praktik terbaik ini: Ketika sponsor eksekutif secara jelas mengomunikasikan dan berbagi visi, arah, dan tujuan yang hendak dicapainya, anggota tim akan tahu apa yang diharapkan dari mereka. Individu dan tim mulai memfokuskan upaya secara intens ke arah yang sama untuk mencapai tujuan-tujuan yang sudah ditentukan ketika para pemimpin terlibat secara aktif. Hasilnya, organisasi memaksimalkan kemampuan untuk mencapai keberhasilan. Ketika Anda melakukan evaluasi atas kesuksesan, Anda dapat mengidentifikasi hambatan dengan lebih baik sehingga hambatan tersebut dapat diatasi melalui campur tangan sponsor eksekutif.

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

Panduan implementasi

- Pada setiap fase perjalanan cloud (migrasi, adopsi, atau optimalisasi), kesuksesan membutuhkan keterlibatan aktif di tingkat kepemimpinan tertinggi dengan sponsor eksekutif yang ditunjuk. Sponsor eksekutif menyelaraskan pola pikir tim, serangkaian keahlian, dan cara bekerja dengan strategi yang ditentukan.
  - Jelaskan alasannya: Berikan kejelasan dan uraikan alasan-alasan di balik visi dan strategi.
  - Tetapkan ekspektasi: Tentukan dan publikasikan tujuan-tujuan yang ingin diraih organisasi Anda, termasuk cara mengukur keberhasilannya.
  - Lacak pencapaian tujuan: Ukur pencapaian tujuan bertahap secara teratur (bukan hanya penyelesaian tugasnya saja). Bagikan hasilnya sehingga tindakan yang tepat dapat dilakukan jika hasil sedang dalam risiko.
  - Sediakan sumber daya yang diperlukan untuk mencapai tujuan Anda: Satukan orang-orang dan tim untuk saling berkolaborasi dan membangun solusi yang tepat yang mewujudkan hasil yang ingin dicapai. Hal ini mengurangi atau menghilangkan gesekan organisasi.

- Dukung tim Anda: Tetap berinteraksi dengan tim Anda sehingga Anda memahami bagaimana kondisi mereka dan mengetahui jika ada faktor eksternal yang memengaruhi mereka. Identifikasi rintangan yang memperlambat kemajuan tim Anda. Bertindaklah atas nama tim Anda untuk membantu mengatasi hambatan dan menghilangkan beban yang tidak perlu. Ketika ada faktor eksternal yang memengaruhi performa tim Anda, lakukan evaluasi kembali tujuan dan sesuaikan target sebagaimana mestinya.
- Dorong penerapan praktik terbaik: Kenali praktik terbaik yang memberikan manfaat-manfaat yang dapat diukur, dan kenali para pembuat dan pengguna praktik tersebut. Dukung adopsi lebih lanjut untuk memperbesar manfaat yang dapat dicapai dengannya.
- Dorong evolusi tim Anda: Ciptakan budaya peningkatan terus-menerus, dan secara proaktif belajarlah dari kemajuan yang dibuat, dan belajarlah pula dari kegagalan yang dialami. Dukung pertumbuhan dan perkembangan perorangan maupun organisasi. Gunakan data dan anekdot untuk mengembangkan visi dan strategi.

Contoh pelanggan

AnyCompany Retail sedang dalam proses transformasi bisnis melalui perombakan pengalaman pelanggan dengan cepat, peningkatan produktivitas, dan percepatan pertumbuhan dengan menggunakan AI generatif.

Langkah-langkah implementasi

1. Bangun kepemimpinan yang berutas tunggal (single-threaded), dan tunjuk sponsor eksekutif utama untuk memimpin dan mendorong transformasi.
2. Tentukan hasil bisnis yang jelas dari transformasi Anda, dan tetapkan kepemilikan serta pertanggunjawabannya. Berdayakan pejabat eksekutif utama dengan wewenang untuk memimpin dan mengambil keputusan-keputusan penting.
3. Pastikan bahwa strategi transformasional Anda sudah sangat jelas dan telah dikomunikasikan secara luas oleh sponsor eksekutif ke setiap tingkat organisasi.
   a. Tetapkan tujuan-tujuan bisnis yang jelas untuk inisiatif IT dan cloud.
   b. Buatlah dokumentasi metrik bisnis utama untuk mendorong transformasi IT dan cloud.
   c. Komunikasikan visi secara konsisten kepada semua tim dan individu perorangan yang bertanggung jawab atas bagian-bagian dari strategi tersebut.
4. Kembangkan matriks perencanaan komunikasi yang menentukan pesan apa yang perlu disampaikan kepada pemimpin, manajer, dan kontributor individu perorangan tertentu. Tentukan orang atau tim yang harus menyampaikan pesan ini.

   a. Jalankan rencana komunikasi secara konsisten dan andal.
   b. Tetapkan dan kelola ekspektasi melalui acara tatap muka yang diadakan secara rutin.
   c. Terima umpan balik mengenai efektivitas komunikasi, dan sesuaikan komunikasinya, dan buatlah rencana komunikasi sesuai dengan itu.
   d. Jadwalkan acara-acara komunikasi untuk memahami tantangan dari tim secara proaktif, dan buatlah sebuah loop umpan balik yang konsisten yang akan memungkinkan penyesuaian arah, jika diperlukan.
5. Libatkan secara aktif setiap inisiatif dari perspektif pimpinan untuk memastikan bahwa semua tim yang terkena dampak memahami hasil yang menjadi tanggung jawab mereka.
6. Pada setiap rapat status, sponsor eksekutif harus mencari penghalang, memeriksa metrik yang ditetapkan, anekdot, atau umpan balik dari tim, dan mengukur kemajuan pencapaian tujuan-tujuan yang ditetapkan.

Tingkat upaya untuk rencana implementasi Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS03-BP04 Komunikasi yang tepat waktu, jelas, dan dapat ditindaklanjuti
- OP11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan
- OPS11-BP07 Melakukan peninjauan metrik operasi

Dokumen terkait:

- Mengurai Kekusutan Organisasi Anda: Sangat Selaras
- Transformasi yang Hidup: Perubahan-perubahan dengan pendekatan pragmatis
- Menjadi Perusahaan yang Siap Menghadapi Masa Depan
- 7 Perangkap yang Perlu Dihindari Saat Membangun CCOE
- Menavigasi Cloud: Indikator Performa Utama untuk Mencapai Keberhasilan

Video terkait:

- AWS re: Invent 2023: Panduan pemimpin untuk AI generatif: Menggunakan sejarah untuk membentuk masa depan (SEG204)

Contoh terkait:

- Prosci: Peran & Pentingnya Sponsor Utama

OPS03-BP02 Anggota tim diberdayakan untuk bertindak ketika hasil dipertaruhkan

Perilaku kepemilikan yang membudaya yang ditanamkan oleh pimpinan akan menimbulkan perasaan diberdayakan pada diri karyawan untuk bertindak atas nama seluruh perusahaan di luar cakupan peran dan pertanggungjawaban mereka. Karyawan dapat bertindak untuk mengidentifikasi risiko secara proaktif saat risiko-risiko itu muncul dan melakukan tindakan sebagaimana mestinya. Dengan budaya seperti ini, karyawan dapat mengambil keputusan bernilai tinggi dengan kesadaran terhadap situasi yang dihadapinya.

Misalnya, Amazon menggunakan Prinsip Kepemimpinan sebagai pedoman untuk mendorong perilaku yang diinginkan bagi para karyawan untuk bergerak maju saat menghadapi kesulitan, memecahkan masalah, menangani konflik, dan mengambil tindakan.

Hasil yang diinginkan: Para pimpinan telah menyebarkan pengaruh budaya baru yang memungkinkan individu dan tim membuat keputusan-keputusan penting, bahkan pada tingkat organisasi yang lebih rendah (selama keputusan-keputusan itu ditentukan dengan izin yang dapat diaudit dan mekanisme keselamatan). Kegagalan tidak dianggap sebagai sesuatu yang buruk, dan tim secara berulang belajar untuk memperbaiki cara-cara mereka dalam mengambil keputusan dan dalam memberikan respons untuk mengatasi situasi-situasi serupa di kesempatan lain. Jika tindakan seseorang mengakibatkan terjadinya perbaikan yang dapat menguntungkan tim lain, mereka membagikan pengetahuan yang didapatkan dari tindakan tersebut secara proaktif. Pimpinan mengukur perbaikan operasional dan memberikan insentif kepada individu perseorangan dan organisasi yang mengadopsi pola-pola tersebut.

Anti-pola umum:

- Tidak ada panduan atau mekanisme yang jelas dalam organisasi terkait hal-hal yang harus dilakukan ketika ada risiko yang diidentifikasi. Misalnya, ketika seorang karyawan melihat adanya serangan phishing, lalu karyawan tersebut tidak melapor ke tim keamanan, dan mengakibatkan sebagian besar organisasi menjadi korban dari serangan tersebut. Hal ini mengakibatkan terjadinya pembobolan data.
- Pelanggan Anda mengeluhkan bahwa layanan tidak tersedia, yang terutama disebabkan oleh deployment yang gagal. Tim SRE Anda bertanggung jawab atas alat deployment, dan rollback

otomatis untuk deployment tersedia di dalam peta strategi jangka panjang mereka. Dalam peluncuran (rollout) aplikasi baru-baru ini, salah satu rekayasawan menemukan solusi untuk mengotomatiskan rollback aplikasi ke versi sebelumnya. Meskipun solusi mereka dapat menjadi pola untuk tim SRE, namun tim lain tidak mengadopsinya, karena tidak ada proses untuk melacak perbaikan tersebut. Organisasi terus diganggu dengan deployment yang gagal dan berdampak pada para pelanggan dan menyebabkan sentimen negatif lebih lanjut.

- Untuk menjaga kepatuhan, tim infosec Anda mengawasi proses model lama untuk merotasi kunci SSH bersama secara rutin atas nama operator yang terhubung ke instans Linux Amazon EC2 mereka. Dibutuhkan beberapa hari bagi tim infosec untuk menyelesaikan proses rotasi kunci ini, dan Anda pun tidak dapat terkoneksi ke instans tersebut. Tidak ada seorang pun di dalam atau di luar infosec yang menyarankan penggunaan opsi lain yang ada di AWS untuk mencapai hasil yang sama.

Manfaat menerapkan praktik terbaik ini: Dengan melakukan desentralisasi atas otoritas untuk membuat keputusan dan memberdayakan tim Anda untuk membuat keputusan kunci, Anda dapat mengatasi masalah dengan lebih cepat dan dengan tingkat keberhasilan yang meningkat. Selain itu, tim mulai menyadari adanya rasa kepemilikan, dan kegagalan dapat diterima. Eksperimen menjadi andalan yang membudaya. Manajer dan direktur tidak merasakan bahwa mereka seolah-olah dikontrol secara berlebihan di setiap aspek pekerjaan mereka.

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

Panduan implementasi

1. Kembangkan budaya yang menganggap kegagalan sebagai sebuah hal lumrah.
2. Tetapkan kepemilikan dan pertanggungjawaban yang jelas untuk berbagai area fungsional di dalam organisasi.
3. Komunikasikan kepemilikan dan akuntabilitas kepada semua orang sehingga setiap individu tahu siapa yang dapat membantu mereka memfasilitasi keputusan yang terdesentralisasi.
4. Tentukan keputusan satu arah dan dua arah Anda untuk membantu individu perseorangan mengetahui kapan mereka perlu melakukan eskalasi ke tingkat kepemimpinan yang lebih tinggi.
5. Ciptakan kesadaran organisasi bahwa semua karyawan diberdayakan untuk melakukan tindakan di berbagai tingkatan ketika ada masalah yang membahayakan hasil. Bekali para anggota tim Anda dengan dokumentasi tata kelola, tingkat izin, alat, dan peluang untuk mempraktikkan keterampilan yang diperlukan untuk memberikan respons secara efektif.

6. Berikan kepada para anggota tim Anda peluang untuk melatih keterampilan yang diperlukan untuk memberikan respons atas berbagai keputusan. Setelah tingkat keputusan ditentukan, jalankan game day untuk memastikan bahwa semua kontributor perorangan memahami dan dapat mendemonstrasikan proses.
   a. Sediakan lingkungan alternatif yang aman di mana proses dan prosedur dapat diuji dan dilatih.
   b. Akui dan ciptakan kesadaran bahwa para anggota tim memiliki wewenang untuk melakukan tindakan ketika hasil mengandung risiko pada tingkat yang telah ditentukan.
   c. Tentukan otoritas yang dimiliki anggota tim Anda untuk mengambil tindakan dengan memberikan izin dan akses ke beban kerja dan komponen yang mereka dukung.
7. Berikan kemampuan kepada tim untuk berbagi pembelajaran mereka (baik keberhasilan maupun kegagalan yang berkaitan dengan operasi).
8. Berdayakan tim untuk menantang status quo, dan sediakan mekanisme untuk melacak dan mengukur perbaikan, serta mengukur dampaknya terhadap organisasi.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP06 Mengevaluasi kompromi sambil mengelola manfaat dan risiko
- OPS02-BP05 Mekanisme tersedia untuk mengidentifikasi tanggung jawab dan kepemilikan

Dokumen terkait:

- Posting Blog AWS | Perusahaan yang tangkas
- Posting Blog AWS | Mengukur kesuksesan: Paradoks dan rencana
- Posting Blog AWS | Melepaskan: Mengaktifkan otonomi dalam tim
- Sentralisasi atau Desentralisasi?

Video terkait:

- re:Invent 2023 | Bagaimana agar Anda tidak menyabotase transformasi Anda (SEG201)
- re:Invent 2021 | Amazon Builders' Library: Keunggulan Operasional di Amazon
- Sentralisasi vs Desentralisasi

Contoh terkait:

- Menggunakan catatan keputusan arsitektur untuk merampingkan proses pengambilan keputusan teknis untuk proyek pengembangan perangkat lunak

OPS03-BP03 Eskalasi didorong

Anggota tim diimbau oleh para pimpinan untuk menyampaikan masalah dan kekhawatiran mereka kepada pengambil keputusan dan pemangku kepentingan pada level yang lebih tinggi, jika mereka meyakini bahwa hal itu menimbulkan risiko pada hasil dan standar yang diharapkan tidak terpenuhi. Ini adalah bagian dari budaya organisasi dan didorong pada semua level. Eskalasi (penyampaian) harus dilakukan sejak dini dan sering kali agar risiko-risiko semacam itu dapat diidentifikasi, dan dicegah sebelum menyebabkan insiden. Pimpinan tidak menegur individu perseorangan karena menyampaikan masalah ke level pimpinan yang lebih tinggi.

Hasil yang diinginkan: Individu perseorangan yang ada di seluruh organisasi merasa nyaman untuk menyampaikan masalah ke tingkat pimpinan langsung dan lebih tinggi. Pimpinan telah dengan sengaja dan sadar menetapkan ekspektasi bahwa tim mereka harus merasa aman untuk menyampaikan masalah apa pun ke pimpinan yang lebih tinggi. Terdapat mekanisme untuk menyampaikan masalah ke pimpinan yang lebih tinggi di setiap tingkat dalam organisasi. Ketika karyawan melakukan eskalasi ke manajer mereka, mereka bersama-sama memutuskan tingkat dampaknya dan apakah masalah tersebut harus disampaikan ke tingkat pimpinan yang lebih tinggi. Untuk memulai eskalasi, karyawan diharuskan untuk menyertakan rencana kerja yang direkomendasikan untuk mengatasi masalah tersebut. Jika manajemen langsung tidak melakukan tindakan dengan tepat waktu, maka karyawan diimbau untuk menyampaikan masalah tersebut ke tingkat pimpinan tertinggi jika mereka merasa sangat yakin bahwa risiko terhadap organisasi tersebut benar-benar perlu dieskalasi.

Anti-pola umum:

- Para pemimpin eksekutif tidak mengajukan pertanyaan yang cukup cermat selama pelaksanaan rapat status program transformasi cloud Anda untuk menemukan letak terjadinya masalah dan hambatan. Yang disajikan sebagai status hanyalah kabar baik. Mereka CIO telah menjelaskan bahwa dia hanya suka mendengar kabar baik, karena tantangan apa pun yang muncul membuat CEO orang berpikir bahwa program tersebut gagal.
- Anda adalah seorang rekayasawan operasi cloud dan Anda melihat bahwa sistem manajemen pengetahuan yang baru tidak diadopsi secara luas oleh tim aplikasi. Perusahaan menginvestasikan waktu satu tahun dan dana beberapa juta dolar untuk mengimplementasikan sistem manajemen

pengetahuan baru tersebut, tetapi orang-orang masih menulis runbook mereka secara lokal dan membagikannya di layanan berbagi cloud organisasi, sehingga pengetahuan terkait beban kerja yang didukung itu menjadi sulit ditemukan. Anda mencoba menyampaikan hal ini kepada pimpinan, karena penggunaan sistem ini secara konsisten dapat meningkatkan efisiensi operasional. Ketika Anda menyampaikannya kepada direktur yang memimpin implementasi sistem manajemen pengetahuan tersebut, ia akan menegur Anda karena hal ini dianggap dapat menciptakan keraguan pada investasi.

- Tim infosec yang bertanggung jawab untuk mengeraskan sumber daya komputasi telah memutuskan untuk menerapkan proses yang mengharuskan melakukan pemindaian yang diperlukan untuk memastikan bahwa EC2 instance sepenuhnya diamankan sebelum tim komputasi merilis sumber daya untuk digunakan. Ini telah menciptakan penundaan waktu seminggu tambahan untuk sumber daya yang akan digunakan, yang merusak merekaSLA. Tim komputasi tidak berani menyampaikan masalah ini kepada VP melalui cloud karena hal ini merusak citra VP keamanan informasi.

Manfaat menjalankan praktik terbaik ini:

Masalah-masalah yang kompleks atau kritis ditangani sebelum berdampak pada bisnis. Lebih sedikit waktu yang terbuang. Risiko diminimalkan. Tim menjadi lebih proaktif dan fokus pada hasil ketika memecahkan masalah.

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

Panduan implementasi

Kemauan dan kemampuan untuk melakukan eskalasi secara bebas di setiap tingkatan di dalam organisasi adalah sebuah landasan organisasi dan budaya yang harus dikembangkan secara sadar melalui pelatihan yang ditekankan, komunikasi kepemimpinan, penetapan ekspektasi, dan deployment mekanisme di seluruh organisasi pada setiap tingkat.

Langkah-langkah implementasi

1. Tentukan kebijakan, standar, dan ekspektasi untuk organisasi Anda.
   a. Pastikan adopsi dan pemahaman yang luas tentang kebijakan, ekspektasi, dan standar.
2. Dorong, latih, dan berdayakan pekerja untuk melakukan eskalasi waktu secara dini dan sering kali ketika standar tidak terpenuhi.
3. Akui pada tingkat organisasi bahwa eskalasi yang dilakukan sejak dini dan sering kali merupakan praktik terbaik. Akui bahwa eskalasi mungkin saja terbukti tidak berdasar, tetapi lebih baik

mengambil kesempatan untuk mencegah terjadinya insiden daripada melewatkan kesempatan tersebut dengan tidak melakukan eskalasi.

   a. Bangun sebuah sebuah mekanisme untuk melakukan eskalasi (seperti sistem kabel Andon).
   b. Miliki prosedur terdokumentasi yang menetapkan kapan dan bagaimana eskalasi harus dilakukan.
   c. Tentukan sekelompok personel dengan otoritas berjenjang untuk melakukan atau menyetujui tindakan, beserta informasi kontak dari setiap pemangku kepentingan.

4. Ketika eskalasi terjadi, eskalasi tersebut harus berlanjut sampai anggota tim yakin bahwa risiko telah ditangani melalui tindakan yang didorong dari para pimpinan.
   a. Eskalasi harus mencakup:
      i. Deskripsi situasi, dan sifat risiko
      ii. Tingkat kekritisan situasi
      iii. Siapa atau apa yang terkena dampak
      iv. Seberapa besar dampak tersebut
      v. Urgensi jika dampak terjadi
      vi. Saran perbaikan dan rencana penanganan
   b. Lindungi karyawan yang melakukan eskalasi. Miliki kebijakan yang melindungi para anggota tim dari tindakan pembalasan jika mereka melakukan eskalasi di sekitar pengambil keputusan atau pemangku kepentingan yang tidak responsif. Terapkan mekanisme untuk mengidentifikasi apakah hal ini terjadi dan beri respons yang tepat.
5. Tumbuhkan budaya loop umpan balik perbaikan berkelanjutan dalam segala hal yang dihasilkan oleh organisasi. Loop umpan balik bertindak sebagai eskalasi kecil kepada individu yang bertanggung jawab, dan mereka mengidentifikasi peluang-peluang perbaikan, bahkan ketika eskalasi tidak diperlukan. Budaya perbaikan berkelanjutan mendorong setiap orang untuk menjadi lebih proaktif.
6. Pimpinan harus menekankan ulang secara berkala kebijakan, standar, mekanisme, dan keinginan untuk mewujudkan eskalasi yang terbuka dan loop umpan balik berkelanjutan tanpa tindakan pembalasan.

Tingkat upaya untuk Rencana Implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP05 Mekanisme tersedia untuk meminta penambahan, perubahan, dan pengecualian

Dokumen terkait:

- Bagaimana Anda menumbuhkan budaya perbaikan berkelanjutan dan belajar dari Andon dan sistem eskalasi?
- Kabel Andon (Revolusi TI)
- AWS DevOps Panduan | Tetapkan jalur eskalasi yang jelas dan dorong ketidaksepakatan yang konstruktif

Video terkait:

- Jeff Bezos tentang cara membuat keputusan (& meningkatkan kecepatan)
- Sistem Produk Toyota: Menghentikan Produksi, sebuah Tombol, dan Papan Listrik Andon
- Kabel Andon di Manufaktur LEAN

Contoh terkait:

- Bekerja dengan rencana eskalasi di Incident Manager

OPS03-BP04 Komunikasi yang tepat waktu, jelas, dan dapat ditindaklanjuti

Pimpinan bertanggung jawab untuk menciptakan komunikasi yang kuat dan efektif, terutama ketika organisasi mengadopsi strategi, teknologi, atau cara kerja baru. Pemimpin harus menetapkan ekspektasi bagi semua staf untuk bekerja untuk mencapai tujuan-tujuan perusahaan. Rancang mekanisme komunikasi yang menciptakan dan memelihara kesadaran di antara tim yang bertanggung jawab untuk menjalankan rencana yang didanai dan disponsori oleh pimpinan. Manfaatkan keragaman lintas organisasi, dan dengarkan dengan penuh perhatian berbagai perspektif yang unik. Gunakan perspektif ini untuk meningkatkan inovasi, menantang asumsi Anda, dan mengurangi risiko bias konfirmasi. Tumbuhkan inklusi, keragaman, dan kemudahan akses dalam tim Anda untuk mendapatkan perspektif yang bermanfaat.

Hasil yang diinginkan: Organisasi Anda merancang strategi-strategi komunikasi untuk mengatasi dampak yang ditimbulkan oleh perubahan-perubahan terhadap organisasi. Tim tetap mendapatkan informasi terbaru dan termotivasi untuk terus bekerja sama satu sama lain, bukan saling melawan. Individu perseorangan memahami betapa pentingnya peran mereka untuk mencapai tujuan-

tujuan yang telah ditetapkan. Email hanyalah mekanisme pasif untuk komunikasi dan digunakan sebagaimana mestinya. Manajemen meluangkan waktu dengan kontributor individual perorangan mereka untuk membantu mereka memahami tanggung jawab, tugas yang harus diselesaikan, dan bagaimana pekerjaan mereka berkontribusi pada keseluruhan misi. Jika perlu, para pemimpin melibatkan personel secara langsung di lokasi-lokasi yang lebih kecil untuk menyampaikan pesan dan memastikan bahwa pesan-pesan ini tersampaikan secara efektif. Sebagai hasil dari strategi komunikasi yang baik, organisasi menunjukkan kinerja sesuai atau melampaui harapan pimpinan. Pimpinan mendorong dan meminta pendapat yang beragam di dalam dan di seluruh tim.

Anti-pola umum:

- Organisasi Anda memiliki rencana lima tahun untuk memigrasi semua beban kerja ke AWS. Kasus bisnis untuk cloud mencakup modernisasi 25% dari semua beban kerja untuk memanfaatkan teknologi nirserver. CIO mengomunikasikan strategi ini untuk mengarahkan laporan dan mengharapkan setiap pemimpin untuk menyampaikan presentasi ini kepada manajer, direktur, dan kontributor perorangan tanpa melakukan komunikasi tatap muka. CIO memantau dari belakang dan mengharapkan organisasinya menjalankan strategi baru tersebut.
- Pimpinan tidak menyediakan atau menggunakan mekanisme untuk umpan balik, dan kesenjangan ekspektasi pun tumbuh, yang menyebabkan proyek terhenti.
- Anda diminta untuk membuat perubahan pada grup keamanan Anda, tetapi Anda tidak diberikan detail apa pun tentang perubahan yang perlu dilakukan, dampak perubahan yang dapat terjadi pada semua beban kerja, dan kapan hal tersebut seharusnya terjadi. Manajer meneruskan sebuah email dari VP InfoSec dan menambahkan pesan “Mari Wujudkan.”
- Perubahan dilakukan pada strategi migrasi Anda yang mengurangi jumlah modernisasi yang direncanakan dari 25% menjadi 10%. Perubahan ini memiliki efek hilir pada organisasi operasi. Mereka tidak diberi tahu tentang perubahan strategis ini sehingga mereka tidak siap dengan kapasitas personel terampil yang memadai untuk mendukung beban kerja yang diangkat dan digeser dalam jumlah yang lebih besar ke AWS.

Manfaat menjalankan praktik terbaik ini:

- Organisasi Anda selalu menerima informasi tentang strategi baru atau perubahan strategi, dan mereka bertindak sebagaimana mestinya dengan motivasi yang kuat untuk saling membantu guna mencapai keseluruhan tujuan dan metrik yang telah ditetapkan oleh pimpinan.
- Mekanisme tersedia dan digunakan untuk memberikan pengingat secara tepat waktu kepada anggota tim tentang risiko-risiko yang diketahui dan peristiwa-peristiwa yang direncanakan.

- Cara kerja baru (termasuk perubahan pada personel atau organisasi, proses, atau teknologi), beserta keterampilan yang dibutuhkan, diadopsi dengan lebih efektif oleh organisasi, dan organisasi Anda menyadari manfaat-manfaat bisnis dengan lebih cepat.
- Anggota tim memiliki konteks yang diperlukan tentang komunikasi-komunikasi yang diterima, dan mereka dapat bekerja dengan lebih efektif.

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

Panduan implementasi

Untuk mengimplementasikan praktik terbaik ini, Anda harus bekerja sama dengan para pemangku kepentingan di seluruh organisasi untuk menyepakati standar komunikasi. Publikasikan standar tersebut ke organisasi Anda. Untuk transisi IT yang signifikan, tim perencanaan yang telah sudah dibentuk dan mapan mempunyai kecenderungan lebih untuk berhasil mengelola dampak perubahan terhadap personelnya daripada organisasi yang mengabaikan praktik ini. Manajemen perubahan bisa lebih menantang untuk organisasi yang lebih besar, karena dukungan yang kuat untuk strategi baru sangat diperlukan dari semua kontributor individual perorangan. Dengan tidak adanya tim perencanaan transisi seperti ini, pimpinan memegang 100% tanggung jawab untuk melakukan komunikasi yang efektif. Saat membentuk tim perencanaan transisi, tugaskan para anggota tim untuk bekerja dengan semua pimpinan organisasi untuk menentukan dan mengelola komunikasi yang efektif di setiap tingkat.

Contoh pelanggan

AnyCompany Retail mendaftar untuk Dukungan Perusahaan AWS dan mengandalkan penyedia layanan pihak ketiga lain untuk operasi cloud-nya. Perusahaan menggunakan obrolan dan chatops sebagai media komunikasi utama mereka untuk aktivitas-aktivitas operasional. Peringatan dan informasi lainnya memenuhi saluran tertentu. Ketika seseorang harus bertindak, mereka menyatakan hasil yang diinginkan dengan jelas, dan dalam banyak kasus, mereka menerima runbook atau playbook yang bisa digunakan. Mereka menjadwalkan perubahan besar pada sistem produksi yang memiliki sebuah kalender perubahan.

Langkah-langkah implementasi

1. Bangunlah sebuah tim inti di dalam organisasi yang memiliki pertanggungjawaban untuk membangun dan memulai rencana komunikasi untuk perubahan yang terjadi di berbagai tingkatan di dalam organisasi.

2. Tetapkan kepemilikan utas tunggal untuk mencapai pengawasan. Bekali masing-masing tim dengan kemampuan untuk melakukan inovasi secara mandiri, dan seimbangkan penggunaan mekanisme yang konsisten, yang memungkinkan inspeksi dan visi direksional pada level yang tepat.
3. Bekerjalah dengan para pemangku kepentingan di seluruh organisasi Anda untuk menyepakati standar komunikasi, praktik, dan rencana.
4. Pastikan bahwa tim komunikasi inti melakukan kolaborasi dengan para pimpinan organisasi dan program untuk menyusun pesan kepada staf yang sesuai atas nama para pemimpin.
5. Bangunlah mekanisme komunikasi strategis untuk mengelola perubahan melalui pengumuman, kalender bersama, rapat umum, dan metode tatap muka atau empat mata sehingga anggota tim memiliki ekspektasi yang sesuai tentang tindakan-tindakan yang harus mereka lakukan.
6. Sediakan konteks, detail, dan waktu yang diperlukan (apabila memungkinkan) untuk menentukan apakah tindakan perlu dilakukan. Ketika tindakan diperlukan, beritahukan tindakan apa yang diperlukan beserta dampaknya.
7. Implementasikan alat-alat yang memudahkan komunikasi taktis, seperti obrolan internal, email, dan manajemen pengetahuan.
8. Implementasikan mekanisme untuk mengukur dan memastikan bahwa semua komunikasi memberikan hasil yang diinginkan.
9. Buatlah sebuah loop umpan balik yang mengukur efektivitas semua komunikasi, terutama ketika komunikasi berkaitan dengan resistensi terhadap perubahan di seluruh organisasi.
10. Untuk semua Akun AWS, tetapkan kontak alternatif untuk penagihan, keamanan, dan operasi. Idealnya, setiap kontak harus berupa distribusi email, bukan kontak individu perseorangan tertentu.
11. Buatlah sebuah rencana eskalasi dan komunikasi eskalasi balik untuk berinteraksi dengan tim internal dan eksternal Anda, termasuk dukungan AWS dan penyedia layanan pihak ketiga lainnya.
12. Mulai dan jalankan strategi komunikasi secara konsisten selama berlangsungnya setiap program transformasi.
13. Prioritaskan tindakan yang dapat diulang, jika memungkinkan, untuk mengaktifkan otomatisasi yang aman dalam skala besar.
14. Ketika komunikasi perlu dilakukan dengan melibatkan tindakan-tindakan otomatis, tujuan komunikasi seharusnya adalah untuk memberikan informasi kepada tim, untuk melakukan audit, atau sebagai bagian dari proses manajemen perubahan.
15. Analisis komunikasi dari sistem peringatan Anda untuk mendeteksi hasil positif palsu atau peringatan yang terus-menerus dibuat. Hapus atau ubah peringatan-peringatan tersebut sehingga

peringatan tersebut dimulai ketika diperlukan intervensi manusia. Jika ada sebuah peringatan muncul, berikan runbook atau playbook.

a. Anda dapat menggunakan AWS Systems Manager Documents untuk membuat playbook dan runbook untuk peringatan.

16. Mekanisme diterapkan untuk memberikan pemberitahuan risiko atau acara-acara yang direncanakan dengan cara yang jelas dan dapat ditindaklanjuti, melalui peringatan yang memadai untuk memberi respons yang sesuai. Gunakan daftar email atau saluran obrolan untuk mengirimkan pemberitahuan sebelum acara yang sudah direncanakan.

a. AWS Chatbot dapat Anda gunakan untuk mengirim peringatan dan menanggapi peristiwa dalam platform pesan organisasi Anda.

17. Berikan sumber informasi yang dapat diakses di mana acara yang sudah direncanakan dapat ditemukan. Beri pemberitahuan tentang acara-acara yang direncanakan dari sistem yang sama.

a. AWS Systems Manager Change Calendar dapat Anda gunakan untuk membuat jendela perubahan ketika perubahan dapat terjadi. Hal ini memberikan para anggota tim pemberitahuan mengenai kapan mereka dapat membuat perubahan dengan aman.

18. Pantau pemberitahuan kerentanan dan informasi patch untuk memahami kerentanan yang memiliki risiko tinggi dan risiko potensial yang berkaitan dengan komponen beban kerja Anda. Berikan pemberitahuan kepada para anggota tim agar mereka dapat bertindak.

a. Anda dapat berlangganan Buletin Keamanan AWS untuk menerima pemberitahuan kerentanan pada AWS.

19. Cari pendapat dan perspektif yang beragam: Dorong kontribusi dari semua orang. Berikan kesempatan komunikasi kepada kelompok yang kurang terwakili. Lakukan rotasi peran dan tanggung jawab dalam rapat.

a. Perluas peran dan tanggung jawab: Sediakan kesempatan bagi anggota tim untuk mengambil peran yang mungkin jarang bisa mereka ambil. Mereka bisa mendapatkan pengalaman dan perspektif dari peran tersebut serta dari interaksi dengan para anggota tim baru yang mungkin tidak akan berinteraksi dengan mereka di luar peran tersebut. Mereka juga dapat membawa pengalaman dan perspektif mereka ke peran baru tersebut serta untuk para anggota tim yang berinteraksi dengan mereka. Begitu perspektif meningkat, lakukan identifikasi terhadap kesempatan bisnis yang muncul atau peluang perbaikan baru. Lakukan rotasi tugas-tugas umum di antara para anggota di dalam tim agar mereka dapat memahami tuntutan dan dampak yang ditimbulkan dari pelaksanaan tugas-tugas yang biasanya dijalankan oleh anggota yang lain.

b. Sediakan lingkungan yang aman dan ramah: Miliki kebijakan dan kontrol yang melindungi mental dan keselamatan fisik anggota tim dalam organisasi Anda. Para anggota tim harus bisa berinteraksi tanpa rasa takut akan pembalasan. Ketika para anggota tim merasa aman dan diterima, mereka mungkin menjadi lebih terlibat dan produktif. Makin beragam organisasi Anda, maka akan makin baik pemahaman Anda tentang orang-orang yang Anda dukung termasuk para pelanggan Anda. Ketika anggota tim Anda merasa nyaman, merasa bebas untuk berbicara, dan meyakini bahwa suara mereka akan didengar, mereka lebih berpeluang untuk membagikan wawasan berharga (misalnya, peluang pemasaran, kebutuhan aksesibilitas, segmen pasar yang belum terlayani, dan risiko-risiko yang tidak diketahui di lingkungan Anda).
c. Dukung anggota tim untuk berpartisipasi penuh: Sediakan sumber daya yang diperlukan bagi karyawan Anda untuk berpartisipasi penuh pada semua aktivitas yang berkaitan dengan pekerjaan. Para anggota tim yang sehari-hari berhadapan dengan tantangan akan mengembangkan keterampilan untuk pekerjaan-pekerjaan di sekitar mereka. Keterampilan yang dikembangkan secara khusus ini bisa memberi keuntungan yang signifikan bagi organisasi Anda. Dukung para anggota tim dengan akomodasi yang diperlukan untuk meningkatkan keuntungan yang bisa Anda terima dari kontribusi mereka.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS03-BP01 Memberikan sponsor eksekutif
- OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur
- OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah

Dokumen terkait:

- Posting blog AWS | Akuntabilitas dan pemberdayaan adalah kunci bagi organisasi tangkas yang berkinerja tinggi
- AWS Executive Insights | Belajar untuk meningkatkan (menskalakan) inovasi, bukan kompleksitas | Single-threaded Leaders
- Buletin Keamanan AWS
- Buka CVE
- Aplikasi Dukungan di Slack untuk Mengelola Kasus Dukungan

- Kelola sumber daya AWS di saluran Slack Anda dengan Amazon Q Developer dalam aplikasi obrolan

Layanan terkait:

- Amazon Q Developer dalam aplikasi obrolan
- AWS Kalender Perubahan Systems Manager
- AWS Dokumen Systems Manager

OPS03-BP05 Mendorong eksperimen

Eksperimen adalah katalis untuk mengubah ide baru menjadi produk dan fitur. Eksperimen mempercepat proses pembelajaran dan membuat anggota tim terus tertarik dan terlibat. Anggota tim didorong untuk sering bereksperimen guna mendorong inovasi. Meskipun hasil yang tidak diinginkan terjadi, ada nilai dalam memiliki pengetahuan tentang apa yang sebaiknya tidak dilakukan. Anggota tim tidak dihukum untuk eksperimen yang berhasil dengan hasil yang tidak diinginkan.

Hasil yang diinginkan:

- Organisasi Anda mendorong eksperimen untuk mendukung inovasi.
- Eksperimen digunakan sebagai peluang untuk belajar.

Anti-pola umum:

- Anda ingin menjalankan pengujian A/B tetapi tidak ada mekanisme untuk menjalankan eksperimen tersebut. Anda melakukan deployment perubahan UI tanpa memiliki kemampuan untuk mengujinya. Tindakan tersebut mengakibatkan pengalaman pelanggan yang negatif.
- Perusahaan Anda hanya memiliki lingkungan produksi dan lingkungan pentahapan. Tidak ada lingkungan sandbox untuk melakukan eksperimen dengan fitur atau produk baru sehingga Anda harus melakukan eksperimen di dalam lingkungan produksi.

Manfaat menjalankan praktik terbaik ini:

- Eksperimen mendorong inovasi.
- Anda dapat bereaksi lebih cepat terhadap umpan balik yang diberikan oleh pengguna melalui eksperimen.

- Organisasi Anda mengembangkan budaya belajar.

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

Panduan implementasi

Eksperimen harus dijalankan dengan cara yang aman. Manfaatkan beberapa lingkungan untuk melakukan eksperimen tanpa membahayakan sumber daya produksi. Gunakan pengujian A/B dan bendera fitur untuk menguji eksperimen. Berikan kepada para anggota tim kemampuan untuk melakukan eksperimen di dalam sebuah lingkungan sandbox.

Contoh pelanggan

AnyCompany Retail mendorong eksperimen. Anggota tim dapat menggunakan 20% dari hari-hari kerja mereka untuk melakukan eksperimen atau mempelajari teknologi baru. Mereka memiliki sebuah lingkungan sandbox di mana mereka dapat berinovasi. Pengujian A/B digunakan untuk fitur-fitur baru guna memvalidasinya dengan umpan balik nyata dari pengguna.

Langkah-langkah implementasi

1. Bekerjasamalah dengan para pimpinan di seluruh organisasi Anda untuk mendukung eksperimen. Para anggota tim harus didorong untuk melakukan eksperimen dengan cara yang aman.
2. Berikan kepada para anggota tim Anda sebuah lingkungan di mana mereka dapat melakukan eksperimen dengan aman. Mereka harus memiliki akses ke sebuah lingkungan yang mirip dengan lingkungan produksi.
   a. Anda dapat menggunakan Akun AWS terpisah untuk membuat lingkungan sebuah sandbox untuk eksperimen. AWS Control Tower dapat digunakan untuk menyediakan akun ini.
3. Gunakan bendera fitur dan pengujian A/B untuk melakukan eksperimen dengan aman dan mengumpulkan umpan balik pengguna.
   a. AWS AppConfig Feature Flags menyediakan kemampuan untuk membuat fitur bendera.
   b. Anda dapat menggunakan versi AWS Lambda untuk menerapkan versi baru dari fungsi untuk uji beta.

Tingkat upaya untuk rencana implementasi: Tinggi. Memberikan kepada para anggota tim sebuah lingkungan untuk melakukan eksperimen dan cara yang aman untuk melakukan eksperimen dapat mengharuskan Anda menanamkan investasi besar. Anda juga mungkin harus melakukan modifikasi terhadap kode aplikasi untuk menggunakan bendera fitur atau mendukung pengujian A/B.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP02 Lakukan analisis pasca-insiden - Belajar dari insiden adalah pendorong penting untuk inovasi bersama dengan eksperimen.
- OPS11-BP03 Menerapkan loop umpan balik - Loop umpan balik adalah bagian penting dari eksperimen.

Dokumen terkait:

- Pandangan Bagian Dalam Budaya Amazon: Eksperimen, Kegagalan, dan Obsesi Pelanggan
- Praktik terbaik untuk membuat dan mengelola akun sandbox di AWS
- Buat Budaya Eksperimen yang Diaktifkan oleh Cloud
- Mengaktifkan eksperimen dan inovasi di cloud di Sulamérica Seguros
- Eksperimen Lebih Banyak, Lebih Sedikit Gagal
- Mengatur Lingkungan AWS Anda dengan Menggunakan Beberapa Akun - Sandbox OU
- Menggunakan AWS AppConfig Feature Flag

Video terkait:

- AWS on Air ft. Amazon CloudWatch Evidently | Peristiwa AWS
- AWS on Air San Fran Summit 2022 ft. Integrasi AWS AppConfig Feature Flags dengan Jira
- AWS re:Invent 2022 - Deployment bukan perilisan: Kontrol peluncuran Anda dengan bendera fitur (BOA305-R)
- Buat Akun AWS Secara Pemrograman dengan AWS Control Tower
- Menyiapkan Lingkungan Multi-Akun AWS yang Menggunakan Praktik Terbaik untuk AWS Organizations

Contoh terkait:

- Sandbox Inovasi AWS
- Personalisasi End-to-End 101 untuk E-Commerce

Layanan terkait:

- Amazon CloudWatch Evidently
- AWS AppConfig
- AWS Control Tower

OPS03-BP06 Mendorong dan mendukung para anggota tim untuk mempertahankan dan mengembangkan tingkat keterampilan mereka

Tim harus mengembangkan tingkat keterampilan mereka untuk mengadopsi perkembangan teknologi, serta untuk mengimbangi perubahan permintaan dan tanggung jawab dalam mendukung beban kerja Anda. Perkembangan keterampilan menggunakan teknologi dapat menjadi sumber kepuasan tim dan mendorong inovasi. Dukung anggota tim Anda untuk mendapatkan dan mempertahankan sertifikasi industri yang memvalidasi dan mengakui perkembangan keterampilan mereka. Terapkan pelatihan silang untuk mendorong transfer pengetahuan dan meminimalkan dampak signifikan yang terjadi karena kehilangan anggota tim berpengalaman yang memiliki keterampilan dan pengetahuan terkait lembaga. Berikan waktu khusus yang terstruktur untuk pembelajaran.

AWS menyediakan sumber daya, termasuk Memulai Pusat Sumber Daya AWS, Blog AWS, Bincang Teknologi Online AWS, Acara dan Webinar AWS, dan Lab AWS Well-Architected, yang memberikan panduan, contoh, dan panduan terperinci untuk mendidik tim Anda.

Sumber daya seperti Dukungan, (AWS re:Post, Pusat Dukungan), dan Dokumentasi AWS akan membantu menghilangkan hambatan teknis dan memperbaiki operasi. Hubungi Dukungan melalui Pusat Dukungan jika Anda memiliki pertanyaan.

AWS juga membagikan praktik dan pola-pola terbaik yang telah kami pelajari melalui pengoperasian AWS di Amazon Builders' Library dan berbagai macam materi pendidikan lainnya yang bermanfaat melalui Blog AWS dan Siniar Resmi AWS.

AWS Training dan Sertifikasi mencakup pelatihan gratis melalui kursus digital mandiri, bersama dengan rencana pembelajaran berdasarkan peran atau domain. Anda juga dapat mengikuti pelatihan yang dibimbing oleh instruktur untuk mendukung perkembangan keterampilan AWS tim Anda.

Hasil yang diinginkan: Organisasi Anda terus-menerus melakukan evaluasi terhadap kesenjangan keterampilan yang terjadi dan mengatasinya dengan membuat anggaran dan menanamkan investasi yang terstruktur. Tim mendorong dan mendukung para anggota mereka dengan aktivitas-aktivitas

peningkatan keterampilan, misalnya dengan memperoleh sertifikasi industri terkemuka. Tim memanfaatkan program berbagi pengetahuan khusus seperti sesi makan siang sambil belajar, hari imersi, hackathon, dan game day. Organisasi Anda menjaga sistem pengetahuannya agar tetap mutakhir dan relevan untuk melatih silang para anggota tim, termasuk pelatihan orientasi karyawan baru.

Anti-pola umum:

- Dengan tidak adanya program pelatihan dan anggaran yang terstruktur, tim akan mengalami ketidakpastian saat mereka mencoba mengimbangi perkembangan teknologi, dan hal ini akan mengakibatkan meningkatnya gesekan.
- Sebagai bagian dari migrasi ke AWS, organisasi Anda menunjukkan adanya kesenjangan keterampilan dan beragamnya tingkat penguasaan keterampilan di antara tim. Tanpa upaya untuk meningkatkan keterampilan, tim akan mendapati diri mereka terbebani dengan manajemen warisan dan tidak efisien dari lingkungan cloud yang akan menyebabkan meningkatnya kerja keras yang harus dilakukan operator. Kelelahan fisik dan mental ini akan meningkatkan tingkat ketidakpuasan karyawan.

Manfaat menerapkan praktik terbaik ini: Ketika organisasi Anda secara sadar berinvestasi dalam meningkatkan keterampilan timnya, hal itu juga akan membantu mempercepat dan meningkatkan skala adopsi dan pengoptimalan cloud. Program pembelajaran yang tertarget dapat mendorong inovasi dan membangun kemampuan operasional bagi tim agar mereka siap menangani peristiwa-peristiwa yang terjadi. Tim secara sadar berinvestasi untuk mengimplementasikan dan mengembangkan praktik terbaik. Tim memiliki semangat yang tinggi, dan anggota tim menghargai kontribusi mereka terhadap bisnis.

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

Panduan implementasi

Untuk mengadopsi teknologi baru, mendorong inovasi, dan mengimbangi laju perubahan-perubahan yang terjadi pada permintaan dan tanggung jawab untuk mendukung beban kerja Anda, teruslah berinvestasi dalam pertumbuhan profesional tim Anda.

Langkah-langkah implementasi

1. Gunakan program advokasi cloud terstruktur: AWS Skills Guild menyediakan pelatihan konsultatif untuk meningkatkan kepercayaan keterampilan cloud dan memicu budaya pembelajaran yang berkelanjutan.

2. Sediakan sumber daya untuk kepentingan edukasi: Sediakan waktu khusus yang terstruktur, akses ke materi pelatihan, sumber daya lab, dan dukung partisipasi untuk mengikuti konferensi dan akses ke organisasi profesional yang memberikan kesempatan untuk belajar dari pendidik dan rekan. Berikan akses kepada anggota tim junior Anda agar dia bisa belajar dari anggota tim senior, atau biarkan anggota tim junior mengamati pekerjaan anggota tim senior serta melihat metode dan keterampilan mereka. Dorong pembelajaran tentang konten yang tidak terkait langsung dengan pekerjaan agar tim junior tersebut memiliki pandangan yang lebih luas.
3. Dorong penggunaan sumber daya teknis ahli: Manfaatkan sumber daya seperti AWS Re:Post untuk mendapatkan akses ke pengetahuan-pengetahuan pilihan dan komunitas yang dinamis.
4. Membangun dan memelihara repositori pengetahuan terkini: Gunakan platform berbagi pengetahuan, misalnya wiki dan runbook. Buat sumber pengetahuan pakar Anda sendiri yang dapat digunakan kembali dengan AWS re:Post Private untuk merampingkan kolaborasi dan meningkatkan produktivitas serta mempercepat proses onboarding karyawan.
5. Edukasi tim dan interaksi antar-tim: Buat rencana untuk kebutuhan anggota tim terkait pembelajaran berkelanjutan. Berikan kesempatan kepada para anggota tim untuk bergabung dengan tim-tim yang lain (baik sementara atau pun seterusnya) agar mereka bisa berbagi keterampilan dan praktik terbaik yang bermanfaat bagi organisasi Anda.
6. Dukung untuk mendapatkan dan mempertahankan sertifikasi industri: Dukung para anggota tim Anda dalam melakukan proses akuisisi dan pemeliharaan atas sertifikasi industri yang memberikan validasi atas kemampuan yang telah mereka pelajari, serta berikan pengakuan atas pencapaian-pencapaian yang mereka raih.

Tingkat upaya untuk rencana implementasi: Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS03-BP01 Memberikan sponsor eksekutif
- OPS11-BP04 Menjalankan manajemen pengetahuan

Dokumen terkait:

- Laporan Resmi AWS | Kerangka Kerja Adopsi Cloud: Perspektif Orang-orang
- Berinvestasi dalam pembelajaran berkelanjutan untuk mengembangkan masa depan organisasi Anda

- AWS Skills Guild
- AWS Training dan Sertifikasi
- Dukungan
- AWS re:Post
- Pusat Sumber Daya untuk Memulai AWS
- Blog AWS
- Kepatuhan AWS Cloud
- AWS Dokumentasi
- Siniar Resmi AWS.
- Bincang Teknologi Online AWS
- Acara dan Webinar AWS
- Lab AWS Well-Architected
- Amazon Builders' Library

Video terkait:

- AWS re:Invent 2023 | Menyempurnakan keterampilan dengan kecepatan cloud: Mengubah karyawan menjadi wirausahawan
- WS re:Invent 2023 | Membangun budaya keingintahuan melalui gamifikasi

OPS03-BP07 Tim sumber daya dengan tepat

Sediakan anggota tim mahir dalam jumlah yang tepat, serta sediakan alat dan sumber daya untuk mendukung kebutuhan beban kerja Anda. Anggota tim yang terlalu terbebani dapat meningkatkan risiko terjadinya kesalahan manusia. Berinvestasi dalam alat dan sumber daya, seperti otomatisasi, dapat meningkatkan (menskalakan) efektivitas tim Anda dan akan membantu mereka dalam mendukung lebih banyak beban kerja tanpa memerlukan kapasitas tambahan.

Hasil yang diinginkan:

- Anda telah mengatur tim Anda dengan tepat untuk mendapatkan keahlian yang dibutuhkan bagi mereka untuk mengoperasikan beban kerja sesuai dengan rencana migrasi Anda. AWS Karena tim Anda telah meningkatkan dirinya sendiri selama proyek migrasi Anda, mereka telah memperoleh

kemahiran dalam AWS teknologi inti yang direncanakan bisnis untuk digunakan saat memigrasi atau memodernisasi aplikasi mereka.

- Anda telah menyelaraskan rencana jumlah personel Anda dengan hati-hati untuk memanfaatkan sumber daya secara efisien dengan memanfaatkan otomatisasi dan alur kerja. Tim yang lebih kecil sekarang dapat mengelola lebih banyak infrastruktur atas nama tim pengembangan aplikasi.
- Dengan perubahan prioritas operasional, kendala-kendala yang dihadapi personel sumber daya diidentifikasi secara proaktif untuk melindungi keberhasilan inisiatif bisnis.
- Metrik-metrik operasional yang melaporkan kerja keras operasional (seperti kelelahan akibat kondisi siaga atau pemanggilan yang berlebihan) ditinjau untuk memastikan bahwa personel tidak kewalahan.

Anti-pola umum:

- Staf Anda belum meningkatkan AWS keterampilan saat Anda mendekati rencana migrasi cloud multi-tahun Anda, yang berisiko mendukung beban kerja dan menurunkan moral karyawan.
- Seluruh organisasi IT Anda sedang beralih ke cara-cara kerja tangkas. Bisnis memprioritaskan portofolio produk dan menetapkan metrik-metrik untuk fitur apa saja yang perlu dikembangkan terlebih dahulu. Proses tangkas Anda tidak mengharuskan tim untuk menetapkan story point ke rencana kerja mereka. Akibatnya, mustahil bagi Anda untuk mengetahui tingkat kapasitas yang dibutuhkan untuk jumlah pekerjaan berikutnya, atau apakah Anda memiliki keterampilan yang tepat yang ditugaskan untuk pekerjaan tersebut.
- Anda meminta AWS mitra memigrasi beban kerja Anda, dan Anda tidak memiliki rencana transisi dukungan untuk tim Anda setelah mitra menyelesaikan proyek migrasi. Tim Anda mengalami kesulitan untuk mendukung beban kerja secara efisien dan efektif.

Manfaat menerapkan praktik terbaik ini: Anda memiliki anggota tim yang terampil yang tersedia di dalam organisasi Anda untuk mendukung beban kerja. Alokasi sumber daya dapat beradaptasi dengan perubahan prioritas tanpa memengaruhi kinerja. Hasilnya adalah tim yang mahir dalam mendukung beban kerja sambil memaksimalkan waktu untuk berkonsentrasi pada inovasi bagi pelanggan, yang pada gilirannya meningkatkan kepuasan karyawan.

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

Panduan implementasi

Perencanaan sumber daya untuk migrasi cloud Anda harus dilakukan pada level organisasi yang selaras dengan rencana migrasi Anda, dan model operasi yang diinginkan yang sedang diimplementasikan untuk mendukung lingkungan cloud baru Anda. Ini harus mencakup pemahaman tentang teknologi-teknologi cloud mana yang di-deploy untuk tim pengembangan aplikasi dan bisnis. Pimpinan infrastruktur dan operasi harus merencanakan analisis kesenjangan keterampilan, pelatihan, dan penetapan peran untuk para rekayasawan yang memimpin adopsi cloud.

Langkah-langkah implementasi

1. Tentukan kriteria keberhasilan untuk keberhasilan tim dengan metrik-metrik operasional yang relevan seperti produktivitas personel (misalnya, biaya untuk mendukung beban kerja tertentu atau jam operator yang dihabiskan selama insiden).
2. Tetapkan perencanaan kapasitas sumber daya dan mekanisme inspeksi untuk memastikan bahwa keseimbangan yang tepat dari kapasitas yang memenuhi syarat benar-benar tersedia saat diperlukan dan dapat disesuaikan dari waktu ke waktu.
3. Ciptakan mekanisme (misalnya, mengirimkan survei bulanan kepada tim) untuk memahami tantangan-tantangan terkait pekerjaan yang memengaruhi tim (seperti meningkatnya tanggung jawab, perubahan teknologi, kehilangan personel, atau peningkatan pelanggan yang didukung).
4. Gunakan mekanisme-mekanisme tersebut untuk berinteraksi dengan tim dan menemukan tren yang mungkin menjadi faktor tantangan produktivitas karyawan. Ketika ada faktor eksternal yang memengaruhi kinerja tim Anda, lakukan evaluasi kembali tujuan dan sesuaikan target sebagaimana mestinya. Identifikasi rintangan yang menghambat kemajuan tim Anda.
5. Lakukan peninjauan secara rutin mengenai apakah sumber daya yang saat ini disediakan masih memadai, atau apakah diperlukan sumber daya tambahan, dan lakukan penyesuaian yang tepat untuk mendukung tim.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS03-BP06 Anggota tim didorong untuk mempertahankan dan mengembangkan keahlian mereka
- OPS09-BP03 Meninjau metrik operasi dan memprioritaskan peningkatan

- OPS10-BP01 Gunakan proses untuk manajemen peristiwa, insiden, dan masalah
- OPS10-BP07 Otomatiskan tanggapan terhadap acara

Dokumen terkait:

- AWS Cloud Kerangka Adopsi: Perspektif Orang
- Menjadi Perusahaan yang Siap Menghadapi Masa Depan
- Prioritaskan Keterampilan Karyawan Anda untuk Mendorong Pertumbuhan Bisnis
- Organisasi berkinerja tinggi - Tim Dua-Pizza Amazon
- Bagaimana Perusahaan dengan Pemanfaatan Cloud yang Matang Meraih Kesuksesannya

Persiapkan

Untuk menyiapkan keunggulan operasional, Anda harus memahami beban kerja Anda serta perkiraan perilakunya. Dengan begitu Anda akan mampu merancangnya agar dapat menyediakan wawasan tentang statusnya dan membangun prosedur untuk mendukungnya.

Desain beban kerja Anda sedemikian rupa sehingga memberikan informasi yang Anda perlukan untuk memahami status internalnya (seperti metrik, log, dan jejak) di semua komponen untuk mendukung observabilitas dan investigasi masalah. Observabilitas lebih dari sekadar pemantauan sederhana, sehingga memberikan pemahaman yang komprehensif tentang cara kerja internal sistem berdasarkan output eksternalnya. Berakar pada metrik, log, dan jejak, observabilitas menawarkan wawasan mendalam tentang perilaku dan dinamika sistem. Dengan observabilitas yang efektif, tim dapat membedakan pola, anomali, dan tren, memungkinkan mereka untuk secara proaktif mengatasi masalah potensial dan menjaga kesehatan sistem yang optimal. Mengidentifikasi indikator kinerja utama (KPI) sangat penting untuk memastikan keselarasan antara kegiatan pemantauan dan tujuan bisnis. Penyelarasan ini memastikan bahwa tim membuat keputusan berbasis data menggunakan metrik yang benar-benar penting, mengoptimalkan kinerja sistem dan hasil bisnis. Selain itu, observabilitas memberdayakan bisnis untuk menjadi proaktif, bukan reaktif. Tim dapat memahami hubungan sebab-akibat dalam sistem mereka, memprediksi dan mencegah masalah, bukan hanya bereaksi terhadapnya. Seiring berkembangnya beban kerja, penting untuk meninjau kembali dan menyempurnakan strategi observabilitas, guna memastikannya tetap relevan dan efektif.

Adopsi pendekatan yang meningkatkan aliran perubahan ke dalam produksi dan yang mencapai pemfaktoran ulang, umpan balik cepat atas kualitas, dan perbaikan bug. Hal-hal ini mempercepat perubahan positif yang memasuki tahap produksi, membatasi masalah yang diterapkan, dan memungkinkan identifikasi serta perbaikan yang cepat terhadap masalah yang muncul dari aktivitas deployment atau yang ditemukan di lingkungan Anda.

Adopsi pendekatan yang memberikan umpan balik cepat atas kualitas dan mencapai pemulihan cepat dari perubahan yang tidak memiliki hasil yang tidak diinginkan. Menggunakan praktik tersebut akan memitigasi dampak masalah akibat deployment perubahan. Antisipasikan perubahan yang tidak berhasil sehingga Anda mampu merespons lebih cepat jika dibutuhkan serta menguji dan memvalidasi perubahan yang Anda buat. Perhatikan aktivitas terencana di lingkungan Anda sehingga Anda dapat mengelola risiko perubahan yang mempengaruhi aktivitas terencana. Prioritaskan perubahan yang sering, kecil, dan dapat dikembalikan untuk membatasi cakupan perubahan. Hal ini menghasilkan pemecahan masalah dan perbaikan yang lebih cepat dengan opsi membatalkan

perubahan. Dengan begitu Anda juga dapat memperoleh manfaat dari perubahan yang berharga secara lebih sering.

Evaluasi kesiapan operasional beban kerja, proses, prosedur, dan personel Anda untuk memahami risiko operasional terkait beban kerja Anda. Gunakan proses yang konsisten (termasuk daftar periksa manual dan otomatis) untuk mengetahui saat Anda siap untuk mengoperasikan beban kerja Anda atau untuk melakukan perubahan. Hal ini juga akan membantu Anda menemukan area mana pun yang harus Anda buatkan rencana untuk ditangani. Miliki runbook yang mendokumentasikan aktivitas rutin serta buku pedoman yang memandu proses penyelesaian masalah Anda. Pahami manfaat dan risiko untuk membuat keputusan yang tepat agar perubahan dapat diterapkan dalam produksi.

AWS memungkinkan Anda menampilkan keseluruhan beban kerja (aplikasi, infrastruktur, kebijakan, tata kelola, dan operasi) sebagai kode. Hal ini berarti Anda dapat menerapkan disiplin rekayasa yang sama yang Anda gunakan untuk kode aplikasi ke setiap elemen tumpukan Anda dan membagikan semuanya ke seluruh tim atau organisasi untuk memperbesar manfaat upaya pengembangan. Gunakan operasi sebagai kode di cloud dan kemampuan untuk bereksperimen dengan aman guna mengembangkan beban kerja Anda, prosedur operasi Anda, serta antisipasi kegagalan. Menggunakan CloudFormation akan memungkinkan Anda untuk memiliki lingkungan pengembangan, pengujian, dan produksi sandbox yang konsisten, bertemplat dengan tingkat kontrol operasi yang makin meningkat.

Berinvestasi dalam implementasi aktivitas operasi sebagai kode untuk memaksimalkan produktivitas personel operasi, meminimalkan tingkat kesalahan, dan mencapai respons otomatis. Gunakan “pre-mortem” untuk mengantisipasi kegagalan dan membuat prosedur ketika diperlukan. Terapkan metadata dengan menggunakan Tag Sumber Daya dan AWS Resource Groups sesuai strategi penandaan yang konsisten untuk mencapai identifikasi sumber daya Anda. Tandai sumber daya Anda untuk pengaturan, akuntansi biaya, kontrol akses, dan penargetan pelaksanaan aktivitas operasi otomatis. Adopsi praktik deployment yang memanfaatkan elastisitas cloud untuk memfasilitasi aktivitas pengembangan, dan pra-deployment sistem untuk implementasi yang lebih cepat. Ketika Anda membuat perubahan pada daftar periksa yang Anda gunakan untuk mengevaluasi beban kerja Anda, rencanakan apa yang akan Anda lakukan dengan sistem langsung yang tidak lagi patuh.

Topik

- Implementasikan observabilitas
- Desain operasi
- Memitigasi risiko deployment

- Kesiapan operasional dan manajemen perubahan

Implementasikan observabilitas

Terapkan observabilitas dalam beban kerja Anda sehingga Anda dapat memahami statusnya dan membuat keputusan berbasis data berdasarkan persyaratan bisnis.

Observabilitas lebih dari sekadar pemantauan sederhana, yang memberikan pemahaman yang komprehensif tentang cara kerja internal sistem berdasarkan output eksternalnya. Berakar pada metrik, log, dan jejak, observabilitas menawarkan wawasan mendalam tentang perilaku dan dinamika sistem. Dengan observabilitas yang efektif, tim dapat membedakan pola, anomali, dan tren, memungkinkan mereka untuk secara proaktif mengatasi masalah potensial dan menjaga kesehatan sistem yang optimal.

Mengidentifikasi indikator kinerja utama (KPI) sangat penting untuk memastikan keselarasan antara kegiatan pemantauan dan tujuan bisnis. Penyelarasan ini memastikan bahwa tim membuat keputusan berbasis data menggunakan metrik yang benar-benar penting, mengoptimalkan kinerja sistem dan hasil bisnis.

Selain itu, observabilitas memberdayakan bisnis untuk menjadi proaktif, bukan reaktif. Tim dapat memahami hubungan sebab-akibat dalam sistem mereka, memprediksi dan mencegah masalah, bukan hanya bereaksi terhadapnya. Seiring berkembangnya beban kerja, penting untuk meninjau kembali dan menyempurnakan strategi observabilitas, guna memastikannya tetap relevan dan efektif.

Praktik terbaik

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS04-BP03 Menerapkan telemetri pengalaman pengguna
- OPS04-BP04 Mengimplementasikan telemetri dependensi
- OPS04-BP05 Melaksanakan penelusuran terdistribusi

OPS04-BP01 Identifikasi indikator kinerja utama

Untuk mengimplementasikan observabilitas dalam beban kerja, Anda memulainya dengan memahami statusnya dan mengambil keputusan berbasis data berdasarkan persyaratan bisnis. Salah satu cara paling efektif untuk memastikan keselarasan antara kegiatan pemantauan dan tujuan bisnis adalah dengan mendefinisikan dan memantau indikator kinerja utama ()KPIs.

Hasil yang diinginkan: Praktik-praktik observabilitas yang efisien yang sangat selaras dengan tujuan bisnis, sehingga memastikan upaya pemantauan selalu memenuhi hasil bisnis yang nyata.

Anti-pola umum:

- Tidak terdefinisiKPIs: Bekerja tanpa jelas KPIs dapat menyebabkan pemantauan terlalu banyak atau terlalu sedikit, kehilangan sinyal vital.
- StatisKPIs: Tidak meninjau kembali atau menyempurnakan KPIs karena beban kerja atau tujuan bisnis berkembang.
- Ketidakselarasan: Berfokus pada metrik-metrik teknis yang tidak berkorelasi langsung dengan hasil bisnis atau yang lebih sulit untuk berkorelasi dengan masalah-masalah dunia nyata.

Manfaat menjalankan praktik terbaik ini:

- Kemudahan identifikasi masalah: Bisnis KPIs sering memunculkan masalah lebih jelas daripada metrik teknis. Penurunan dalam bisnis KPI dapat menunjukkan masalah dengan lebih efektif daripada memilah-milah berbagai metrik teknis.
- Keselarasan bisnis: Memastikan bahwa kegiatan pemantauan secara langsung adalah aktivitas yang mendukung tujuan bisnis.
- Efisiensi: Prioritaskan untuk melakukan pemantauan sumber daya dan memberikan perhatian pada metrik-metrik yang penting.
- Proaktif: Kenali dan atasi masalah sebelum masalah itu memunculkan dampak bisnis yang lebih luas.

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

Panduan implementasi

Untuk secara efektif mendefinisikan beban kerjaKPIs:

1. Mulai dengan hasil bisnis: Sebelum menyelami metrik, pahami dahulu hasil bisnis yang diinginkan. Apakah peningkatan penjualan, keterlibatan pengguna yang lebih tinggi, atau waktu respons yang lebih cepat?
2. Korelasikan metrik teknis dengan tujuan bisnis: Tidak semua metrik teknis memiliki dampak langsung pada hasil bisnis Anda. Identifikasi mereka yang melakukannya, tetapi seringkali lebih mudah untuk mengidentifikasi masalah menggunakan bisnis. KPI

3. Gunakan Amazon CloudWatch: Gunakan CloudWatch untuk menentukan dan memantau metrik yang mewakili Anda. KPIs
4. Tinjau dan perbarui secara teraturKPIs: Saat beban kerja dan bisnis Anda berkembang, jaga agar tetap relevan. KPIs
5. Libatkan pemangku kepentingan: Libatkan tim teknis dan bisnis dalam mendefinisikan dan meninjau. KPIs

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- the section called “OPS04-BP02 Melaksanakan telemetri aplikasi”
- the section called “OPS04-BP03 Menerapkan telemetri pengalaman pengguna”
- the section called “OPS04-BP04 Mengimplementasikan telemetri dependensi”
- the section called “OPS04-BP05 Melaksanakan penelusuran terdistribusi”

Dokumen terkait:

- AWS Praktik Terbaik Observabilitas
- CloudWatch Panduan Pengguna
- AWS Kursus Pembuat Keterampilan Observabilitas

Video terkait:

- Mengembangkan strategi observabilitas

Contoh terkait:

- Lokakarya Satu Observabilitas

OPS04-BP02 Melaksanakan telemetri aplikasi

Telemetri aplikasi berfungsi sebagai fondasi observabilitas beban kerja Anda. Sangat penting bagi Anda untuk menghadirkan telemetri yang menawarkan wawasan yang dapat ditindaklanjuti

tentang keadaan aplikasi Anda serta pencapaian hasil teknis dan bisnis. Dari pemecahan masalah hingga mengukur dampak fitur baru atau memastikan keselarasan dengan indikator kinerja kunci bisnis (KPIs), telemetri aplikasi menginformasikan cara Anda membangun, mengoperasikan, dan mengembangkan beban kerja Anda.

Metrik, log, dan jejak merupakan tiga pilar observabilitas utama. Ketiganya berfungsi sebagai alat diagnostik yang mampu menggambarkan keadaan aplikasi Anda. Seiring waktu, tiga hal ini akan membantu menciptakan garis acuan dan mengidentifikasi setiap anomali. Namun, untuk memastikan keselarasan antara kegiatan pemantauan dan tujuan bisnis, penting untuk menentukan dan memantau. KPIs Bisnis KPIs sering membuatnya lebih mudah untuk mengidentifikasi masalah dibandingkan dengan metrik teknis saja.

Jenis telemetri lainnya, seperti pemantauan pengguna nyata (RUM) dan transaksi sintetis, melengkapi sumber data primer ini. RUMmenawarkan wawasan tentang interaksi pengguna waktu nyata, sedangkan transaksi sintetis mensimulasikan perilaku pengguna potensial, membantu mendeteksi kemacetan sebelum pengguna nyata menghadapinya.

Hasil yang diinginkan: Dapatkan wawasan yang dapat ditindaklanjuti mengenai performa beban kerja Anda. Wawasan ini akan memungkinkan Anda untuk mengambil keputusan yang proaktif tentang optimalisasi performa, mencapai peningkatan stabilitas beban kerja, merampingkan proses CI/CD, dan memanfaatkan sumber daya secara efektif.

Anti-pola umum:

- Observabilitas yang tidak lengkap: Mengabaikan penggunaan observabilitas di setiap lapisan beban kerja, sehingga mengakibatkan titik buta yang dapat membuat performa sistem vital dan wawasan perilaku menjadi tidak jelas.
- Tampilan data terfragmentasi: Ketika data tersebar di beberapa alat dan sistem, mempertahankan pandangan yang menyeluruh tentang kondisi dan performa beban kerja Anda akan menjadi sesuatu yang sulit dilakukan.
- Masalah yang dilaporkan pengguna: Tanda bahwa deteksi masalah proaktif melalui telemetri dan pemantauan bisnis kurang. KPI

Manfaat menjalankan praktik terbaik ini:

- Pengambilan keputusan berdasarkan informasi: Dengan wawasan dari telemetri dan bisnisKPIs, Anda dapat membuat keputusan berbasis data.

- Peningkatan efisiensi operasional: Pemanfaatan sumber daya berbasis data akan menghasilkan efektivitas biaya.
- Penyempurnaan stabilitas beban kerja: Deteksi dan penyelesaian masalah yang lebih cepat akan menghasilkan peningkatan waktu aktif.
- Perampingan proses CI/CD: Wawasan dari data telemetri dapat memfasilitasi penyempurnaan proses dan pengiriman kode yang andal.

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

Panduan implementasi

Untuk menerapkan telemetri aplikasi untuk beban kerja Anda, gunakan layanan AWS seperti Amazon dan. CloudWatch AWS X-Ray Amazon CloudWatch menyediakan rangkaian alat pemantauan yang komprehensif, memungkinkan Anda mengamati sumber daya dan aplikasi di dalam AWS dan lingkungan lokal. Layanan ini mengumpulkan, melacak, dan menganalisis metrik, menggabungkan dan memantau data log, dan memberikan respons terhadap perubahan yang terjadi dalam sumber daya Anda, menyempurnakan pemahaman Anda tentang bagaimana beban kerja Anda beroperasi. Secara bersamaan, AWS X-Ray memungkinkan Anda melacak, menganalisis, dan men-debug aplikasi Anda, memberi Anda pemahaman mendalam tentang perilaku beban kerja Anda. Dengan fitur seperti peta layanan, distribusi latensi, dan jadwal penelusuran, AWS X-Ray memberikan wawasan tentang kinerja beban kerja Anda dan hambatan yang mempengaruhinya.

Langkah-langkah implementasi

1. Identifikasi data apa yang akan dikumpulkan: Pastikan metrik, log, dan jejak penting yang akan menawarkan wawasan substansial tentang kondisi, performa, dan perilaku beban kerja Anda.
2. Menyebarkan CloudWatchagen: Agen berperan penting dalam pengadaan metrik dan log sistem dan aplikasi dari beban kerja Anda dan infrastruktur dasarnya. CloudWatch CloudWatch Agen juga dapat digunakan untuk mengumpulkan OpenTelemetry atau jejak X-Ray dan mengirimkannya ke X-Ray.
3. Menerapkan deteksi anomali untuk log dan metrik: Gunakan deteksi anomali CloudWatch Log dan deteksi anomali CloudWatch Metrik untuk secara otomatis mengidentifikasi aktivitas yang tidak biasa dalam operasi aplikasi Anda. Alat-alat ini menggunakan algoritma machine learning untuk mendeteksi dan memberikan peringatan tentang anomali yang ditemukan, yang dapat meningkatkan kemampuan pemantauan Anda dan mempercepat waktu respons terhadap adanya potensi gangguan atau ancaman keamanan. Siapkan fitur-fitur ini untuk mengelola kesehatan dan keamanan aplikasi secara proaktif.

4. Mengamankan data log sensitif: Gunakan perlindungan data Amazon CloudWatch Logs untuk menutupi informasi sensitif di dalam log Anda. Fitur ini akan membantu Anda menjaga privasi dan kepatuhan dengan melakukan deteksi otomatis dan pengaburan data sensitif sebelum diakses. Menerapkan penyembunyian data untuk menangani dan melindungi detail sensitif dengan aman seperti informasi yang dapat diidentifikasi secara pribadi (). PII
5. Tentukan dan pantau bisnisKPIs: Tetapkan metrik khusus yang selaras dengan hasil bisnis Anda.
6. Instrumentasikan aplikasi Anda dengan AWS X-Ray: Selain menggunakan CloudWatch agen, penting untuk menginstruksikan aplikasi Anda untuk memancarkan data jejak. Proses ini dapat memberikan wawasan lebih lanjut tentang perilaku dan performa beban kerja Anda.
7. Standardisasi pengumpulan data di seluruh aplikasi Anda: Lakukan standardisasi terhadap praktik-praktik pengumpulan data di seluruh aplikasi Anda. Keseragaman bermanfaat dalam mengorelasikan dan menganalisis data, sehingga itu akan memberikan pandangan yang komprehensif tentang perilaku aplikasi Anda.
8. Menerapkan observabilitas lintas akun: Tingkatkan efisiensi pemantauan di beberapa akun dengan observabilitas CloudWatch lintas akun Akun AWS Amazon. Dengan fitur ini, Anda dapat menggabungkan metrik, log, dan alarm dari akun yang berbeda ke dalam satu tampilan, yang menyederhanakan manajemen dan meningkatkan waktu respons untuk masalah yang diidentifikasi di seluruh lingkungan organisasi Anda. AWS
9. Menganalisis dan bertindak berdasarkan data: Setelah pengumpulan dan normalisasi data dilakukan, gunakan Amazon CloudWatch untuk analisis metrik dan log, dan AWS X-Rayuntuk analisis jejak. Analisis tersebut dapat menghasilkan wawasan penting tentang kondisi, performa, dan perilaku beban kerja Anda, sehingga dapat memandu Anda dalam proses pengambilan keputusan.

Tingkat upaya untuk rencana implementasi: Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Tentukan beban kerja KPIs
- OPS04-BP03 Melaksanakan telemetri aktivitas pengguna
- OPS04-BP04 Menerapkan telemetri ketergantungan
- OPS04-BP05 Menerapkan ketertelusuran transaksi

Dokumen terkait:

- Praktik Terbaik Observabilitas AWS
- Panduan Pengguna CloudWatch
- AWS X-Ray Panduan Pengembang
- Menginstrumentasikan sistem terdistribusi untuk visibilitas operasional
- Kursus Skill Builder Observabilitas AWS
- Apa yang Baru dengan Amazon CloudWatch
- Apa yang baru dengan AWS X-Ray

Video terkait:

- AWS re:invent 2022 - Praktik terbaik observabilitas di Amazon
- AWS re:invent 2022 - Mengembangkan strategi observabilitas

Contoh terkait:

- Lokakarya Satu Observabilitas
- AWS Perpustakaan Solusi: Pemantauan Aplikasi dengan Amazon CloudWatch

OPS04-BP03 Menerapkan telemetri pengalaman pengguna

Memperoleh wawasan yang mendalam tentang pengalaman dan interaksi pelanggan dengan aplikasi Anda adalah hal yang krusial. Pemantauan pengguna nyata (RUM) dan transaksi sintetis berfungsi sebagai alat yang ampuh untuk tujuan ini. RUMmenyediakan data tentang interaksi pengguna nyata yang memberikan perspektif kepuasan pengguna tanpa filter, sementara transaksi sintetis mensimulasikan interaksi pengguna, membantu mendeteksi potensi masalah bahkan sebelum berdampak pada pengguna nyata.

Hasil yang diinginkan: Pandangan yang menyeluruh tentang pengalaman pelanggan, deteksi masalah yang proaktif, dan optimalisasi interaksi pengguna untuk memberikan pengalaman digital yang mulus (seamless).

Anti-pola umum:

- Aplikasi tanpa pemantauan pengguna nyata (RUM):

    - Deteksi masalah yang tertunda: TanpaRUM, Anda mungkin tidak menyadari kemacetan atau masalah kinerja hingga pengguna mengeluh. Pendekatan reaktif ini dapat menyebabkan pelanggan menjadi tidak puas.
    - Kurangnya wawasan pengalaman pengguna: Tidak menggunakan RUM berarti Anda kehilangan data penting yang menunjukkan bagaimana pengguna nyata berinteraksi dengan aplikasi Anda, sehingga membatasi kemampuan Anda untuk mengoptimalkan pengalaman pengguna.
- Aplikasi tanpa transaksi sintetis:
    - Kasus edge yang terlewatkan: Transaksi-transaksi sintetis akan membantu Anda untuk menguji jalur dan fungsi yang mungkin jarang digunakan oleh pengguna biasa, tetapi sangat penting untuk fungsi bisnis tertentu. Tanpanya, jalur-jalur tersebut bisa mengalami kesalahan fungsi dan luput dari perhatian.
    - Memeriksa masalah saat aplikasi tidak digunakan: Pengujian sintetis yang dilakukan secara rutin dapat memberikan simulasi saat-saat ketika pengguna nyata tidak berinteraksi secara aktif dengan aplikasi Anda, sehingga hal itu akan memastikan sistem selalu berfungsi dengan benar.

Manfaat menjalankan praktik terbaik ini:

- Deteksi masalah proaktif: Identifikasi dan atasi potensi masalah sebelum berdampak pada pengguna nyata.
- Pengalaman pengguna yang dioptimalkan: Umpan balik berkelanjutan dari RUM bantuan dalam menyempurnakan dan meningkatkan pengalaman pengguna secara keseluruhan.
- Wawasan tentang performa perangkat dan browser: Memahami performa aplikasi Anda di berbagai perangkat dan browser, sehingga memungkinkan optimalisasi lebih lanjut.
- Alur kerja bisnis yang divalidasi: Transaksi-transaksi sintetis rutin akan memastikan fungsionalitas inti dan jalur-jalur penting tetap berjalan dan efisien.
- Performa aplikasi yang ditingkatkan: Manfaatkan wawasan yang dikumpulkan dari data pengguna nyata untuk meningkatkan responsivitas dan keandalan aplikasi.

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

Panduan implementasi

Untuk memanfaatkan RUM dan transaksi sintetis untuk telemetri aktivitas pengguna, AWS menawarkan layanan seperti Amazon dan CloudWatch RUMAmazon CloudWatch Synthetics.

Metrik, log, dan jejak, ditambah dengan data aktivitas pengguna, memberikan sebuah pandangan yang komprehensif tentang status operasional aplikasi dan pengalaman pengguna.

Langkah-langkah implementasi

1. Menerapkan Amazon CloudWatch RUM: Integrasikan aplikasi Anda CloudWatch RUM untuk mengumpulkan, menganalisis, dan menyajikan data pengguna nyata.
   a. Gunakan CloudWatch RUM JavaScript perpustakaan untuk berintegrasi RUM dengan aplikasi Anda.
   b. Siapkan dasbor untuk memvisualisasikan dan memantau data pengguna nyata.
2. Configure CloudWatch Synthetics: Buat kenari, atau rutinitas skrip, yang mensimulasikan interaksi pengguna dengan aplikasi Anda.
   a. Tentukan alur kerja dan jalur aplikasi kritis.
   b. Desain kenari menggunakan skrip CloudWatch Synthetics untuk mensimulasikan interaksi pengguna untuk jalur ini.
   c. Jadwalkan dan pantau canary agar berjalan pada interval-interval tertentu, sehingga memastikan pemeriksaan performa yang konsisten.
3. Menganalisis dan bertindak berdasarkan data: Memanfaatkan data dari RUM dan transaksi sintetis untuk mendapatkan wawasan dan mengambil tindakan korektif ketika anomali terdeteksi. Gunakan CloudWatch dasbor dan alarm untuk tetap mendapat informasi.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS04-BP04 Mengimplementasikan telemetri dependensi
- OPS04-BP05 Melaksanakan penelusuran terdistribusi

Dokumen terkait:

- CloudWatch RUMPanduan Amazon
- Panduan Amazon CloudWatch Synthetics

Video terkait:

- Optimalkan aplikasi melalui wawasan pengguna akhir dengan Amazon CloudWatch RUM
- AWS di Air ft. Pemantauan Pengguna Nyata untuk Amazon CloudWatch

Contoh terkait:

- Lokakarya Satu Observabilitas
- Git Repository untuk Amazon CloudWatch RUM Web Client
- Menggunakan Amazon CloudWatch Synthetics untuk mengukur waktu buka halaman

OPS04-BP04 Mengimplementasikan telemetri dependensi

Telemetri dependensi sangat penting untuk memantau kondisi dan performa layanan dan komponen eksternal yang diandalkan oleh beban kerja Anda. Hal ini memberikan wawasan berharga tentang keterjangkauan, batas waktu, dan peristiwa penting lainnya yang terkait dengan dependensi seperti DNS, basis data, atau API pihak ketiga. Ketika Anda menginstrumentasi aplikasi Anda agar menghasilkan metrik, log, dan jejak tentang dependensi ini, Anda mendapatkan pemahaman yang lebih jelas tentang potensi kemacetan, masalah performa, atau kegagalan yang dapat memengaruhi beban kerja Anda.

Hasil yang diinginkan: Pastikan bahwa dependensi yang diandalkan beban kerja Anda menunjukkan performa yang sesuai harapan, sehingga Anda dapat secara proaktif mengatasi masalah-masalah dan memastikan performa beban kerja yang optimal.

Anti-pola umum:

- Mengabaikan dependensi eksternal: Hanya berfokus pada metrik aplikasi internal sambil mengabaikan metrik yang berkaitan dengan dependensi eksternal.
- Kurangnya pemantauan proaktif: Menunggu masalah muncul alih-alih terus memantau kondisi dan performa dependensi.
- Pemantauan model silo: Menggunakan beberapa alat pemantauan yang berbeda-beda sehingga wawasan tentang kondisi dependensi menjadi terfragmentasi dan tidak konsisten.

Manfaat menjalankan praktik terbaik ini:

- Peningkatan keandalan beban kerja: Dengan memastikan bahwa dependensi eksternal terus-menerus tersedia dan berkinerja optimal.
- Deteksi dan penyelesaian masalah yang lebih cepat: Secara proaktif mengidentifikasi dan menangani masalah pada dependensi sebelum berdampak pada beban kerja.
- Pandangan menyeluruh: Mendapatkan pandangan yang menyeluruh tentang komponen internal dan eksternal yang memengaruhi kondisi beban kerja.
- Peningkatan skalabilitas beban kerja: Dengan memahami batas skalabilitas dan karakteristik performa dependensi eksternal.

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

Panduan implementasi

Implementasikan telemetri dependensi dengan memulai melakukan identifikasi layanan, infrastruktur, dan proses yang digunakan oleh beban kerja Anda. Ukur seperti apa kondisi yang baik ketika dependensi berfungsi sesuai harapan, kemudian tentukan data apa yang akan diperlukan untuk mengukur kondisi-kondisi itu. Dengan informasi tersebut, Anda dapat membuat dasbor dan peringatan yang memberikan wawasan kepada tim operasi Anda tentang status dependensi tersebut. Gunakan alat-alat AWS untuk menemukan dan mengukur dampak yang ditimbulkan ketika dependensi tidak dapat menunjukkan hasil yang sesuai kebutuhan. Selalu tinjau ulang strategi Anda agar memperhitungkan perubahan-perubahan dalam prioritas, sasaran, dan wawasan yang diperoleh.

Langkah-langkah implementasi

Cara mengimplementasikan telemetri dependensi secara efektif:

1. Identifikasi dependensi eksternal: Lakukan kolaborasi dengan pemangku kepentingan untuk menentukan dependensi eksternal yang diandalkan oleh beban kerja Anda. Dependensi eksternal dapat mencakup layanan-layanan seperti basis data eksternal, API pihak ketiga, rute konektivitas jaringan ke lingkungan lain, dan layanan DNS. Langkah pertama menuju telemetri dependensi yang efektif adalah memiliki pemahaman yang menyeluruh tentang apa saja dependensi tersebut.
2. Kembangkan strategi pemantauan: Setelah Anda memiliki gambaran yang jelas tentang dependensi eksternal Anda, rancanglah strategi pemantauan yang disesuaikan dengan dependensi tersebut. Hal ini melibatkan pemahaman terhadap tingkat kekritisan setiap dependensi, perilaku yang diharapkan, dan perjanjian atau target tingkat layanan (SLA atau SLT)

terkait. Siapkan peringatan proaktif untuk memberi tahu Anda tentang perubahan status atau penyimpangan-penyimpangan performa.

3. Gunakan pemantauan jaringan: Gunakan Monitor Internet dan Monitor Jaringan, yang memberikan wawasan komprehensif mengenai kondisi internet dan jaringan global. Alat-alat ini akan membantu Anda untuk memahami dan merespons pemadaman, gangguan, atau penurunan kinerja yang memengaruhi dependensi eksternal Anda.
4. Terus dapatkan informasi dengan AWS Health. AWS Health adalah sumber informasi otoritatif tentang kondisi sumber daya AWS Cloud Anda. Gunakan AWS Health untuk memvisualisasikan dan menerima notifikasi tentang peristiwa layanan terkini dan perubahan yang akan datang, seperti peristiwa siklus hidup yang direncanakan, sehingga Anda dapat mengambil langkah-langkah untuk mengurangi dampaknya.
    a. Buat notifikasi peristiwa AWS Health sesuai keperluan yang dikirim ke saluran email dan obrolan melalui Notifikasi Pengguna AWS serta integrasikan secara programatis dengan alat pemantauan dan peringatan Anda melalui Amazon EventBridge atau API AWS Health.
    b. Rencanakan dan lacak progres pada peristiwa kesehatan yang memerlukan tindakan dengan mengintegrasikan dengan manajemen perubahan atau alat ITSM (seperti Jira atau ServiceNow) yang mungkin sudah Anda gunakan melalui Amazon EventBridge atau API AWS Health.
    c. Jika Anda menggunakan AWS Organizations, aktifkan tampilan organisasi untuk AWS Health guna menggabungkan peristiwa AWS Health di seluruh akun.
5. Lakukan instrumentasi pada aplikasi Anda dengan AWS X-Ray: AWS X-Ray memberikan wawasan tentang bagaimana performa aplikasi dan dependensi yang mendasarinya. Dengan melacak permintaan dari awal hingga akhir, Anda dapat mengidentifikasi kemacetan atau kegagalan yang terjadi dalam layanan eksternal atau komponen yang diandalkan oleh aplikasi Anda.
6. Gunakan Amazon DevOps Guru: Layanan berbasis machine learning ini mengidentifikasi masalah operasional, memprediksi kapan masalah kritis mungkin terjadi, dan merekomendasikan tindakan spesifik yang harus diambil. Layanan ini sangat bermanfaat untuk mendapatkan wawasan tentang dependensi dan memastikan bahwa dependensi bukan merupakan sumber masalah operasional.
7. Pantau secara rutin: Terus pantau metrik dan log yang berkaitan dengan dependensi eksternal. Siapkan peringatan untuk perilaku tak terduga atau performa yang menurun.
8. Lakukan validasi setelah perubahan: Setiap kali ada pembaruan atau perubahan yang dilakukan pada salah satu dependensi eksternal, Anda harus melakukan validasi terhadap performa dan memeriksa keselarasannya dengan persyaratan-persyaratan aplikasi Anda.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Menentukan KPI beban kerja
- OPS04-BP02 Mengimplementasikan telemetri aplikasi
- OPS04-BP03 Mengimplementasikan telemetri aktivitas pengguna
- OPS04-BP05 Mengimplementasikan keterlacakan transaksi
- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti

Dokumen terkait:

- Panduan Pengguna Dasbor Health Pribadi Amazon
- Panduan Pengguna Monitor Internet AWS
- Panduan Developer AWS X-Ray
- Panduan Pengguna DevOps Guru AWS

Video terkait:

- Visibilitas tentang bagaimana masalah internet memengaruhi performa aplikasi
- Pengantar tentang Amazon DevOps Guru
- Mengelola peristiwa siklus hidup sumber daya dalam skala besar dengan AWS Health

Contoh terkait:

- AWS Health Aware
- Menggunakan Penyaringan Berbasis Tag untuk Mengelola Pemantauan dan Peringatan AWS Health dengan Skala Besar

OPS04-BP05 Melaksanakan penelusuran terdistribusi

Penelusuran terdistribusi menawarkan cara untuk memantau dan memvisualisasikan permintaan yang melintasi berbagai komponen sistem terdistribusi. Dengan menangkap data jejak dari berbagai

sumber dan menganalisisnya dalam tampilan terpadu, tim dapat lebih memahami bagaimana permintaan mengalir, di mana kemacetan terjadi, dan di mana upaya pengoptimalan harus difokuskan.

Hasil yang diinginkan: Dapatkan tampilan menyeluruh permintaan-permintaan yang mengalir melewati sistem terdistribusi Anda, sehingga akan memungkinkan Anda untuk melakukan debugging yang presisi, performa yang dioptimalkan, dan pengalaman pengguna yang lebih baik.

Anti-pola umum:

- Instrumentasi yang tidak konsisten: Tidak semua layanan yang ada dalam sebuah sistem terdistribusi diinstrumentasi untuk penelusuran.
- Mengabaikan latensi: Hanya berfokus pada kesalahan dan tidak mempertimbangkan latensi atau penurunan performa yang terjadi secara bertahap.

Manfaat menjalankan praktik terbaik ini:

- Gambaran umum sistem yang komprehensif: Memberikan visualisasi dari seluruh jalur permintaan, dari masuk hingga keluar.
- Debugging yang disempurnakan: Mengidentifikasi dengan cepat di mana kegagalan atau masalah performa terjadi.
- Pengalaman pengguna yang ditingkatkan: Melakukan pemantauan dan optimalisasi berdasarkan data pengguna aktual, yang akan memastikan bahwa sistem memenuhi tuntutan dunia nyata.

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

Panduan implementasi

Mulailah dengan mengidentifikasi semua elemen beban kerja Anda yang memerlukan instrumentasi. Setelah semua komponen diperhitungkan, manfaatkan alat seperti AWS X-Ray dan OpenTelemetry untuk mengumpulkan data jejak untuk analisis dengan alat seperti X-Ray dan Amazon CloudWatch ServiceLens Map. Terlibat dalam ulasan reguler dengan pengembang, dan lengkapi diskusi ini dengan alat seperti Amazon DevOps Guru, X-Ray Analytics, dan X-Ray Insights untuk membantu mengungkap temuan yang lebih dalam. Buatlah peringatan dari data jejak untuk memberikan notifikasi kapan hasil, sebagaimana didefinisikan dalam rencana pemantauan beban kerja, mengandung risiko.

Langkah-langkah implementasi

Cara mengimplementasikan penelusuran terdistribusi secara efektif:

1. Adopsi AWS X-Ray: Integrasikan X-Ray ke dalam aplikasi Anda untuk mendapatkan wawasan tentang perilakunya, memahami performanya, dan mengenali kemacetan. Manfaatkan Wawasan X-Ray untuk analisis jejak otomatis.
2. Instrumen layanan Anda: Verifikasi bahwa setiap layanan, dari AWS Lambdafungsi hingga EC2instance, mengirimkan data jejak. Semakin banyak layanan yang Anda instrumen, semakin jelas end-to-end tampilan.
3. Menggabungkan Pemantauan Pengguna CloudWatch Nyata dan pemantauan sintetis: Integrasikan Pemantauan Pengguna Nyata (RUM) dan pemantauan sintetis dengan X-Ray. Hal ini akan memungkinkan perekaman pengalaman pengguna dunia nyata dan simulasi interaksi pengguna untuk mengidentifikasi masalah-masalah potensial yang mungkin terjadi.
4. Gunakan CloudWatch agen: Agen dapat mengirim jejak dari X-Ray atau OpenTelemetry, meningkatkan kedalaman wawasan yang diperoleh.
5. Gunakan Amazon DevOps Guru: DevOps Guru menggunakan data dari X-Ray,, CloudWatch AWS Config, dan AWS CloudTrail untuk memberikan rekomendasi yang dapat ditindaklanjuti.
6. Lakukan analisis jejak: Tinjau data jejak secara rutin untuk membedakan pola, anomali, atau kemacetan yang dapat memengaruhi performa aplikasi Anda.
7. Siapkan peringatan: Konfigurasikan alarm CloudWatchuntuk pola yang tidak biasa atau latensi yang diperpanjang, memungkinkan pengalamatan masalah proaktif.
8. Peningkatan terus-menerus: Tinjau ulang strategi penelusuran Anda saat layanan ditambahkan atau dimodifikasi untuk menangkap semua titik data yang relevan.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS04-BP03 Menerapkan telemetri pengalaman pengguna
- OPS04-BP04 Mengimplementasikan telemetri dependensi

Dokumen terkait:

- AWS X-Ray Panduan Pengembang
- Panduan Pengguna CloudWatch agen Amazon
- Panduan Pengguna Amazon DevOps Guru

Video terkait:

- Gunakan AWS X-Ray Wawasan
- AWS di Air ft. Observabilitas: Amazon CloudWatch dan AWS X-Ray

Contoh terkait:

- Instrumentasi aplikasi Anda untuk AWS X-Ray

Desain operasi

Terapkan pendekatan yang meningkatkan aliran perubahan ke dalam produksi dan yang membantu pemfaktoran ulang, umpan balik cepat atas kualitas, dan perbaikan bug. Ini mempercepat perubahan yang bermanfaat memasuki produksi, membatasi masalah yang di-deploy, dan menyediakan identifikasi cepat serta perbaikan masalah akibat aktivitas deployment.

Di AWS, Anda dapat menampilkan keseluruhan beban kerja Anda (aplikasi, infrastruktur, kebijakan, tata kelola, dan operasi) dalam bentuk kode. Beban kerja Anda dapat ditetapkan dalam kode dan diperbarui menggunakan kode. Ini artinya Anda dapat menerapkan disiplin teknik yang sama yang Anda gunakan untuk kode aplikasi pada setiap elemen tumpukan Anda.

Praktik terbaik

- OPS05-BP01 Menggunakan kontrol versi
- OPS05-BP02 Menguji dan memvalidasi perubahan
- OPS05-BP03 Menggunakan sistem manajemen konfigurasi
- OPS05-BP04 Menggunakan sistem manajemen build dan deployment
- OPS05-BP05 Melakukan manajemen patch
- OPS05-BP06 Bagikan standar desain
- OPS05-BP07 Mengimplementasikan praktik untuk meningkatkan kualitas kode

- OPS05-BP08 Menggunakan beberapa lingkungan
- OPS05-BP09 Lakukan perubahan yang sering, kecil, dan reversibel
- OPS05-BP10 Mengotomatiskan integrasi dan deployment sepenuhnya

OPS05-BP01 Menggunakan kontrol versi

Gunakan kontrol versi untuk memungkinkan pelacakan perubahan dan rilis.

Banyak layanan AWS yang menawarkan kemampuan kontrol versi. Gunakan revisi atau sistem kontrol sumber seperti Git untuk mengelola kode dan artefak lainnya, seperti templat AWS CloudFormation yang dikendalikan versi untuk infrastruktur Anda.

Hasil yang diinginkan: Tim Anda berkolaborasi dalam menangani kode. Saat digabungkan, kode tersebut konsisten dan tidak ada perubahan yang hilang. Kesalahan mudah dibatalkan melalui penentuan versi yang benar.

Anti-pola umum:

- Anda telah mengembangkan dan menyimpan kode di stasiun kerja Anda. Anda mengalami kegagalan penyimpanan yang tidak dapat dipulihkan di stasiun kerja dan kemudian kode Anda hilang.
- Setelah menimpa kode yang ada dengan perubahan Anda, Anda dapat memulai ulang aplikasi namun aplikasi sudah tidak dapat beroperasi lagi. Anda tidak bisa membatalkan perubahan.
- Anda memiliki write lock pada file laporan yang perlu diedit orang lain. Mereka meminta Anda untuk berhenti mengerjakannya agar mereka bisa menyelesaikan tugas-tugas mereka.
- Tim penelitian Anda telah mengerjakan sebuah analisis mendetail yang membentuk pekerjaan mendatang Anda. Seseorang secara tidak sengaja menyimpan daftar belanjanya dan menimpa laporan akhir. Anda tidak bisa membatalkan perubahan dan harus membuat ulang laporan tersebut.

Manfaat menjalankan praktik terbaik ini: Dengan menggunakan kemampuan kontrol versi, Anda dapat dengan mudah kembali ke versi sebelumnya dengan status yang baik, dan membatasi risiko kehilangan aset.

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

Panduan implementasi

Pelihara aset di repositori dengan kontrol versi. Tindakan ini mendukung pelacakan perubahan, deployment versi baru, deteksi perubahan pada versi yang ada, dan pengembalian ke versi sebelumnya (misalnya, kembali ke versi dengan status yang baik apabila terjadi kegagalan). Integrasikan kemampuan kontrol versi dari sistem manajemen konfigurasi Anda ke dalam prosedur Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP04 Menggunakan sistem manajemen build dan deployment

Video terkait:

- AWS re:Invent 2023 - Bagaimana Lockheed Martin membangun perangkat lunak lebih cepat, didukung oleh DevSecOps
- AWS re:Invent 2023 - Bagaimana GitHub mengoperasionalkan AI untuk kolaborasi dan produktivitas tim

OPS05-BP02 Menguji dan memvalidasi perubahan

Setiap perubahan yang di-deploy harus diuji untuk menghindari kesalahan dalam lingkungan produksi. Praktik terbaik ini difokuskan untuk menguji perubahan-perubahan dari kontrol versi hingga build artefak. Di samping perubahan kode aplikasi, pengujian juga harus menyertakan infrastruktur, konfigurasi, kontrol keamanan, dan prosedur-prosedur operasi. Ada banyak bentuk pengujian, dari uji unit hingga analisis komponen perangkat lunak (SCA). Makin ke kiri pengujian dalam proses integrasi dan pengiriman perangkat lunak akan menghasilkan tingkat kepastian kualitas artefak yang lebih tinggi.

Organisasi Anda harus mengembangkan standar-standar pengujian untuk semua artefak perangkat lunak. Pengujian otomatis dapat mengurangi kerja yang melelahkan dan mencegah terjadinya kesalahan dalam pengujian manual. Uji manual mungkin diperlukan dalam beberapa kasus. Pengembang harus memiliki akses ke hasil uji otomatis untuk menciptakan loop umpan balik yang meningkatkan kualitas perangkat lunak.

Hasil yang diinginkan: Perubahan perangkat lunak Anda diuji sebelum dikirim. Pengembang memiliki akses ke hasil pengujian dan validasi. Organisasi Anda memiliki standar pengujian yang berlaku untuk semua perubahan perangkat lunak.

Anti-pola umum:

- Anda men-deploy perubahan perangkat lunak baru tanpa melakukan pengujian apa pun. Perangkat lunak gagal berjalan dalam lingkungan produksi, dan mengakibatkan matinya sistem.
- Grup keamanan baru di-deploy dengan AWS CloudFormation tanpa diuji di dalam sebuah lingkungan pra-produksi. Grup keamanan tersebut menjadikan aplikasi Anda tidak terjangkau oleh para pelanggan Anda.
- Sebuah metode diubah tanpa pengujian unit. Perangkat lunak gagal saat di-deploy ke lingkungan produksi.

Manfaat menerapkan praktik terbaik ini: Perubahan tingkat kegagalan deployment perangkat lunak berkurang. Kualitas perangkat lunak meningkat. Pengembang memiliki kesadaran yang lebih tinggi tentang kelayakan kode mereka. Kebijakan keamanan dapat diluncurkan dengan penuh keyakinan untuk mendukung kepatuhan organisasi. Perubahan infrastruktur, misalnya pembaruan kebijakan penskalaan otomatis, diuji di awal untuk memenuhi kebutuhan lalu lintas.

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

Panduan implementasi

Pengujian dilakukan pada semua perubahan, dari kode aplikasi hingga infrastruktur, sebagai bagian dari praktik integrasi berkelanjutan yang Anda lakukan. Hasil-hasil pengujian dipublikasikan sehingga pengembang memiliki umpan balik dengan cepat. Organisasi memiliki standar pengujian bahwa semua perubahan harus lulus.

Gunakan kekuatan AI generatif dengan Amazon Q Developer untuk meningkatkan produktivitas pengembang dan kualitas kode Anda. Amazon Q Developer menyertakan pembuatan saran kode (berdasarkan model bahasa besar), produksi pengujian unit (termasuk kondisi batas), dan peningkatan keamanan kode melalui deteksi dan perbaikan kerentanan keamanan.

Contoh pelanggan

Sebagai bagian dari pipeline integrasi berkelanjutan mereka, AnyCompany Retail melakukan beberapa jenis pengujian pada semua artefak perangkat lunak. Mereka mempraktikkan pengembangan yang didorong pengujian sehingga semua perangkat lunak memiliki pengujian-

pengujian unit. Begitu artefak dibangun, mereka kemudian menjalankan pengujian secara menyeluruh. Setelah pengujian putaran pertama selesai, mereka menjalankan pemindaian keamanan aplikasi statis, yang mencari kerentanan yang dikenali. Pengembang menerima pesan setelah setiap gerbang pengujian dilalui. Setelah semua pengujian selesai, artefak perangkat lunak kemudian disimpan di dalam sebuah repositori artefak.

Langkah-langkah implementasi

1. Bekerjalah dengan para pemangku kepentingan yang ada di organisasi Anda untuk mengembangkan sebuah standar pengujian untuk artefak perangkat lunak. Pengujian standar apa yang harus dilalui oleh semua artefak? Apakah ada persyaratan kepatuhan atau tata kelola yang harus disertakan di dalam cakupan pengujian? Apakah Anda perlu melakukan pengujian kualitas kode? Setelah pengujian selesai dilakukan, siapa yang perlu mengetahuinya?
    1. Arsitektur Referensi Pipeline Deployment AWS berisi daftar tepercaya untuk jenis-jenis pengujian yang dapat dilakukan pada artefak perangkat lunak sebagai bagian dari pipeline integrasi.
2. Instrumentasikan aplikasi Anda dengan pengujian-pengujian yang diperlukan berdasarkan standar pengujian perangkat lunak Anda. Setiap set pengujian harus diselesaikan dalam waktu kurang dari sepuluh menit. Pengujian harus berjalan sebagai bagian dari pipeline integrasi.
    a. Gunakan Amazon Q Developer, sebuah alat AI generatif yang dapat membantu Anda membuat kasus pengujian unit (termasuk ketentuan batas), yang dapat menghasilkan fungsi dengan menggunakan kode dan komentar, dan menerapkan algoritme terkenal.
    b. Gunakan Amazon CodeGuru Reviewer untuk melakukan pengujian kode aplikasi Anda apakah ada cacat.
    c. Anda dapat menggunakan AWS CodeBuild untuk melakukan pengujian pada artefak perangkat lunak.
    d. AWS CodePipeline dapat mengorkestrasi pengujian perangkat lunak Anda ke dalam pipeline.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP01 Menggunakan kontrol versi
- OPS05-BP06 Membagikan standar desain
- OPS05-BP07 Mengimplementasikan praktik untuk meningkatkan kualitas kode
- OPS05-BP10 Mengotomatiskan integrasi dan deployment sepenuhnya

Dokumen terkait:

- Adopsi pendekatan pengembangan berbasis pengujian
- Mengakselerasi Siklus Hidup Pengembangan Perangkat Lunak Anda dengan Amazon Q
- Amazon Q Developer, sekarang tersedia secara umum, menyertakan pratinjau kemampuan-kemampuan baru untuk menata kembali pengalaman pengembang
- Lembar Cheat Utama untuk Menggunakan Amazon Q Developer di IDE Anda
- Beban kerja Shift-Left, memanfaatkan AI untuk Pembuatan Uji
- Pusat Amazon Q Developer
- 10 cara untuk membangun aplikasi lebih cepat dengan menggunakan Amazon CodeWhisperer
- Melihat melampaui cakupan kode dengan Amazon CodeWhisperer
- Praktik Terbaik untuk Rekayasa Perintah dengan Amazon CodeWhisperer
- Pipeline Pengujian AWS CloudFormation Otomatis dengan TaskCat dan CodePipeline
- Membangun pipeline CI/CD DevSecOps AWS yang menyeluruh dengan alat-alat SCA, SAST, dan DAST sumber terbuka
- Memulai pengujian aplikasi nirserver
- Pipeline CI/CD adalah pemandu utama rilisan saya
- Laporan Resmi tentang Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS

Video terkait:

- Menerapkan API dengan Amazon Q Developer Agent untuk Pengembangan Perangkat Lunak
- Menginstal, Mengonfigurasi, & Menggunakan Amazon Q Developer dengan JetBrains IDE (Cara)
- Menguasai seni Amazon CodeWhisperer - Daftar putar YouTube
- AWS re:Invent 2020: Infrastruktur yang dapat diuji: Pengujian integrasi di AWS
- AWS Summit ANZ 2021 - Mendorong strategi yang mengutamakan pengujian dengan CDK dan pengembangan yang didorong pengujian
- Menguji Infrastruktur Anda sebagai Kode dengan AWS CDK

Sumber daya terkait:

- Arsitektur Referensi Pipeline Deployment AWS - Aplikasi

- Pipeline DevSecOps Kubernetes AWS
- Menjalankan pengujian unit untuk aplikasi Node.js dari GitHub dengan menggunakan AWS CodeBuild
- Menggunakan Serverspec untuk pengembangan kode infrastruktur yang didorong pengujian

Layanan terkait:

- Amazon Q Developer
- Amazon CodeGuru Reviewer
- AWS CodeBuild
- AWS CodePipeline

OPS05-BP03 Menggunakan sistem manajemen konfigurasi

Gunakan sistem manajemen konfigurasi untuk membuat dan melacak perubahan konfigurasi. Sistem ini mengurangi kesalahan yang disebabkan oleh proses manual dan meminimalkan tingkat upaya untuk melakukan deployment perubahan.

Manajemen konfigurasi statis menetapkan nilai saat melakukan inisialisasi atas sebuah sumber daya yang diharapkan tetap konsisten selama masa pakai sumber daya tersebut. Manajemen konfigurasi dinamis menetapkan nilai saat inisialisasi. Nilai ini dapat atau diharapkan berubah selama masa pakai sumber daya. Misalnya, Anda dapat mengatur sebuah pengalih fitur untuk mengaktifkan fungsionalitas dalam kode Anda melalui sebuah perubahan konfigurasi, atau mengubah tingkat kerincian log selama insiden.

Konfigurasi harus di-deploy dalam status yang diketahui dan konsisten. Anda harus menggunakan inspeksi otomatis untuk selalu memantau konfigurasi sumber daya di seluruh lingkungan dan wilayah. Kontrol ini harus didefinisikan sebagai kode dan manajemen otomatis untuk memastikan aturan diterapkan secara konsisten di seluruh lingkungan. Perubahan konfigurasi harus diperbarui melalui prosedur kontrol perubahan yang disepakati dan diterapkan secara konsisten, dengan tetap memperhatikan kontrol versi. Konfigurasi aplikasi harus dikelola secara independen dan tidak bergantung pada kode aplikasi dan infrastruktur. Hal ini memungkinkan deployment yang konsisten di banyak lingkungan. Perubahan konfigurasi tidak akan mengakibatkan pembangunan kembali atau pemindahan aplikasi.

Hasil yang diinginkan: Anda mengonfigurasi, memvalidasi, dan melakukan deployment sebagai bagian dari pipeline integrasi berkelanjutan, pengiriman berkelanjutan (CI/CD) Anda. Anda memantau

untuk memvalidasi bahwa konfigurasi sudah benar. Hal ini akan meminimalkan dampak apa pun yang terjadi terhadap pelanggan dan pengguna akhir.

Anti-pola umum:

- Anda memperbarui konfigurasi server web secara manual di seluruh armada dan beberapa server menjadi tidak responsif karena terjadinya kesalahan pembaruan.
- Anda secara manual memperbarui armada server aplikasi Anda selama berjam-jam. Ketidaksesuaian dalam konfigurasi selama terjadi perubahan dapat menyebabkan perilaku yang tak terduga.
- Seseorang telah memperbarui grup keamanan Anda dan server web Anda tidak dapat diakses lagi. Tanpa mengetahui apa yang telah diubah, Anda menghabiskan banyak waktu untuk menyelidiki masalah tersebut sehingga waktu pemulihan akan semakin panjang.
- Anda mendorong konfigurasi pra-produksi ke dalam lingkungan produksi melalui CI/CD tanpa melakukan validasi. Anda mengekspos pengguna dan pelanggan ke data dan layanan yang salah.

Manfaat menerapkan praktik terbaik ini: Mengadopsi sistem manajemen konfigurasi akan meminimalkan tingkat upaya untuk membuat dan melacak perubahan, serta akan mengurangi frekuensi kesalahan yang disebabkan oleh penggunaan prosedur manual. Sistem manajemen konfigurasi dapat memberikan jaminan sehubungan dengan persyaratan tata kelola, kepatuhan, dan peraturan.

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

Panduan implementasi

Sistem manajemen konfigurasi digunakan untuk melacak dan mengimplementasikan perubahan-perubahan yang dibuat pada konfigurasi aplikasi dan lingkungan. Sistem manajemen konfigurasi juga digunakan untuk mengurangi terjadinya kesalahan yang disebabkan oleh proses-proses yang diselesaikan secara manual, membuat perubahan konfigurasi menjadi dapat diulang dan dapat diaudit, serta mengurangi tingkat upaya.

Pada AWS, Anda dapat menggunakan AWS Config untuk terus memantau konfigurasi sumber daya AWS Anda di seluruh akun dan Wilayah. Dengan demikian, Anda dapat melacak riwayat konfigurasi mereka, memahami bagaimana perubahan konfigurasi akan memengaruhi sumber daya lainnya, dan mengauditnya terhadap konfigurasi yang diharapkan atau diinginkan dengan menggunakan Aturan AWS Config dan AWS Config Conformance Packs.

Untuk konfigurasi dinamis dalam aplikasi yang berjalan di instans Amazon EC2, AWS Lambda, kontainer, aplikasi seluler, atau perangkat IoT, Anda dapat menggunakan AWS AppConfig untuk mengonfigurasi, memvalidasi, men-deploy, dan memantaunya di seluruh lingkungan.

Langkah-langkah implementasi

1. Lakukan identifikasi pemilik konfigurasi.
   a. Buat agar para pemilik konfigurasi menyadari tentang kepatuhan, tata kelola, atau peraturan apa pun.
2. Lakukan identifikasi terhadap item-item konfigurasi dan hasil kerja.
   a. Item-item konfigurasi adalah semua konfigurasi aplikasi dan lingkungan yang dipengaruhi oleh sebuah deployment yang dilakukan di dalam pipeline CI/CD Anda.
   b. Hasil kerja antara lain kriteria keberhasilan, validasi, dan hal-hal yang harus dipantau.
3. Pilihlah alat-alat yang bisa digunakan untuk melakukan manajemen konfigurasi berdasarkan kebutuhan bisnis dan pipeline pengiriman Anda.
4. Pertimbangkan deployment tertimbang seperti deployment canary untuk perubahan-perubahan konfigurasi yang signifikan guna meminimalkan dampak konfigurasi yang salah.
5. Integrasikan manajemen konfigurasi Anda ke dalam pipeline CI/CD Anda.
6. Validasikan semua perubahan yang didorong.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS06-BP01 Rencana untuk perubahan yang gagal
- OPS06-BP02 Uji penerapan
- OPS06-BP03 Menggunakan strategi deployment yang aman
- OPS06-BP04 Mengotomatiskan pengujian dan rollback

Dokumen terkait:

- AWS Control Tower
- Akselerator Zona Landasan AWS
- AWS Config
- Apa itu AWS Config?

- AWS AppConfig
- Apa itu AWS CloudFormation?
- Alat Pengembang AWS
- AWS CodeBuild
- AWS CodePipeline
- AWS CodeDeploy

Video terkait:

- AWS re:Invent 2022 - Tata kelola dan kepatuhan proaktif untuk beban kerja AWS
- AWS re:Invent 2020: Capai kepatuhan sebagai kode dengan menggunakan AWS Config
- Kelola dan Deploy Konfigurasi Aplikasi dengan AWS AppConfig

OPS05-BP04 Menggunakan sistem manajemen build dan deployment

Gunakan sistem manajemen build dan deployment. Sistem ini mengurangi kesalahan yang disebabkan oleh proses manual dan meminimalkan tingkat upaya untuk melakukan deployment perubahan.

Di AWS, Anda dapat membangun pipeline integrasi berkelanjutan dan deployment berkelanjutan (CI/CD) dengan menggunakan layanan-layanan seperti Alat Developer AWS (misalnya AWS CodeBuild, AWS CodePipeline, dan AWS CodeDeploy).

Hasil yang diinginkan: Sistem manajemen build dan deployment Anda mendukung sistem integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) yang ada di organisasi Anda yang menyediakan kemampuan-kemampuan untuk mengotomatisasi peluncuran yang aman dengan konfigurasi yang benar.

Anti-pola umum:

- Setelah menyusun kode Anda pada sistem pengembangan, Anda menyalin file yang dapat dieksekusi ke sistem produksi namun file tersebut gagal memulai. File log lokal mengindikasikan bahwa kegagalan tersebut dikarenakan terjadinya kehilangan dependensi.
- Anda berhasil membangun aplikasi Anda dengan fitur-fitur baru pada lingkungan pengembangan Anda dan memberikan kodenya ke tim jaminan kualitas (QA). Kode tersebut gagal dalam pengujian QA karena ada aset statis yang hilang.

- Pada hari Jumat, setelah melakukan upaya keras, Anda berhasil membangun aplikasi Anda secara manual di lingkungan pengembangan Anda termasuk fitur-fitur yang baru Anda kodekan. Pada hari Senin, Anda tidak dapat mengulangi langkah-langkah yang membuat Anda berhasil membangun aplikasi tersebut.
- Anda melakukan pengujian yang telah Anda siapkan untuk rilisan baru Anda. Kemudian Anda menghabiskan minggu selanjutnya untuk mempersiapkan lingkungan pengujian dan melakukan seluruh pengujian integrasi yang ada sekarang disusul dengan pengujian kinerja. Kode baru tersebut ternyata memiliki dampak kinerja yang tidak dapat diterima dan harus dikembangkan ulang dan kemudian diuji ulang.

Manfaat menjalankan praktik terbaik ini: Dengan menyediakan mekanisme untuk mengatasi aktivitas build dan deployment, Anda mengurangi upaya yang diperlukan untuk melakukan tugas-tugas berulang, membebaskan anggota tim Anda untuk fokus pada tugas-tugas kreatif mereka yang berharga, serta mengurangi terjadinya kesalahan akibat pelaksanaan prosedur yang dilakukan secara manual.

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

Panduan implementasi

Sistem manajemen build dan deployment digunakan untuk melacak dan mengimplementasikan perubahan, mengurangi kesalahan yang disebabkan oleh proses yang dilakukan secara manual, dan mengurangi upaya yang diperlukan untuk melakukan deployment dengan aman. Lakukan otomatisasi sepenuhnya terhadap pipeline integrasi dan deployment dari check-in kode hingga build, pengujian, deployment, dan validasi. Hal ini akan mempersingkat waktu tunggu (lead time), mengurangi biaya, mendorong peningkatan frekuensi perubahan, mengurangi tingkat upaya, dan meningkatkan kolaborasi.

Langkah-langkah implementasi

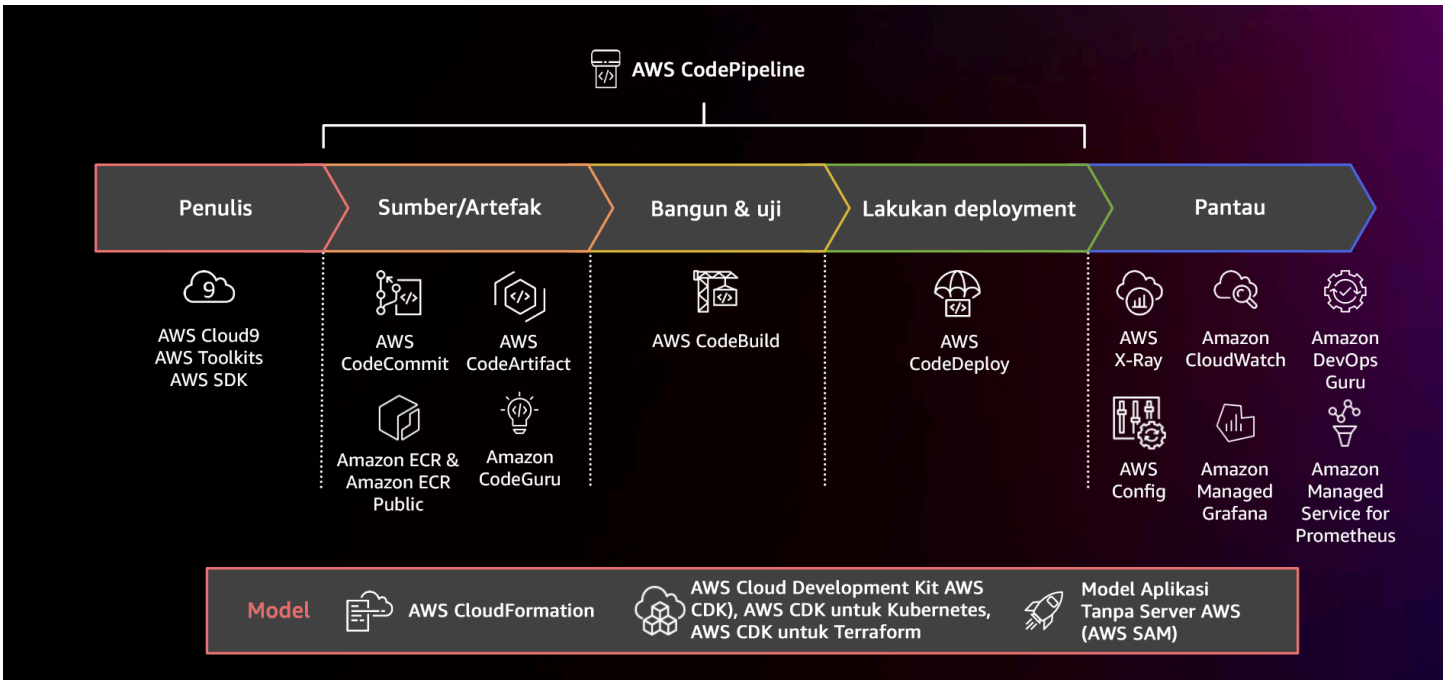


Diagram yang menunjukkan pipeline CI/CD menggunakan AWS CodePipeline dan layanan terkait

1. Gunakan sistem kontrol versi untuk menyimpan dan mengelola aset (seperti dokumen, kode sumber, dan file biner).
2. Gunakan CodeBuild untuk mengompilasikan kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap untuk di-deploy.
3. Gunakan CodeDeploy sebagai sebuah layanan deployment yang mengotomatiskan deployment aplikasi ke instans Amazon EC2, instans on-premise, fungsi AWS Lambda nirserver, atau Amazon ECS.
4. Pantau deployment Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS06-BP04 Mengotomatiskan pengujian dan rollback

Dokumen terkait:

- Alat Developer AWS

- Apa itu AWS CodeBuild?
- AWS CodeBuild
- Apa itu AWS CodeDeploy?

Video terkait:

- AWS re:Invent 2022 - Praktik terbaik AWS Well-Architected untuk DevOps di AWS

OPS05-BP05 Melakukan manajemen patch

Lakukan manajemen patch untuk mendapatkan fitur, menangani permasalahan, dan menjaga kepatuhan terhadap tata kelola. Otomatiskan manajemen patch untuk mengurangi kesalahan yang disebabkan oleh proses manual, menskalakan, dan mengurangi upaya untuk melakukan patch.

Manajemen patch dan kerentanan adalah bagian dari aktivitas manajemen manfaat dan risiko Anda. Lebih baik Anda memiliki infrastruktur tetap dan melakukan deployment beban kerja pada status yang diketahui baik dan terverifikasi. Jika tidak memungkinkan, opsi yang tersisa ialah menerapkan patching.

AWS Health adalah sumber informasi otoritatif tentang peristiwa siklus hidup yang direncanakan dan peristiwa lain yang memerlukan tindakan yang memengaruhi kesehatan sumber daya AWS Cloud Anda. Anda harus mengetahui perubahan dan pembaruan yang akan datang yang harus dilakukan. Peristiwa siklus hidup utama yang direncanakan dikirim setidaknya enam bulan sebelumnya.

Amazon EC2 Image Builder menyediakan pipeline untuk memperbarui image mesin. Sebagai bagian dari manajemen patch, pertimbangkan Amazon Machine Image (AMI) menggunakan pipeline image AMI atau image kontainer dengan pipeline citra Docker, sedangkan AWS Lambda menyediakan pola untuk runtime kustom dan pustaka patch untuk menghilangkan kerentanan.

Anda harus mengelola pembaruan untuk Amazon Machine Images untuk citra Linux atau Windows Server menggunakan Amazon EC2 Image Builder. Anda dapat menggunakan Amazon Elastic Container Registry (Amazon ECR) dengan pipeline yang ada sekarang untuk mengelola citra Amazon ECS dan mengelola citra Amazon EKS. Lambda menyertakan fitur manajemen versi.

Patching tidak boleh dilakukan pada sistem produksi tanpa mengujinya terlebih dahulu di sebuah lingkungan yang aman. Patch hanya bisa diterapkan jika mendukung hasil operasi atau bisnis. Pada AWS, Anda dapat menggunakan AWS Systems Manager Patch Manager untuk mengotomatiskan

proses patching sistem terkelola dan menjadwalkan aktivitas menggunakan Systems Manager Maintenance Windows.

Hasil yang diinginkan: Citra AMI dan kontainer Anda diberikan patch, diperbarui, dan siap diluncurkan. Anda dapat melacak status dari semua citra yang di-deploy dan mengetahui kepatuhan patch. Anda dapat melaporkan status saat ini dan memiliki proses untuk memenuhi kebutuhan-kebutuhan kepatuhan Anda.

Anti-pola umum:

- Anda diberi tugas untuk menerapkan semua patch keamanan baru dalam waktu dua jam yang menyebabkan terjadinya beberapa kali pemadaman akibat ketidaksesuaian aplikasi dengan patch.
- Pustaka yang tidak di-patch dapat menimbulkan konsekuensi yang tidak dinginkan karena pihak yang tidak diketahui memanfaatkan kerentanan di dalamnya untuk mengakses beban kerja Anda.
- Anda melakukan patching pada lingkungan pengembangan secara otomatis tanpa memberikan notifikasi kepada developer. Anda menerima beberapa keluhan dari pengembang bahwa lingkungan mereka tidak lagi beroperasi sesuai dengan yang diharapkan.
- Anda belum menerapkan patch pada perangkat lunak komersial siap pakai di sebuah instans tetap. Ketika Anda mengalami masalah pada perangkat lunak dan menghubungi vendornya, Anda diberi tahu bahwa versi tersebut tidak didukung dan Anda harus melakukan patching pada tingkat tertentu untuk menerima bantuan.
- Patch yang baru-baru ini dirilis untuk perangkat lunak enkripsi yang Anda gunakan memiliki peningkatan performa yang signifikan. Sistem Anda yang tidak di-patching tetap memiliki masalah performa akibat tidak dilakukannya patching.
- Anda mendapatkan notifikasi tentang kerentanan zero-day yang memerlukan perbaikan darurat dan Anda harus menerapkan patching pada semua lingkungan Anda secara manual.
- Anda tidak mengetahui tindakan krusial yang diperlukan untuk memelihara sumber daya Anda, seperti pembaruan versi wajib, karena Anda tidak meninjau peristiwa siklus hidup terencana mendatang dan informasi lainnya. Anda kehilangan waktu krusial untuk perencanaan dan pelaksanaan, sehingga mengakibatkan perubahan darurat untuk tim Anda dan potensi dampak atau waktu henti yang tidak terduga.

Manfaat menerapkan praktik terbaik ini: Dengan menjalankan proses manajemen patching, termasuk kriteria Anda untuk patching dan metodologi untuk distribusi ke seluruh lingkungan Anda, Anda dapat menskalakan dan melaporkan tingkat patching Anda. Ini memberikan jaminan seputar patching keamanan dan memastikan visibilitas yang jelas tentang status perbaikan yang diketahui yang

sekarang sedang dilakukan. Hal ini mendorong adopsi fitur dan kemampuan yang diinginkan, penyingkiran masalah secara cepat, dan kepatuhan yang berkelanjutan terhadap tata kelola. Implementasikan sistem manajemen dan otomatisasi patching untuk mengurangi tingkat upaya untuk men-deploy patch dan mengurangi kesalahan yang disebabkan oleh proses yang dilakukan secara manual.

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

Panduan implementasi

Lakukan patching pada sistem untuk menyelesaikan masalah, untuk mendapatkan fitur atau kemampuan yang diinginkan, dan untuk tetap patuh terhadap kebijakan tata kelola serta persyaratan dukungan vendor. Pada sistem tetap, lakukan deployment dengan rangkaian patching yang sesuai untuk mencapai hasil yang diinginkan. Lakukan otomatisasi mekanisme manajemen patching untuk mengurangi waktu yang telah berlalu untuk melakukan patching, untuk mencegah kesalahan yang disebabkan oleh proses manual, dan mengurangi upaya dalam melakukan patching.

Langkah-langkah implementasi

Untuk Amazon EC2 Image Builder:

1. Dengan menggunakan Amazon EC2 Image Builder, tentukan detail pipeline:
    a. Buatlah sebuah pipeline citra dan beri nama
    b. Tentukan jadwal pipeline dan zona waktu
    c. Konfigurasikan dependensi apa pun
2. Pilih resep:
    a. Pilih resep yang sudah ada atau buat resep baru
    b. Pilih jenis citra
    c. Beri nama dan versi resep Anda
    d. Pilih citra dasar Anda
    e. Tambahkan komponen build dan tambahkan ke registri target
3. Opsional - tentukan konfigurasi infrastruktur Anda.
4. Opsional - tentukan pengaturan konfigurasi.
5. Tinjau pengaturan.
6. Pertahankan kebersihan resep secara teratur.

Untuk Systems Manager Patch Manager:

1. Buat acuan dasar patch.
2. Pilih metode operasi patching.
3. Aktifkan pelaporan dan pemindaian kepatuhan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS06-BP04 Mengotomatiskan pengujian dan rollback

Dokumen terkait:

- Apa itu Amazon EC2 Image Builder
- Buat pipeline citra dengan menggunakan Amazon EC2 Image Builder
- Buat pipeline citra kontainer
- AWS Systems Manager Patch Manager
- Menggunakan Patch Manager
- Menggunakan laporan kepatuhan patch
- Alat Developer AWS

Video terkait:

- CI/CD untuk Aplikasi Nirserver di AWS
- Mendesain dengan Mempertimbangkan Operasional

Contoh terkait:

- Tutorial AWS Systems Manager Patch Manager

OPS05-BP06 Bagikan standar desain

Bagikan praktik terbaik kepada seluruh tim untuk meningkatkan kesadaran dan memaksimalkan manfaat dari upaya-upaya pengembangan. Dokumentasikan dan jaga agar hal ini selalu mutakhir seiring perkembangan yang terjadi pada arsitektur Anda. Jika standar bersama telah diterapkan di

dalam organisasi Anda, tersedianya mekanisme merupakan hal yang sangat penting untuk meminta penambahan, perubahan, dan pengecualian terhadap standar. Tanpa opsi ini, standar akan menjadi penghambat inovasi.

Hasil yang diinginkan: Standar desain dibagikan ke semua tim yang ada dalam organisasi Anda. Mereka didokumentasikan dan disimpan up-to-date sebagai praktik terbaik berkembang.

Anti-pola umum:

- Dua tim pengembangan masing-masing telah membuat sebuah layanan autentikasi pengguna. Pengguna Anda harus mempertahankan rangkaian kredensial terpisah untuk masing-masing bagian sistem yang ingin diakses.
- Setiap tim mengelola infrastruktur mereka sendiri. Persyaratan kepatuhan baru memaksakan penerapan sebuah perubahan pada infrastruktur Anda dan setiap tim mengimplementasikannya dengan cara yang berbeda.

Manfaat menerapkan praktik terbaik ini: Penggunaan standar bersama akan mendukung adopsi praktik terbaik dan memaksimalkan manfaat dari upaya-upaya pengembangan yang dilakukan. Mendokumentasikan dan memperbarui standar desain membuat organisasi Anda tetap up-to-date dengan praktik terbaik serta persyaratan keamanan dan kepatuhan.

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

Panduan implementasi

Berbagi praktik terbaik, standar desain, daftar periksa, prosedur operasi, panduan, dan persyaratan tata kelola yang ada sekarang dengan semua tim. Buatlah prosedur-prosedur untuk meminta perubahan, penambahan, dan pengecualian standar desain untuk mendukung peningkatan dan inovasi. Buat tim mengetahui tentang konten yang dipublikasikan. Memiliki mekanisme untuk menjaga standar desain up-to-date saat praktik terbaik baru muncul.

Contoh pelanggan

AnyCompany Retail memiliki tim arsitektur lintas fungsi yang menciptakan pola arsitektur perangkat lunak. Tim ini membangun arsitektur dengan kepatuhan dan tata kelola bawaan. Tim yang mengadopsi standar bersama ini mendapatkan manfaat dari memiliki kepatuhan dan tata kelola bawaan. Mereka dapat membangun di atas standar desain dengan cepat. Tim arsitektur mengadakan pertemuan setiap tiga bulan untuk mengevaluasi pola-pola arsitektur dan melakukan pembaruan, jika perlu.

Langkah-langkah implementasi

1. Identifikasikan sebuah tim lintas fungsi yang memegang kepemilikan atas pengembangan dan pembaruan standar desain. Tim ini harus bekerja sama dengan para pemangku kepentingan yang ada di seluruh organisasi Anda untuk mengembangkan standar desain, standar operasi, daftar periksa, panduan, dan persyaratan tata kelola. Dokumentasikan standar desain dan bagikan dalam organisasi Anda.
   a. AWS Service Catalog dapat digunakan untuk membuat portofolio yang mewakili standar desain dengan menggunakan infrastruktur sebagai kode. Anda dapat berbagi portofolio dengan semua akun.
2. Miliki mekanisme untuk menjaga standar desain up-to-date karena praktik terbaik baru diidentifikasi.
3. Jika standar desain diterapkan secara terpusat, Anda harus memiliki proses untuk meminta perubahan, pembaruan, dan pengecualian.

Tingkat upaya untuk rencana implementasi: Sedang. Untuk mengembangkan sebuah proses untuk membuat dan berbagi standar desain mungkin diperlukan kerja sama dan koordinasi dengan para pemangku kepentingan yang ada di seluruh organisasi Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP03 Mengevaluasi persyaratan tata kelola - Persyaratan tata kelola memengaruhi standar desain.
- OPS01-BP04 Evaluasi persyaratan kepatuhan - Kepatuhan adalah input penting dalam membuat standar desain.
- OPS07-BP02 Memastikan peninjauan yang konsisten terkait kesiapan operasional - Daftar periksa kesiapan operasional merupakan sebuah mekanisme untuk mengimplementasikan standar desain ketika Anda merancang desain beban kerja Anda.
- OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan - Memperbarui standar desain merupakan bagian dari peningkatan berkelanjutan.
- OPS11-BP04 Melakukan manajemen pengetahuan - Sebagai bagian dari praktik manajemen pengetahuan Anda, dokumentasikan dan bagikan standar desain.

Dokumen terkait:

- Otomatiskan AWS Backup s dengan AWS Service Catalog
- AWS Service Catalog Akun Factory-Enhanced
- Bagaimana Expedia Group membangun Database as a Service (DBaaS) menawarkan menggunakan AWS Service Catalog
- Mempertahankan visibilitas tentang penggunaan pola-pola arsitektur cloud
- Sederhanakan berbagi AWS Service Catalog portofolio Anda dalam pengaturan AWS Organizations

Video terkait:

- AWS Service Catalog — Memulai
- AWS re:invent 2020: Kelola AWS Service Catalog portofolio Anda seperti seorang ahli

Contoh terkait:

- AWS Service Catalog Arsitektur Referensi
- AWS Service Catalog Lokakarya

Layanan terkait:

- AWS Service Catalog

OPS05-BP07 Mengimplementasikan praktik untuk meningkatkan kualitas kode

Implementasikan praktik untuk meningkatkan kualitas kode dan meminimalkan kecacatan. Beberapa contohnya termasuk, pengembangan yang didorong pengujian, peninjauan kode, pengadopsian standar, dan pemrograman berpasangan. Sertakan praktik-praktik ini ke dalam integrasi berkelanjutan dan proses penyampaian hasil Anda.

Hasil yang diinginkan: Organisasi Anda menggunakan praktik-praktik terbaik seperti peninjauan kode atau pemrograman berpasangan untuk meningkatkan kualitas kode. Pengembang dan operator mengadopsi praktik-praktik terbaik dalam kualitas kode sebagai bagian dari siklus hidup pengembangan perangkat lunak.

Anti-pola umum:

- Anda mempercayakan kode ke cabang utama aplikasi tanpa peninjauan kode. Perubahan otomatis melakukan deployment ke lingkungan produksi dan menyebabkan penghentian produksi.
- Aplikasi baru dikembangkan tanpa melakukan pengujian integrasi, unit, atau pengujian menyeluruh. Tidak ada cara untuk menguji aplikasi sebelum deployment.
- Tim Anda membuat perubahan-perubahan manual pada lingkungan produksi untuk mengatasi kecacatan. Perubahan tidak melalui proses pengujian atau peninjauan kode dan tidak direkam atau dicatat log-nya melalui proses penyampaian hasil dan integrasi berkelanjutan.

Manfaat menerapkan praktik terbaik ini: Dengan mengadopsi praktik-praktik untuk meningkatkan kualitas kode, Anda dapat membantu meminimalkan masalah-masalah yang terjadi di lingkungan produksi. Praktik-praktik terbaik kualitas kode mencakup pemrograman berpasangan, tinjauan kode, dan implementasi alat-alat produktivitas AI.

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

Panduan implementasi

Implementasikan praktik-praktik untuk meningkatkan kualitas kode guna meminimalkan terjadinya kecacatan sebelum dilakukan deployment terhadapnya. Gunakan praktik-praktik, misalnya pengembangan berbasis pengujian, peninjauan kode, dan pemrograman berpasangan, untuk meningkatkan kualitas pengembangan Anda.

Gunakan kekuatan AI generatif dengan Amazon Q Developer untuk meningkatkan produktivitas pengembang dan kualitas kode Anda. Amazon Q Developer menyertakan pembuatan saran kode (berdasarkan model bahasa besar), produksi pengujian unit (termasuk kondisi batas), dan peningkatan keamanan kode melalui deteksi dan perbaikan kerentanan keamanan.

Contoh pelanggan

AnyCompany Retail mengadopsi beberapa praktik untuk meningkatkan kualitas kode. Mereka telah mengadopsi pengembangan berbasis pengujian sebagai standar untuk menulis aplikasi. Untuk beberapa fitur baru, pengembang mereka akan memasangkan program menjadi satu saat proses sprint. Setiap permintaan penarikan (pull request) akan melewati peninjauan kode oleh pengembang senior sebelum diintegrasikan dan dilakukan deployment.

Langkah-langkah implementasi

1. Adopsi praktik-praktik kualitas kode, misalnya praktik pengembangan berbasis pengujian, peninjauan kode, dan pemrograman berpasangan, ke dalam proses penyampaian hasil dan

integrasi berkelanjutan Anda. Gunakan teknik-teknik ini untuk meningkatkan kualitas perangkat lunak.

a. Gunakan Amazon Q Developer, sebuah alat AI generatif yang dapat membantu Anda membuat kasus pengujian unit (termasuk ketentuan batas), yang dapat menghasilkan fungsi menggunakan kode dan komentar, menerapkan algoritme terkenal, mendeteksi pelanggaran kebijakan keamanan dan kerentanan dalam kode Anda, mendeteksi rahasia, memindai infrastruktur sebagai kode (IaC), kode dokumen, dan mempelajari pustaka kode pihak ketiga dengan lebih cepat.
b. Amazon CodeGuru Reviewer dapat memberikan rekomendasi pemrograman untuk kode Python dan Java menggunakan machine learning.

Tingkat upaya untuk rencana implementasi: Sedang. Ada banyak cara untuk mengimplementasikan praktik-praktik terbaik ini, tetapi membuat organisasi mau mengadopsinya mungkin akan menjadi hal yang sulit.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP02 Menguji dan memvalidasi perubahan
- OPS05-BP06 Membagikan standar desain

Dokumen terkait:

- Adopsi pendekatan pengembangan berbasis pengujian
- Mengakselerasi Siklus Hidup Pengembangan Perangkat Lunak Anda dengan Amazon Q
- Amazon Q Developer, sekarang tersedia secara umum, menyertakan pratinjau kemampuan-kemampuan baru untuk menata kembali pengalaman pengembang
- Lembar Cheat Utama untuk Menggunakan Amazon Q Developer di IDE Anda
- Beban kerja Shift-Left, memanfaatkan AI untuk Pembuatan Uji
- Pusat Amazon Q Developer
- 10 cara untuk membangun aplikasi lebih cepat dengan menggunakan Amazon CodeWhisperer
- Melihat melampaui cakupan kode dengan Amazon CodeWhisperer
- Praktik Terbaik untuk Rekayasa Perintah dengan Amazon CodeWhisperer
- Panduan Perangkat Lunak Tangkas

- Pipeline CI/CD adalah pemandu utama rilisan saya
- Lakukan otomatisasi peninjauan kode dengan Amazon CodeGuru Reviewer
- Adopsi pendekatan pengembangan berbasis pengujian
- Bagaimana DevFactory membangun aplikasi yang lebih baik dengan Amazon CodeGuru
- Tentang Pemrograman Berpasangan
- RENGA Inc. mengotomatiskan peninjauan kode dengan Amazon CodeGuru
- Seni Pengembangan Tangkas: Pengembangan yang Didorong Pengujian
- Mengapa peninjauan kode itu penting (dan sesungguhnya menghemat waktu!)

Video terkait:

- Menerapkan API dengan Amazon Q Developer Agent untuk Pengembangan Perangkat Lunak
- Menginstal, Mengonfigurasi, & Menggunakan Amazon Q Developer dengan JetBrains IDE (Cara)
- Menguasai seni Amazon CodeWhisperer - Daftar putar YouTube
- AWS re:Invent 2020: Peningkatan berkelanjutan kualitas kode dengan Amazon CodeGuru
- AWS Summit ANZ 2021 - Mendorong strategi yang mengutamakan pengujian dengan CDK dan pengembangan yang didorong pengujian

Layanan terkait:

- Amazon Q Developer
- Amazon CodeGuru Reviewer
- Amazon CodeGuru Profiler

OPS05-BP08 Menggunakan beberapa lingkungan

Gunakan beberapa lingkungan untuk bereksperimen, mengembangkan, dan menguji beban kerja Anda. Gunakan tingkat kontrol berjenjang seiring lingkungan mendekati tahap produksi untuk mendapatkan keyakinan bahwa beban kerja Anda beroperasi sesuai keinginan ketika di-deploy.

Hasil yang diinginkan: Anda memiliki beberapa lingkungan yang mencerminkan kebutuhan-kebutuhan kepatuhan dan tata kelola Anda. Anda menguji dan mempromosikan kode melalui lingkungan-lingkungan yang ada di jalur Anda menuju produksi.

1. Organisasi Anda melakukan hal ini melalui pembentukan zona landasan, yang menyediakan tata kelola, kontrol, otomatisasi akun, jaringan, keamanan, dan observabilitas operasional. Kelola kemampuan zona landasan ini menggunakan beberapa lingkungan. Contoh umumnya adalah organisasi sandbox untuk mengembangkan dan menguji perubahan pada zona landasan berbasis AWS Control Tower, yang mencakup AWS IAM Identity Center dan kebijakan seperti kebijakan kontrol layanan (SCP). Semua elemen ini dapat secara signifikan memengaruhi akses dan pengoperasian Akun AWS di dalam zona landasan.
2. Selain layanan ini, tim Anda memperluas kemampuan zona landasan dengan solusi yang diterbitkan oleh AWS dan partner AWS atau sebagai solusi kustom yang dikembangkan dalam organisasi Anda. Contoh solusi yang diterbitkan oleh AWS termasuk Customizations for AWS Control Tower (CfCT) dan AWS Control Tower Account Factory for Terraform (AFT).
3. Organisasi Anda menerapkan prinsip pengujian, promosi kode, dan perubahan kebijakan yang sama untuk zona landasan di semua lingkungan dalam jalur Anda menuju produksi. Strategi ini menyediakan lingkungan zona landasan yang stabil dan aman untuk tim aplikasi dan beban kerja Anda.

Anti-pola umum:

- Anda sedang melakukan pengembangan di sebuah lingkungan pengembangan bersama dan developer lain menimpa perubahan kode Anda.
- Kontrol keamanan terbatas di lingkungan pengembangan bersama Anda melarang Anda untuk melakukan eksperimen dengan layanan dan fitur baru.
- Anda melakukan pengujian beban pada sistem produksi Anda dan menyebabkan terjadinya pemadaman (outage) terhadap pengguna Anda.
- Kesalahan fatal yang menyebabkan hilangnya data terjadi di lingkungan produksi. Di lingkungan produksi, Anda mencoba membuat ulang kondisi yang menyebabkan data hilang tersebut sehingga Anda dapat mengidentifikasi bagaimana hal tersebut bisa terjadi dan mencegahnya agar tidak terjadi kembali. Untuk mencegah kejadian hilang data lainnya selama proses pengujian, Anda terpaksa menjadikan aplikasi tidak tersedia untuk pengguna.
- Anda mengoperasikan layanan multi-tenant dan tidak dapat mendukung permintaan lingkungan khusus yang diajukan oleh pelanggan.
- Anda mungkin tidak selalu melakukan pengujian, tetapi ketika Anda melakukannya, Anda melakukan pengujian tersebut di lingkungan produksi.
- Anda percaya bahwa dengan satu lingkungan tunggal, cakupan dampak perubahannya hanya terjadi di dalam lingkungan tersebut.

- Anda meningkatkan kemampuan zona landasan penting, tetapi perubahan tersebut mengganggu kemampuan tim Anda untuk menyediakan akun bagi proyek baru atau beban kerja yang sudah ada.
- Anda menerapkan kontrol baru pada Akun AWS Anda, tetapi perubahan tersebut memengaruhi kemampuan tim beban kerja Anda untuk menerapkan perubahan di dalam Akun AWS mereka.

Manfaat menjalankan praktik terbaik ini: Saat Anda melakukan deployment beberapa lingkungan, Anda dapat mendukung beberapa lingkungan pengembangan, pengujian, dan produksi secara serentak tanpa menimbulkan konflik di antara developer atau komunitas pengguna. Untuk kemampuan kompleks seperti zona landasan, hal ini secara signifikan mengurangi risiko perubahan, menyederhanakan proses perbaikan, dan mengurangi risiko pembaruan penting terhadap lingkungan. Organisasi yang menggunakan zona landasan secara otomatis mendapatkan manfaat dari penggunaan strategi multi-akun di lingkungan AWS mereka, dengan struktur akun, tata kelola, jaringan, dan konfigurasi keamanan. Seiring organisasi Anda berkembang dari waktu ke waktu, zona landasan dapat berevolusi untuk mengamankan serta mengatur beban kerja dan sumber daya Anda.

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

Panduan implementasi

Gunakan beberapa lingkungan dan sediakan lingkungan sandbox developer yang memiliki kontrol minimum untuk membantu eksperimen. Sediakan lingkungan pengembangan individu untuk membantu melakukan pekerjaan secara paralel, sehingga ketangkasan pengembangan akan meningkat. Implementasikan kontrol yang lebih kuat di lingkungan tersebut ketika mendekati produksi agar developer dapat membuat inovasi. Gunakan infrastruktur sebagai kode dan sistem manajemen konfigurasi untuk men-deploy lingkungan yang dikonfigurasi sesuai dengan kontrol yang ada di dalam lingkungan produksi guna memastikan sistem beroperasi sesuai keinginan saat di-deploy. Saat lingkungan tidak digunakan, nonaktifkan lingkungan tersebut untuk menghindari timbulnya biaya-biaya terkait sumber daya tidak terpakai (misalnya sistem pengembangan di malam hari dan di akhir pekan). Deploy lingkungan setara produksi saat melakukan pengujian beban untuk meningkatkan hasil yang valid.

Tim seperti rekayasa platform, jaringan, dan operasi keamanan sering mengelola kemampuan di tingkat organisasi dengan persyaratan yang berbeda. Pemisahan akun saja tidak cukup untuk menyediakan dan memelihara lingkungan terpisah untuk eksperimen, pengembangan, dan pengujian. Dalam kasus seperti itu, buat instans AWS Organizations terpisah.

Sumber daya

Dokumen terkait:

- Penjadwal Instans di AWS
- Apa itu AWS CloudFormation?
- Mengatur Lingkungan AWS Anda Menggunakan Beberapa Akun - Beberapa organisasi - Uji perubahan pada lingkungan AWS Anda secara keseluruhan
- Panduan AWS Control Tower

OPS05-BP09 Lakukan perubahan yang sering, kecil, dan reversibel

Gunakan perubahan yang sering, kecil, dan dapat dikembalikan untuk mengurangi cakupan perubahan. Ketika digunakan bersamaan dengan sistem manajemen perubahan, sistem manajemen konfigurasi, dan sistem build serta pengiriman, perubahan yang sering, kecil, dan dapat dikembalikan dapat mengurangi cakupan dan dampak perubahan. Hal ini menghasilkan pemecahan masalah yang lebih efektif dan remediasi yang lebih cepat dengan opsi untuk membatalkan perubahan.

Anti-pola umum:

- Anda melakukan deployment versi baru aplikasi Anda setiap tiga bulan sekali dengan periode perubahan yang mengharuskan layanan inti dinonaktifkan.
- Anda sering kali membuat perubahan pada skema basis data Anda tanpa melacak perubahan dalam sistem manajemen Anda terlebih dahulu.
- Anda melakukan pembaruan secara manual di tempat, menimpa instalasi dan konfigurasi yang ada, dan tidak memiliki rencana roll-back yang jelas.

Manfaat menerapkan praktik terbaik ini: Upaya pengembangan akan menjadi lebih cepat dengan menerapkan perubahan kecil dalam frekuensi yang rapat. Ketika perubahan itu berukuran kecil, perubahan tersebut akan jauh lebih mudah diidentifikasi jika terdapat konsekuensi yang tidak diinginkan, serta lebih mudah untuk dikembalikan. Ketika perubahan dapat dikembalikan, risiko implementasi perubahan menjadi lebih kecil karena pemulihannya lebih mudah diterapkan. Proses perubahan memiliki risiko yang lebih kecil dan dampak kegagalan perubahan menjadi berkurang.

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

Panduan implementasi

Gunakan perubahan yang sering, kecil, dan dapat dikembalikan untuk mengurangi cakupan dan dampak yang ditimbulkan perubahan tersebut. Hal ini akan memudahkan Anda dalam melakukan pemecahan masalah, membantu proses remediasi yang lebih cepat, dan menyediakan opsi untuk membatalkan perubahan. Hal ini juga akan meningkatkan rasio nilai yang dapat Anda berikan ke bisnis.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP03 Menggunakan sistem manajemen konfigurasi
- OPS05-BP04 Menggunakan sistem manajemen build dan deployment
- OPS06-BP04 Mengotomatiskan pengujian dan rollback

Dokumen terkait:

- Menerapkan Layanan Mikro pada AWS
- Layanan Mikro - Observabilitas

OPS05-BP10 Mengotomatiskan integrasi dan deployment sepenuhnya

Otomatiskan build, deployment, dan pengujian beban kerja. Hal ini mengurangi kesalahan yang disebabkan oleh proses manual, dan mengurangi upaya untuk melakukan deployment perubahan.

Terapkan metadata menggunakan Tag Sumber Daya dan AWS Resource Groups sesuai strategi penandaan yang konsisten untuk mencapai identifikasi sumber daya Anda. Berikan tag pada sumber daya Anda untuk organisasi, akuntansi biaya, kontrol akses, dan penargetan pelaksanaan aktivitas operasi yang diotomatiskan.

Hasil yang diinginkan: Pengembang menggunakan alat untuk mengirimkan kode dan mencapai produksi. Pengembang tidak harus masuk log in ke dalam Konsol Manajemen AWS untuk memberikan pembaruan. Terdapat jejak audit penuh untuk perubahan dan konfigurasi, sehingga hal itu cukup untuk memenuhi kebutuhan tata kelola dan kepatuhan. Proses dapat diulang dan distandardisasi di seluruh tim. Pengembang bebas untuk memusatkan perhatian pada pengembangan dan pendorongan kode, sehingga akan meningkatkan produktivitas.

Anti-pola umum:

- Pada hari Jumat, Anda selesai menulis kode baru untuk cabang fitur Anda. Pada hari Senin, setelah menjalankan skrip pengujian kualitas kode dan setiap skrip pengujian unit, Anda mendaftarkan kode tersebut untuk rilisan terjadwal berikutnya.
- Anda ditugaskan untuk membuat kode perbaikan untuk sebuah masalah besar yang dapat memengaruhi banyak pelanggan di lingkungan produksi. Setelah menguji perbaikan tersebut, Anda melakukan commit terhadap kode Anda dan mengirimkan manajemen perubahan melalui email untuk meminta persetujuan deployment ke lingkungan produksi.
- Sebagai pengembang, Anda masuk log in ke Konsol Manajemen AWS untuk membuat sebuah lingkungan pengembangan baru dengan menggunakan metode dan sistem yang tidak standar.

Manfaat menerapkan praktik terbaik ini: Dengan mengimplementasikan sistem manajemen build dan deployment otomatis, Anda dapat mengurangi kesalahan yang disebabkan proses yang diselesaikan secara manual dan mengurangi upaya yang diperlukan untuk melakukan deployment perubahan, sehingga akan membantu anggota tim Anda berkonsentrasi untuk menghadirkan nilai bisnis. Anda meningkatkan kecepatan pengiriman selama proses menuju lingkungan produksi.

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

Panduan implementasi

Anda menggunakan sistem manajemen build dan deployment untuk melacak dan mengimplementasikan perubahan, mengurangi kesalahan yang disebabkan oleh proses yang diselesaikan secara manual, dan mengurangi upaya yang diperlukan. Lakukan otomatisasi sepenuhnya terhadap pipeline integrasi dan deployment dari check-in kode hingga build, pengujian, deployment, dan validasi. Hal ini dapat mengurangi waktu tunggu, mendorong peningkatan frekuensi perubahan, mengurangi tingkat upaya, meningkatkan kecepatan masuk pasar, menghasilkan peningkatan produktivitas, dan meningkatkan keamanan kode Anda selama proses Anda menuju lingkungan produksi.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP03 Menggunakan sistem manajemen konfigurasi
- OPS05-BP04 Menggunakan sistem manajemen build dan deployment

Dokumen terkait:

- Apa itu AWS CodeBuild?
- Apa itu AWS CodeDeploy?

Video terkait:

- AWS re:Invent 2022 - Praktik terbaik AWS Well-Architected untuk DevOps di AWS

Memitigasi risiko deployment

Adopsi pendekatan yang memberikan umpan balik cepat atas kualitas dan menyediakan pemulihan cepat dari perubahan yang tidak memiliki hasil yang tidak diinginkan. Menggunakan praktik tersebut akan memitigasi dampak masalah akibat deployment perubahan.

Desain beban kerja Anda harus menjelaskan bagaimana beban kerja akan diterapkan, diperbarui, dan dioperasikan. Anda akan mengimplementasikan praktik-praktik rekayasa yang selaras dengan pengurangan cacat serta perbaikan yang cepat dan aman.

Praktik terbaik

- OPS06-BP01 Rencana untuk perubahan yang gagal
- OPS06-BP02 Uji penerapan
- OPS06-BP03 Menggunakan strategi deployment yang aman
- OPS06-BP04 Mengotomatiskan pengujian dan rollback

OPS06-BP01 Rencana untuk perubahan yang gagal

Rencanakan untuk kembali ke keadaan yang diketahui pasti baik, atau perbaiki di lingkungan produksi jika deployment menyebabkan hasil yang tidak diinginkan. Adanya kebijakan untuk menetapkan rencana semacam ini bermanfaat bagi semua tim dalam mengembangkan strategi untuk pulih dari perubahan yang gagal. Beberapa contoh strategi adalah langkah deployment dan rollback, kebijakan perubahan, penanda fitur, pemisahan lalu lintas, dan pergeseran lalu lintas. Rilis tunggal dapat mencakup beberapa perubahan komponen yang terkait. Strategi harus memberikan kemampuan untuk bertahan atau pulih dari kegagalan perubahan komponen apa pun.

Hasil yang diinginkan: Anda telah menyiapkan sebuah rencana pemulihan yang mendetail untuk perubahan Anda apabila perubahan tersebut tidak berhasil. Selain itu, Anda juga telah mengurangi

ukuran rilis untuk meminimalkan dampak-dampak potensial yang mungkin ditimbulkan terhadap komponen beban kerja lainnya. Hasilnya, Anda telah mengurangi dampak bisnis Anda dengan mempersingkat potensi waktu henti yang mungkin diakibatkan oleh kegagalan perubahan dan meningkatkan fleksibilitas serta efisiensi waktu pemulihan.

Anti-pola umum:

- Anda melakukan deployment dan aplikasi Anda menjadi tidak stabil, namun sepertinya masih ada pengguna yang aktif di sistem. Anda harus memutuskan apakah akan melakukan roll back terhadap perubahan yang akan berdampak pada pengguna aktif atau menunggu untuk melakukan roll back perubahan tersebut karena tahu bagaimana pun juga pengguna dapat terkena dampaknya.
- Setelah Anda membuat perubahan rutin, lingkungan baru Anda dapat diakses tetapi salah satu subnet Anda menjadi tidak dapat dijangkau. Anda harus memutuskan apakah akan melakukan roll back terhadap semuanya atau mencoba memperbaiki subnet yang tidak dapat diakses tersebut. Sementara Anda sedang memutuskan hal ini, subnet tersebut tetap tidak dapat dijangkau.
- Sistem Anda tidak dirancang dapat diperbarui dengan rilis-rilis yang lebih kecil. Akibatnya, Anda mengalami kesulitan dalam membatalkan perubahan massal tersebut selama deployment yang gagal.
- Anda tidak menggunakan infrastruktur sebagai kode (IaC) dan Anda melakukan pembaruan secara manual pada infrastruktur Anda sehingga mengakibatkan terjadinya konfigurasi yang tidak diinginkan. Anda tidak dapat melacak dan membatalkan perubahan manual secara efektif.
- Karena Anda belum mengukur peningkatan frekuensi deployment Anda, tim Anda kemudian mengalami kesulitan untuk mengurangi ukuran perubahan mereka dan meningkatkan rencana rollback mereka untuk setiap perubahan, yang berimbas pada risiko yang lebih besar dan tingkat kegagalan yang meningkat.
- Anda tidak mengukur total durasi pemadaman (outage) yang disebabkan oleh perubahan yang tidak berhasil. Tim Anda tidak dapat memprioritaskan dan meningkatkan proses deployment serta efektivitas rencana pemulihannya.

Manfaat membangun praktik terbaik ini: Memiliki rencana untuk pulih dari perubahan yang gagal meminimalkan waktu rata-rata untuk memulihkan (MTTR) dan mengurangi dampak bisnis Anda.

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

Panduan implementasi

Kebijakan dan praktik yang konsisten serta terdokumentasi yang diadopsi oleh tim rilis akan memungkinkan organisasi untuk merencanakan apa yang seharusnya terjadi apabila terjadi kegagalan perubahan. Kebijakan tersebut harus memungkinkan perbaikan ke depan (fixing forward) dalam keadaan tertentu. Dalam situasi apa pun, rencana perbaikan ke depan atau rollback harus didokumentasikan dan diuji dengan baik sebelum melakukan deployment ke lingkungan produksi langsung sehingga waktu yang diperlukan untuk mengembalikan perubahan dapat diminimalkan.

Langkah-langkah implementasi

1. Buatlah dokumentasi kebijakan yang mengharuskan tim memiliki rencana efektif untuk mengembalikan perubahan dalam periode tertentu.
   a. Kebijakan harus menentukan kapan situasi perbaikan ke depan diperbolehkan.
   b. Rencana rollback yang terdokumentasi harus dapat diakses oleh semua pihak yang terlibat.
   c. Tentukan persyaratan-persyaratan untuk rollback (misalnya, ketika ternyata ada deployment perubahan tidak sah).
2. Lakukan analisis terhadap tingkat dampak yang ditimbulkan oleh semua perubahan yang berkaitan dengan setiap komponen dari sebuah beban kerja.
   a. Buatlah perubahan-perubahan berulang memungkinkan untuk distandardisasi, dijadikan templat, dan diotorisasi di awal jika perubahan-perubahan tersebut mengikuti alur kerja yang konsisten yang memberlakukan kebijakan perubahan.
   b. Kurangi potensi dampak yang mungkin ditimbulkan oleh setiap perubahan dengan menjadikan ukuran perubahan lebih kecil sehingga waktu pemulihan yang dibutuhkan menjadi lebih singkat dan menyebabkan lebih sedikit dampak bisnis.
   c. Pastikan prosedur rollback akan mengembalikan kode ke keadaan yang pasti baik untuk menghindari terjadinya insiden, jika memungkinkan.
3. Integrasikan alat-alat dan alur kerja untuk menegakkan kebijakan Anda secara terprogram.
4. Buat agar data tentang perubahan dapat dilihat oleh para pemilik beban kerja lain untuk meningkatkan kecepatan diagnosis perubahan yang gagal yang tidak dapat dibatalkan.
   a. Ukur keberhasilan praktik ini dengan menggunakan data perubahan yang terlihat dan identifikasi setiap peningkatan iteratif yang mungkin dilakukan.
5. Gunakan alat-alat pemantauan untuk memverifikasi keberhasilan atau kegagalan sebuah deployment untuk mempercepat pengambilan keputusan saat melakukan rollback.

6. Ukur durasi pemadaman (outage) Anda selama terjadi kegagalan perubahan untuk terus meningkatkan kualitas rencana pemulihan Anda.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS06-BP04 Mengotomatiskan pengujian dan rollback

Dokumen terkait:

- AWS Builders Library | Memastikan Keamanan Rollback Selama Penerapan
- AWS Whitepaper | Ubah Manajemen di Cloud

Video terkait:

- re:Invent 2019 | Pendekatan Amazon untuk deployment ketersediaan tinggi

OPS06-BP02 Uji penerapan

Uji prosedur rilis dalam tahap praproduksi dengan menggunakan konfigurasi deployment, kontrol keamanan, langkah, dan prosedur yang sama seperti dalam tahap produksi. Lakukan validasi bahwa semua langkah yang di-deploy selesai sesuai harapan, seperti dengan memeriksa file, konfigurasi, dan layanan. Uji lebih lanjut semua perubahan dengan pengujian fungsional, integrasi, dan beban, beserta pemantauan apa pun seperti pemeriksaan kondisi. Dengan melakukan pengujian ini, Anda dapat mengidentifikasi masalah deployment lebih awal dengan peluang untuk merencanakan dan menanggulanginya sebelum produksi.

Anda dapat membuat lingkungan paralel sementara untuk menguji setiap perubahan. Lakukan otomatisasi deployment lingkungan pengujian dengan menggunakan infrastruktur sebagai kode (IaC) untuk membantu mengurangi jumlah pekerjaan yang terlibat dan memastikan stabilitas, konsistensi, dan pengiriman fitur yang lebih cepat.

Hasil yang diinginkan: Organisasi Anda mengadopsi budaya pengembangan berbasis pengujian yang mencakup pengujian deployment. Ini akan memastikan bahwa tim akan berkonsentrasi untuk

menghadirkan nilai bisnis, bukan mengelola rilis. Tim terlibat sejak dini setelah identifikasi risiko deployment untuk menentukan arah mitigasi yang sesuai.

Anti-pola umum:

- Selama rilis produksi, deployment yang belum teruji sering kali akan menyebabkan masalah-masalah yang memerlukan penyelesaian dan eskalasi.
- Rilis Anda berisi infrastruktur sebagai kode (IaC) yang memperbarui sumber daya yang ada sekarang. Anda tidak yakin apakah IaC berjalan dengan sukses atau akan menyebabkan dampak pada sumber daya.
- Anda men-deploy sebuah fitur baru ke aplikasi Anda. Fitur tersebut tidak berfungsi sesuai keinginan dan masalah ini baru dapat diketahui setelah dilaporkan oleh para pengguna yang terdampak.
- Anda memperbarui sertifikat Anda. Anda secara tidak sengaja menginstal sertifikat ke komponen-komponen yang salah, yang akhirnya tidak terdeteksi dan berdampak pada pengunjung situs web karena koneksi yang aman ke situs web tidak dapat dibuat.

Manfaat menerapkan praktik terbaik ini: Pengujian ekstensif selama tahap pra-produksi dalam prosedur deployment serta perubahan-perubahan yang dimunculkannya dapat meminimalkan potensi dampak yang mungkin dapat ditimbulkan terhadap lingkungan produksi yang disebabkan oleh langkah-langkah deployment. Hal ini akan meningkatkan kepercayaan diri selama rilis produksi dan meminimalkan dukungan operasional tanpa memperlambat kecepatan penyampaian perubahan yang hendak dilakukan.

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

Panduan implementasi

Menguji proses deployment Anda sama pentingnya dengan menguji perubahan yang dihasilkan dari deployment Anda. Hal ini dapat dicapai dengan menguji langkah-langkah deployment Anda di lingkungan pra-produksi yang semaksimal mungkin mencerminkan produksi. Masalah-masalah umum, seperti langkah-langkah deployment yang tidak lengkap atau salah, atau kesalahan konfigurasi, dapat terdeteksi sebelum masuk ke lingkungan produksi. Selain itu, Anda dapat menguji langkah-langkah pemulihan Anda.

Contoh pelanggan

Sebagai bagian dari pipeline continuous integration and continuous delivery (CI/CD), AnyCompany Retail melakukan langkah-langkah yang ditentukan yang diperlukan untuk merilis pembaruan

infrastruktur dan perangkat lunak bagi pelanggannya dalam lingkungan seperti produksi. Pipeline tersebut terdiri dari langkah pra-pemeriksaan untuk mendeteksi penyimpangan (mendeteksi perubahan pada sumber daya yang dilakukan di luar IaC Anda) di dalam sumber daya sebelum deployment, serta melakukan validasi terhadap tindakan-tindakan yang dilakukan IaC setelah inisiasi. Tahap ini memvalidasi langkah-langkah deployment, seperti memverifikasi bahwa file dan konfigurasi tertentu sudah siap dan layanan-layanan sudah berada dalam status berjalan serta merespons dengan benar pemeriksaan kondisi pada host lokal sebelum didaftarkan ulang dengan penyeimbang beban. Selain itu, semua perubahan menandai sejumlah pengujian otomatis, misalnya pengujian fungsional, keamanan, regresi, integrasi, dan beban.

Langkah-langkah implementasi

1. Lakukan pemeriksaan pra-instalasi untuk mencerminkan lingkungan pra-produksi ke lingkungan produksi.
   a. Gunakan deteksi drift untuk mendeteksi kapan sumber daya telah diubah di luar. CloudFormation
   b. Gunakan set perubahan untuk memvalidasi bahwa maksud pembaruan tumpukan cocok dengan tindakan yang CloudFormation dilakukan saat set perubahan dimulai.
2. Ini akan memicu langkah persetujuan manual di AWS CodePipeline untuk mengotorisasi deployment ke lingkungan pra-produksi.
3. Gunakan konfigurasi penerapan seperti AWS CodeDeploy AppSpecfile untuk menentukan langkah penerapan dan validasi.
4. Jika berlaku, berintegrasi AWS CodeDeploy dengan AWS layanan lain atau berintegrasi AWS CodeDeploy dengan produk dan layanan mitra.
5. Pantau penerapan menggunakan Amazon CloudWatch, AWS CloudTrail, dan pemberitahuan SNS acara Amazon.
6. Lakukan pengujian otomatis pasca-deployment, termasuk pengujian fungsional, keamanan, regresi, integrasi, dan beban.
7. Memecahkan masalah deployment.
8. Validasi yang berhasil terhadap langkah-langkah sebelumnya seharusnya menginisiasi alur kerja persetujuan manual untuk memberikan otorisasi deployment ke produksi.

Tingkat upaya untuk rencana implementasi: Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP02 Menguji dan memvalidasi perubahan

Dokumen terkait:

- AWS Perpustakaan Pembangun | Mengotomatiskan penerapan yang aman dan lepas tangan | Uji Penerapan
- AWS Whitepaper | Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan AWS
- Kisah Apollo - Mesin Deployment Amazon
- Cara menguji dan men-debug AWS CodeDeploy secara lokal sebelum Anda mengirimkan kode Anda
- Mengintegrasikan Pengujian Konektivitas Jaringan dengan Deployment Infrastruktur

Video terkait:

- re:Invent 2020 | Menguji perangkat lunak dan sistem di Amazon

Contoh terkait:

- Tutorial | Menyebarkan dan ECS layanan Amazon dengan tes validasi

OPS06-BP03 Menggunakan strategi deployment yang aman

Peluncuran produksi yang aman mengontrol aliran perubahan yang bermanfaat dengan tujuan untuk meminimalkan dampak yang dirasakan oleh pelanggan dari perubahan tersebut. Kontrol keselamatan menyediakan mekanisme-mekanisme inspeksi untuk memvalidasi hasil yang diinginkan dan membatasi ruang lingkup dampak yang ditimbulkan oleh cacat apa pun yang disebabkan oleh perubahan atau kegagalan deployment. Peluncuran yang aman dapat mencakup strategi seperti feature-flag, one-box, rolling (rilis canary), immutable, pemisahan lalu lintas, dan deployment blue/green.

Hasil yang diinginkan: Organisasi Anda menggunakan sebuah sistem integrasi berkelanjutan pengiriman berkelanjutan (CI/CD) yang menyediakan kemampuan-kemampuan untuk

mengotomatiskan peluncuran (rollout) dengan aman. Tim diharuskan menggunakan strategi peluncuran aman yang sesuai.

Anti-pola umum:

- Anda melakukan deployment perubahan yang tidak berhasil ke seluruh lingkungan produksi secara sekaligus. Akibatnya, semua pelanggan merasakan dampaknya secara bersamaan.
- Cacat akibat deployment serentak yang dilakukan ke semua sistem memerlukan rilis darurat. Diperlukan waktu beberapa hari untuk memperbaikinya untuk semua pelanggan.
- Untuk mengelola rilis produksi diperlukan perencanaan dan partisipasi dari beberapa tim. Hal ini akan menghambat kemampuan Anda untuk melakukan pembaruan fitur bagi pelanggan Anda dalam rentang waktu yang berdekatan (frequent).
- Anda melakukan deployment yang dapat diubah dengan melakukan modifikasi terhadap sistem yang sudah ada. Setelah mengetahui bahwa perubahan yang di-deploy tidak berhasil, Anda terpaksa melakukan modifikasi terhadap sistem sekali lagi untuk memulihkan versi yang sebelumnya, dan hal ini memperpanjang waktu pemulihan Anda.

Manfaat menerapkan praktik terbaik ini: Deployment otomatis menyeimbangkan kecepatan peluncuran (roll-out) dengan menghadirkan perubahan yang bermanfaat secara konsisten kepada para pelanggan. Pembatasan dampak dapat mencegah kegagalan deployment yang mahal dan memaksimalkan kemampuan tim untuk merespons kegagalan tersebut dengan efisien.

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

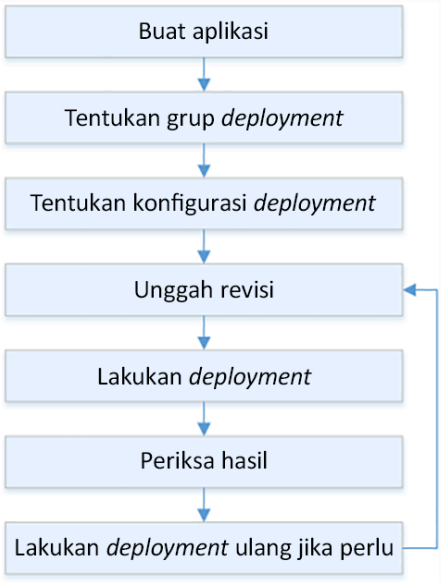
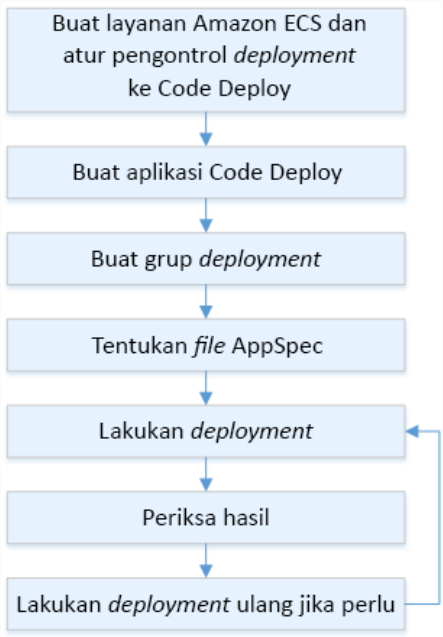
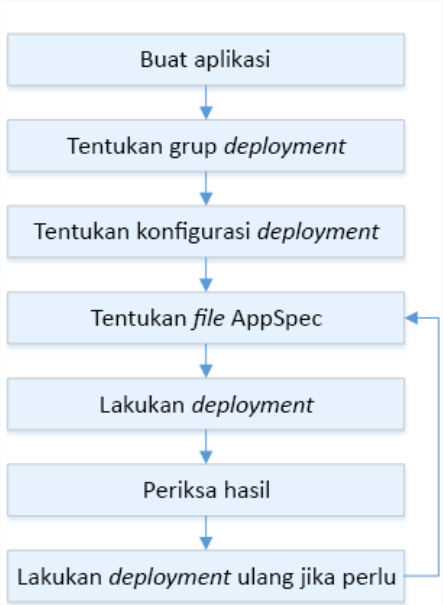
Panduan implementasi

Kegagalan pengiriman yang terjadi secara berkelanjutan dapat menyebabkan berkurangnya ketersediaan layanan dan buruknya pengalaman pelanggan. Untuk memaksimalkan tingkat keberhasilan deployment, terapkan kontrol keamanan dalam proses rilis menyeluruh (end-to-end) untuk meminimalkan kesalahan-kesalahan deployment, dengan tujuan untuk mencapai nol kegagalan deployment.

Contoh pelanggan

AnyCompany Retail memiliki misi untuk mencapai deployment dengan waktu henti yang minim hingga nol, yang berarti pengguna sama sekali tidak ada merasakan dampak selama deployment. Untuk mencapai tujuan tersebut, perusahaan telah membuat pola-pola deployment (lihat diagram

alur kerja berikut), seperti deployment blue/green dan deployment bergulir (rolling). Semua tim mengadopsi satu atau beberapa pola tersebut di dalam pipeline CI/CD mereka.

Alur kerja CodeDeploy untuk Amazon EC2	Alur kerja CodeDeploy untuk Amazon ECS	Alur kerja CodeDeploy untuk Lambda
		

Langkah-langkah implementasi

1. Gunakan alur kerja persetujuan untuk memulai urutan langkah-langkah peluncuran (roll-out) produksi setelah promosi ke produksi.
2. Gunakan sistem deployment otomatis seperti AWS CodeDeploy. AWS CodeDeploy Opsi deployment mencakup deployment pengganti untuk EC2/On-Premise dan deployment blue/green untuk EC2/On-Premise, AWS Lambdadan Amazon ECS (lihat diagram alur kerja sebelumnya).
   a. Jika berlaku, integrasikan AWS CodeDeploy dengan layanan-layanan AWS lain atau integrasikan AWS CodeDeploy dengan produk dan layanan mitra.
3. Gunakan deployment blue/gree untuk basis data seperti Amazon Aurora dan Amazon RDS.
4. Pantau deployment dengan menggunakan Amazon CloudWatch, AWS CloudTrail dan pemberitahuan acara Amazon Simple Notification Service (Amazon SNS).
5. Lakukan pengujian otomatis pasca-deployment, antara lain pengujian fungsional, keamanan, regresi, integrasi, dan uji beban.

6. Memecahkan masalah deployment.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP02 Menguji dan memvalidasi perubahan
- OPS05-BP09 Lakukan perubahan yang sering, kecil, dan reversibel
- OPS05-BP10 Mengotomatiskan integrasi dan deployment sepenuhnya

Dokumen terkait:

- Pustaka Builder AWS | Melakukan otomatisasi deployment secara aman dan otonom | Deployment produksi
- AWS Pustaka Builder | Pipeline CI/CD saya adalah pemandu utama rilis saya | Rilis produksi otomatis yang aman
- Laporan Resmi AWS | Mempraktikkan Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS | Metode deployment
- AWS CodeDeploy Panduan Pengguna
- Bekerja dengan konfigurasi deployment di AWS CodeDeploy
- Siapkan deployment rilis canary API Gateway
- Tipe Deployment Amazon ECS
- Deployment Blue/Green yang Dikelola Sepenuhnya di Amazon Aurora dan Amazon RDS
- Deployment Blue/Green dengan AWS Elastic Beanstalk

Video terkait:

- re:Invent 2020 | Hands-off: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon
- re:Invent 2019 | Pendekatan deployment ketersediaan tinggi Amazon

Contoh terkait:

- Coba Sampel Deployment Blue/Green di AWS CodeDeploy

- Lokakarya | Membangun pipeline CI/CD untuk deployment canary Lambda menggunakan AWS CDK
- Lokakarya | Membangun pipeline Biru/Hijau DevOps pertama Anda dengan Amazon ECS
- Lokakarya | Membangun pipeline Biru/Hijau DevOps pertama Anda dengan Amazon EKS
- Lokakarya | EKS GitOps dengan ArgoCD
- Lokakarya | CI/CD di Lokakarya AWS
- Mengimplementasikan CI/CD lintas akun dengan AWS SAM untuk fungsi Lambda berbasis kontainer

OPS06-BP04 Mengotomatiskan pengujian dan rollback

Untuk meningkatkan kecepatan, keandalan, dan keyakinan pada proses deployment Anda, miliki strategi untuk kemampuan pengujian dan rollback otomatis di lingkungan praproduksi dan produksi. Otomatiskan pengujian saat melakukan deployment ke produksi untuk menyimulasikan interaksi manusia dan sistem yang memverifikasi perubahan yang sedang di-deploy. Otomatiskan rollback untuk kembali ke keadaan pasti baik sebelumnya dengan cepat. Rollback harus dimulai secara otomatis pada kondisi yang telah ditentukan di awal seperti ketika hasil perubahan yang Anda inginkan tidak tercapai atau ketika pengujian otomatis mengalami kegagalan. Mengotomatiskan kedua aktivitas ini dapat memperbaiki tingkat keberhasilan untuk deployment Anda, meminimalkan waktu pemulihan, dan mengurangi potensi dampak terhadap bisnis.

Hasil yang diinginkan: Strategi-strategi pengujian dan rollback otomatis Anda diintegrasikan ke dalam pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) Anda. Pemantauan Anda dapat melakukan validasi berdasarkan kriteria keberhasilan Anda dan memulai rollback otomatis setelah terjadi kegagalan. Hal ini akan meminimalkan dampak apa pun yang terjadi terhadap pelanggan dan pengguna akhir. Misalnya, ketika semua hasil pengujian telah terpenuhi, Anda meneruskan kode Anda ke lingkungan produksi tempat pengujian regresi otomatis dimulai, dengan memanfaatkan kasus-kasus pengujian yang sama. Jika hasil pengujian regresi yang didapatkan tidak sesuai dengan harapan, maka rollback otomatis akan dimulai dalam alur kerja pipeline.

Anti-pola umum:

- Sistem Anda tidak dirancang dapat diperbarui dengan rilis-rilis yang lebih kecil. Akibatnya, Anda mengalami kesulitan dalam membatalkan perubahan massal tersebut selama deployment yang gagal.

- Proses deployment Anda terdiri dari serangkaian langkah-langkah manual. Setelah melakukan deployment perubahan ke beban kerja, Anda mulai melakukan pengujian pasca-deployment. Setelah pengujian selesai, Anda menyadari bahwa beban kerja Anda tidak dapat dioperasikan dan koneksi pelanggan terputus. Kemudian Anda mulai melakukan rollback ke versi sebelumnya. Semua langkah manual ini dapat menghambat pemulihan sistem secara keseluruhan dan akan menyebabkan dampak yang berkepanjangan terhadap pelanggan Anda.
- Anda menghabiskan waktu mengembangkan kasus-kasus pengujian otomatis untuk fungsionalitas yang jarang digunakan dalam aplikasi Anda, sehingga memperkecil laba atas investasi (roi) dalam kemampuan pengujian otomatis Anda.
- Rilis Anda terdiri dari aplikasi, infrastruktur, patch, dan pembaruan konfigurasi yang tidak bergantung satu sama lain. Namun demikian, Anda memiliki satu pipeline CI/CD yang mengirimkan semua perubahan dalam satu waktu sekaligus. Kegagalan yang terjadi pada satu komponen memaksa Anda untuk mengembalikan semua perubahan, dan membuat rollback Anda menjadi kompleks dan tidak efisien.
- Tim Anda menyelesaikan tugas-tugas coding dalam sprint one dan memulai tugas sprint two, tetapi rencana Anda tidak menyertakan pengujian sampai sprint three. Akibatnya, pengujian-pengujian otomatis mengungkap cacat dari sprint one yang harus diselesaikan sebelum pengujian hasil kerja sprint two dapat dimulai dan seluruh rilis menjadi tertunda, sehingga menurunkan nilai pengujian otomatis Anda.
- Kasus pengujian regresi otomatis Anda untuk rilis produksi sudah selesai, tetapi Anda tidak memantau kondisi beban kerja. Karena Anda tidak memiliki visibilitas mengenai apakah layanan telah dimulai ulang atau belum, Anda menjadi tidak yakin apakah rollback diperlukan atau rollback sudah terjadi.

Manfaat menerapkan praktik terbaik ini: Pengujian otomatis meningkatkan transparansi proses pengujian Anda dan kemampuan Anda untuk mencakup lebih banyak fitur dalam satu periode waktu yang lebih singkat. Dengan melakukan pengujian dan validasi terhadap perubahan-perubahan dalam produksi, Anda dapat mengidentifikasi masalah dengan cepat. Peningkatan konsistensi dengan alat-alat pengujian otomatis memungkinkan deteksi kecacatan yang lebih baik. Dengan melakukan rollback otomatis ke versi sebelumnya, dampak-dampak yang ditimbulkan terhadap para pelanggan diminimalkan. Rollback otomatis pada akhirnya akan memunculkan keyakinan yang lebih tinggi pada kemampuan deployment Anda dengan mengurangi dampak bisnis. Secara keseluruhan, kemampuan ini berkurang time-to-delivery sambil memastikan kualitas.

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

Panduan implementasi

Lakukan otomatisasi terhadap pengujian lingkungan yang di-deploy untuk mengonfirmasi hasil-hasil yang diinginkan dengan lebih cepat. Lakukan otomatisasi terhadap rollback ke keadaan yang diketahui baik sebelumnya ketika hasil yang ditetapkan di awal tidak tercapai, untuk mempersingkat waktu pemulihan dan mengurangi kesalahan yang disebabkan oleh proses-proses yang dilakukan secara manual. Integrasikan alat-alat pengujian dengan alur kerja pipeline Anda untuk menguji dan meminimalkan input manual secara konsisten. Prioritaskan otomatisasi kasus pengujian, seperti kasus pengujian yang memitigasi risiko terbesar dan kasus pengujian yang harus sering diuji dengan setiap perubahan. Selain itu, otomatiskan rollback berdasarkan kondisi tertentu yang telah ditentukan sebelumnya di awal dalam rencana pengujian Anda.

Langkah-langkah implementasi

1. Bangun sebuah siklus hidup pengujian untuk siklus hidup pengembangan Anda yang menentukan setiap tahap proses pengujian mulai dari perencanaan persyaratan hingga pengembangan kasus pengujian, konfigurasi alat, pengujian otomatis, dan penutupan kasus pengujian.
   a. Buatlah sebuah pendekatan pengujian khusus beban kerja dari strategi pengujian Anda secara keseluruhan.
   b. Pertimbangkan strategi pengujian berkelanjutan jika diperlukan di seluruh siklus hidup pengembangan.
2. Pilih alat-alat otomatis untuk pengujian dan rollback berdasarkan kebutuhan bisnis dan investasi pipeline Anda.
3. Tentukan kasus pengujian mana yang ingin Anda otomatisasi dan mana yang harus dilakukan secara manual. Anda dapat menentukannya berdasarkan prioritas nilai bisnis dari fitur yang sedang diuji. Selaraskan semua anggota tim dengan rencana ini dan pastikan pertanggungjawabannya untuk melakukan pengujian manual.
   a. Terapkan kemampuan-kemampuan pengujian otomatis ke kasus pengujian tertentu yang cocok untuk otomatisasi, seperti kasus berulang atau yang sering dijalankan, kasus yang memerlukan tugas berulang, atau kasus yang diperlukan di beberapa konfigurasi.
   b. Tentukan skrip otomatisasi pengujian serta kriteria keberhasilan di dalam alat otomatisasi sehingga otomatisasi alur kerja yang berkelanjutan dapat dimulai ketika ada kasus tertentu yang mengalami kegagalan.
   c. Tentukan kriteria kegagalan khusus untuk melakukan rollback otomatis.

4. Prioritaskan otomatisasi pengujian untuk mendorong hasil yang konsisten dengan pengembangan kasus pengujian menyeluruh di mana kompleksitas dan interaksi manusia memiliki risiko kegagalan yang lebih tinggi.
5. Integrasikan alat-alat pengujian otomatis dan rollback Anda ke dalam pipeline CI/CD Anda.
    a. Kembangkan kriteria keberhasilan yang jelas untuk perubahan Anda.
    b. Lakukan pemantauan dan pengamatan untuk mendeteksi kriteria-kriteria ini dan secara otomatis membatalkan perubahan ketika kriteria rollback tertentu terpenuhi.
6. Lakukan berbagai jenis pengujian produksi otomatis, seperti:
    a. Pengujian A/B untuk menunjukkan hasil yang dibandingkan dengan versi saat ini antara dua kelompok pengujian pengguna.
    b. Pengujian canary yang memungkinkan Anda untuk meluncurkan perubahan Anda pada subset pengguna sebelum merilisnya ke semua pengguna.
    c. Pengujian penandaan fitur (feature-flag testing) yang memungkinkan satu per satu fitur dari versi baru untuk ditandai atau dihapus tandanya dari luar aplikasi sehingga setiap fitur baru dapat divalidasi satu per satu.
    d. Pengujian regresi untuk memverifikasi fungsionalitas baru dengan komponen-komponen yang saling terkait.
7. Lakukan pemantauan pada aspek operasional aplikasi, transaksi, dan interaksi dengan aplikasi dan komponen lain. Kembangkan laporan untuk menunjukkan keberhasilan perubahan berdasarkan beban kerja sehingga Anda dapat mengidentifikasi bagian otomatisasi dan alur kerja apa yang dapat dioptimalkan lebih lanjut.
    a. Kembangkan laporan hasil pengujian yang membantu Anda mengambil keputusan cepat terkait apakah prosedur rollback perlu diinvokasi.
    b. Terapkan sebuah strategi yang dapat memungkinkan rollback otomatis berdasarkan kondisi kegagalan yang telah ditentukan di awal yang dihasilkan dari satu atau beberapa metode pengujian Anda.
8. Kembangkan kasus pengujian otomatis untuk memungkinkan penggunaan ulang di seluruh perubahan berulang di masa mendatang.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS06-BP01 Rencana untuk perubahan yang gagal
- OPS06-BP02 Uji penerapan

Dokumen terkait:

- AWS Builders Library | Memastikan keamanan rollback selama penerapan
- Menerapkan ulang dan mengembalikan penerapan dengan AWS CodeDeploy
- 8 praktik terbaik saat mengotomatiskan penerapan Anda dengan AWS CloudFormation

Contoh terkait:

- Pengujian UI tanpa server menggunakan Selenium,, AWS Lambda, dan Alat AWS Fargate Pengembang AWS

Video terkait:

- re:Invent 2020 | Hands-off: Mengotomatiskan pipeline pengiriman berkelanjutan di Amazon
- re:Invent 2019 | Pendekatan deployment ketersediaan tinggi Amazon

Kesiapan operasional dan manajemen perubahan

Evaluasi kesiapan operasional beban kerja, proses, prosedur, dan personel Anda untuk memahami risiko operasional terkait beban kerja Anda. Kelola alur perubahan ke dalam lingkungan Anda.

Anda harus menggunakan proses yang konsisten (termasuk daftar periksa manual dan otomatis) untuk mengetahui saat Anda siap untuk mengoperasionalkan beban kerja Anda atau untuk melakukan perubahan. Hal ini juga akan membantu Anda menemukan area mana pun yang Anda butuhkan untuk membuat rencana untuk ditangani. Anda akan memiliki runbook yang mendokumentasikan aktivitas rutin serta pedoman yang memandu proses penyelesaian masalah Anda. Gunakan mekanisme untuk mengelola perubahan yang mendukung penghadiran nilai bisnis dan bantu mitigasi risiko terkait perubahan.

Praktik terbaik

- OPS07-BP01 Memastikan kemampuan personel
- OPS07-BP02 Memastikan peninjauan yang konsisten terkait kesiapan operasional

- OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur
- OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah
- OPS07-BP05 Membuat keputusan yang tepat untuk melakukan deployment sistem dan perubahan
- OPS07-BP06 Membuat rencana dukungan untuk beban kerja produksi

OPS07-BP01 Memastikan kemampuan personel

Miliki mekanisme untuk memvalidasi bahwa Anda memiliki jumlah personel terlatih yang sesuai untuk mendukung beban kerja. Mereka harus diberi pelatihan tentang platform dan layanan yang membentuk beban kerja Anda. Berikan kepada mereka pengetahuan yang diperlukan untuk mengoperasikan beban kerja. Anda harus memiliki cukup banyak personel terlatih untuk mendukung pengoperasian normal beban kerja dan menyelesaikan masalah-masalah terkait insiden yang terjadi. Anda harus memiliki cukup banyak personel sehingga Anda dapat melakukan rotasi untuk personel yang siap tugas mendadak dan personel yang liburan guna menghindari lelah ekstrim pada personel.

Hasil yang diinginkan:

- Ada cukup banyak personel terlatih untuk mendukung beban kerja pada saat beban kerja tersedia.
- Anda memberikan pelatihan tentang perangkat lunak dan layanan yang membentuk beban kerja Anda kepada personel.

Anti-pola umum:

- Melakukan deployment beban kerja tanpa anggota tim yang terlatih untuk mengoperasikan platform dan layanan yang digunakan.
- Tidak memiliki cukup banyak personel untuk mendukung pelaksanaan rotasi personel yang siap tugas mendadak atau personel yang sedang libur.

Manfaat menjalankan praktik terbaik ini:

- Memiliki anggota tim yang terampil membantu dukungan yang efektif untuk beban kerja.
- Dengan cukup banyak anggota tim, Anda dapat mendukung beban kerja dan pelaksanaan rotasi personel yang siap tugas mendadak sekaligus mengurangi risiko personel yang terlalu lelah.

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

Panduan implementasi

Pastikan bahwa terdapat personel yang terlatih dengan memadai untuk mendukung beban kerja. Pastikan Anda memiliki jumlah anggota tim yang cukup untuk menangani aktivitas-aktivitas operasional dalam kondisi normal, termasuk pelaksanaan rotasi personel yang siap bertugas mendadak.

Contoh pelanggan

AnyCompany Retail memastikan tim yang mendukung beban kerja memiliki staf yang terlatih dalam jumlah yang sesuai. Mereka memiliki cukup banyak rekayasawan untuk mendukung pelaksanaan rotasi personel yang siap tugas mendadak. Personel mendapatkan pelatihan tentang perangkat lunak dan platform yang merupakan dasar pembangunan beban kerja dan mereka didorong untuk mendapatkan sertifikasi. Ada cukup banyak personel sehingga orang dapat mengambil cuti sambil tetap ada dukungan untuk beban kerja dan rotasi personel yang siap tugas mendadak.

Langkah-langkah implementasi

1. Tetapkan jumlah personel yang memadai untuk mengoperasikan dan mendukung beban kerja Anda, termasuk tugas siap sedia, masalah keamanan, dan peristiwa siklus hidup, seperti tugas akhir dukungan dan rotasi sertifikat.
2. Latih personel Anda tentang perangkat lunak dan platform yang membentuk beban kerja Anda.
   a. Pelatihan dan Sertifikasi AWS memiliki pustaka kursus tentang AWS. Kursus-kursus ini disediakan gratis dan berbayar, baik secara online maupun tatap muka.
   b. AWS menyelenggarakan acara dan webinar tempat Anda belajar AWS dari para ahli.
3. Lakukan tindakan berikut secara rutin.
   - Lakukan evaluasi terhadap ukuran dan keterampilan tim seiring perubahan kondisi pengoperasian dan beban kerja.
   - Sesuaikan ukuran dan keterampilan tim agar memenuhi persyaratan-persyaratan operasional.
   - Periksa kemampuan dan kapasitas untuk mengatasi peristiwa siklus hidup yang direncanakan, peristiwa keamanan yang tidak direncanakan, dan notifikasi operasional melalui AWS Health.

Tingkat upaya untuk rencana implementasi: Tinggi. Mempekerjakan dan melatih tim untuk mendukung beban kerja mengharuskan Anda melakukan upaya yang cukup besar, tetapi hal itu akan memberikan manfaat besar dalam jangka panjang.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP04 Melakukan manajemen pengetahuan - Anggota tim harus memiliki informasi yang diperlukan untuk mengoperasikan dan mendukung beban kerja. Manajemen pengetahuan merupakan kunci untuk menyediakan informasi tersebut.

Dokumen terkait:

- Acara dan Webinar AWS
- Pelatihan dan Sertifikasi AWS

OPS07-BP02 Memastikan peninjauan yang konsisten terkait kesiapan operasional

Gunakan Peninjauan Kesiapan Operasional (ORR) untuk memvalidasi bahwa Anda dapat mengoperasikan beban kerja Anda. ORR adalah sebuah mekanisme yang dikembangkan di Amazon untuk memvalidasi bahwa tim dapat mengoperasikan beban kerja mereka dengan aman. ORR adalah sebuah proses peninjauan dan inspeksi menggunakan daftar periksa persyaratan. ORR adalah sebuah pengalaman layanan mandiri yang digunakan tim untuk memastikan beban kerja mereka. ORR mencakup praktik-praktik terbaik dari pelajaran yang kami dapatkan selama bertahun-tahun membangun perangkat lunak.

Daftar periksa ORR terdiri dari rekomendasi berkaitan dengan arsitektur, proses operasional, manajemen peristiwa, dan kualitas rilis. Proses Koreksi Kesalahan (CoE) kami merupakan pendorong utama item-item ini. Analisis pasca-insiden Anda sendiri harus mendorong pengembangan ORR Anda. ORR tidak hanya tentang mengikuti praktik terbaik tapi juga mencegah berulangnya peristiwa yang telah Anda lihat sebelumnya. Terakhir, keamanan, pengelolaan, dan kepatuhan terhadap persyaratan juga dapat disertakan di dalam sebuah ORR.

Jalankan ORR sebelum beban kerja meluncur ke ketersediaan umum dan kemudian ke seluruh siklus pengembangan perangkat lunak. Menjalankan ORR sebelum peluncuran dapat meningkatkan kemampuan Anda untuk mengoperasikan beban kerja dengan aman. Jalankan kembali ORR Anda secara berkala pada beban kerja untuk mengetahui penyimpangan dari praktik terbaik. Anda dapat memiliki daftar periksa ORR untuk peluncuran layanan baru dan ORR untuk peninjauan secara berkala. Ini membantu Anda untuk tetap up to date dengan praktik terbaik yang muncul dan

menggabungkan pelajaran yang didapatkan dari analisis pasca-insiden. Saat penggunaan cloud Anda sudah matang, Anda dapat membangun persyaratan ORR ke dalam arsitektur Anda secara default.

Hasil yang Diinginkan: Anda memiliki daftar periksa ORR dengan praktik terbaik untuk organisasi Anda. ORR dilakukan sebelum peluncuran beban kerja. ORR dijalankan secara berkala selama berlangsungnya siklus beban kerja.

Anti-pola umum:

- Anda meluncurkan beban kerja tanpa mengetahui apakah Anda dapat mengoperasikannya atau tidak.
- Persyaratan pengelolaan dan keamanan tidak diikutsertakan ketika menjamin peluncuran beban kerja.
- Beban kerja tidak dievaluasi kembali secara berkala.
- Beban kerja diluncurkan tanpa menerapkan prosedur-prosedur yang diperlukan.
- Anda melihat berulangnya kegagalan yang disebabkan akar masalah yang sama di beberapa beban kerja.

Manfaat menjalankan praktik terbaik ini:

- Beban kerja Anda mencakup praktik terbaik dalam hal arsitektur, proses, dan manajemen.
- Pelajaran yang didapatkan digabungkan dalam proses ORR.
- Prosedur yang diperlukan tersedia ketika beban kerja diluncurkan.
- ORR dijalankan di seluruh siklus perangkat lunak beban kerja Anda.

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

Panduan implementasi

ORR adalah dua hal: proses dan daftar periksa. Proses ORR Anda harus diadopsi oleh organisasi Anda dan didukung oleh seorang sponsor eksekutif. Minimal, ORR harus dilakukan sebelum beban kerja meluncur ke ketersediaan umum. Jalankan ORR di seluruh siklus pengembangan perangkat lunak untuk tetap up to date dengan praktik-praktik terbaik atau persyaratan baru. Daftar periksa ORR harus mencakup item-item konfigurasi, persyaratan keamanan dan pengelolaan, serta praktik terbaik dari organisasi Anda. Seiring waktu, Anda dapat menggunakan layanan-layanan, seperti AWS

Config, AWS Security Hub CSPM, dan AWS Control TowerGuardrails, untuk membangun praktik terbaik dari ORR ke pagar pembatas untuk deteksi otomatis atas praktik terbaik.

Contoh pelanggan

Setelah terjadi beberapa insiden produksi, AnyCompany Retail memutuskan untuk menerapkan proses ORR. Mereka membangun daftar periksa yang terdiri dari praktik terbaik, persyaratan pengelolaan dan kepatuhan, serta pelajaran yang didapatkan dari pemadaman (outage). Beban kerja baru melakukan ORR sebelum diluncurkan. Setiap beban kerja melakukan ORR setiap tahun dengan subset praktik terbaik untuk menggabungkan praktik-praktik terbaik dan persyaratan baru yang ditambahkan ke daftar periksa ORR. Seiring waktu, AnyCompany Retail digunakan AWS Config untuk mendeteksi beberapa praktik terbaik, mempercepat proses ORR.

Langkah-langkah implementasi

Untuk mempelajari lebih lanjut tentang ORR, baca laporan resmi Ulasan Kesiapan Operasional (ORR). Laporan resmi ini menyediakan detail informasi tentang riwayat proses ORR, cara membangun praktik ORR Anda sendiri, dan cara mengembangkan daftar periksa ORR Anda. Langkah-langkah berikut ini merupakan versi singkat dari dokumen tersebut. Untuk pemahaman yang mendalam tentang apa itu ORR dan cara membangunnya, sebaiknya baca laporan resmi tersebut.

1. Kumpulkan para pemangku kepentingan utama, termasuk para perwakilan dari bagian keamanan, operasi, dan pengembangan.
2. Minta setiap pemangku kepentingan untuk menyediakan setidaknya satu persyaratan. Untuk iterasi pertama, coba batasi jumlah item menjadi tiga puluh atau kurang.
   - Lampiran B: Contoh pertanyaan ORR dari laporan resmi Peninjauan Kesiapan Operasional (ORR) yang berisi sampel pertanyaan yang dapat Anda gunakan untuk memulai.
3. Kumpulkan persyaratan Anda ke dalam lembar kerja.
   - Anda dapat menggunakan lensa khusus di AWS Well-Architected Tool untuk mengembangkan ORR Anda dan membagikannya di seluruh akun dan Organisasi AWS Anda.
4. Identifikasi satu beban kerja untuk diberikan ORR. Idealnya adalah beban kerja sebelum peluncuran atau beban kerja internal.
5. Pelajari daftar periksa ORR dan catat semua penemuan yang dibuat. Penemuannya mungkin akan dapat diterima jika terdapat mitigasi. Untuk penemuan yang tidak memiliki mitigasi, tambahkan beban kerja ke backlog item Anda dan implementasikan sebelum peluncuran.
6. Seiring waktu, lanjutkan penambahan praktik terbaik dan persyaratan ke daftar periksa ORR Anda.

Pelanggan Dukungan dengan Dukungan Perusahaan dapat meminta Lokakarya Tinjauan Kesiapan Operasional dari Manajer Akun Teknis mereka. Lokakarya ini adalah sesi kerja mundur interaktif untuk membuat daftar periksa ORR Anda sendiri.

Tingkat upaya untuk rencana implementasi: Tinggi. Untuk mengadopsi praktik ORR pada organisasi Anda diperlukan sponsor eksekutif dan dukungan pemangku kepentingan. Buat dan perbarui daftar periksa dengan masukan dari seluruh organisasi Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP03 Mengevaluasi persyaratan tata kelola – Persyaratan tata kelola sangat sesuai untuk daftar periksa ORR.
- OPS01-BP04 Evaluasi persyaratan kepatuhan – Terkadang persyaratan kepatuhan tercantum di daftar periksa ORR. Terkadang persyaratan kepatuhan adalah proses yang terpisah.
- OPS03-BP07 Tim sumber daya dengan tepat – Kemampuan tim merupakan kandidat yang bagus untuk persyaratan ORR.
- OPS06-BP01 Rencana untuk perubahan yang gagal – Rencana rollback atau rollforward harus dibuat sebelum Anda meluncurkan beban kerja Anda.
- OPS07-BP01 Memastikan kemampuan personel – Untuk mendukung beban kerja, Anda harus memiliki personel yang diperlukan.
- SEC01-BP03 Mengidentifikasi dan memvalidasi tujuan kontrol – Tujuan kontrol keamanan membuat persyaratan ORR yang sangat baik.
- REL13-BP01 Tentukan tujuan pemulihan untuk waktu henti dan kehilangan data – Rencana pemulihan bencana adalah persyaratan ORR yang baik.
- COST02-BP01 Kembangkan kebijakan berdasarkan persyaratan organisasi Anda – Kebijakan manajemen biaya baik untuk dimasukkan dalam daftar periksa ORR Anda.

Dokumen terkait:

- AWS Control Tower - Pagar Pembatas di AWS Control Tower
- AWS Well-Architected Tool - Lensa Kustom
- Templat Peninjauan Kesiapan Operasional oleh Adrian Hornsby
- Laporan Resmi Peninjauan Kesiapan Operasional (ORR)

Video terkait:

- AWS Dukungan Anda | Membangun Peninjauan Kesiapan Operasional (ORR) yang Efektif

Contoh terkait:

- Sampel Lensa Peninjauan Kesiapan Operasional (ORR)

Layanan terkait:

- AWS Config
- AWS Control Tower
- AWS Security Hub CSPM
- AWS Well-Architected Tool

OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur

Runbook adalah sebuah proses terdokumentasi untuk meraih hasil tertentu. Runbook terdiri dari serangkaian langkah yang diikuti seseorang untuk menyelesaikan sesuatu. Runbook telah digunakan dalam operasi sejak masa-masa awal industri penerbangan. Dalam operasi cloud, kita menggunakan runbook untuk mengurangi risiko dan meraih hasil-hasil yang diinginkan. Dalam bentuk paling sederhananya, runbook adalah daftar periksa untuk menyelesaikan tugas.

Runbook adalah bagian penting dari operasi beban kerja Anda. Mulai dari pelaksanaan orientasi untuk anggota tim baru hingga melakukan deployment rilis besar, runbook adalah proses terkodifikasi yang dapat memberikan hasil-hasil yang konsisten, siapa pun yang menggunakannya. Runbook harus dipublikasikan di lokasi sentral dan diperbarui seiring berkembangnya proses karena memperbarui runbook adalah komponen utama dari proses manajemen perubahan. Runbook juga harus menyertakan panduan tentang cara menangani kesalahan, alat, izin, pengecualian, dan eskalasi jika terjadi masalah.

Saat organisasi Anda matang, mulailah mengotomatiskan runbook. Mulailah dengan runbook yang singkat dan sering kali digunakan. Gunakan bahasa skrip untuk mengotomatiskan langkah-langkah atau mempermudah pelaksanaan langkah-langkah. Seiring Anda mengotomatiskan beberapa runbook pertama, Anda harus mendedikasikan waktu untuk mengotomatiskan runbook yang lebih kompleks. Seiring waktu, sebagian besar runbook Anda harus diotomatiskan dalam cara tertentu.

Hasil yang diinginkan: Tim Anda memiliki kumpulan panduan langkah demi langkah untuk melakukan tugas-tugas beban kerja. Runbook berisi hasil yang diinginkan, alat dan izin yang diperlukan, serta petunjuk untuk penanganan kesalahan. Runbook disimpan di sebuah lokasi sentral (sistem kontrol versi) dan sering diperbarui. Misalnya, runbook Anda menyediakan kemampuan bagi tim Anda untuk memantau, mengomunikasikan, dan merespons peristiwa AWS Health untuk akun-akun penting selama terjadi alarm aplikasi, masalah operasional, dan peristiwa siklus hidup yang sudah direncanakan.

Anti-pola umum:

- Mengandalkan ingatan untuk menyelesaikan setiap langkah dari suatu proses.
- Menerapkan perubahan secara manual tanpa menggunakan daftar periksa.
- Anggota tim yang berbeda-beda melakukan proses yang sama, tetapi dengan langkah atau hasil yang berbeda.
- Membiarkan runbook tidak selaras dengan perubahan sistem dan otomatisasi.

Manfaat menjalankan praktik terbaik ini:

- Mengurangi tingkat kesalahan untuk tugas-tugas yang dilakukan manual.
- Operasi dilakukan secara konsisten.
- Anggota tim baru dapat mulai melakukan tugas dengan lebih cepat.
- Runbook dapat diotomatiskan untuk mengurangi upaya yang diperlukan.

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

Panduan implementasi

Runbook dapat memiliki beberapa bentuk, bergantung pada tingkat kematangan organisasi Anda. Minimal, runbook harus terdiri dari dokumen teks langkah demi langkah. Hasil yang diinginkan harus ditunjukkan dengan jelas. Buatlah dokumentasi yang jelas mengenai izin atau alat khusus yang diperlukan. Berikan panduan mendetail tentang cara menangani kesalahan dan cara melakukan eskalasi jika terjadi kesalahan. Cantumkan pemilik runbook dan publikasikan di sebuah lokasi sentral. Setelah runbook Anda didokumentasikan, kemudian validasi dengan meminta orang lain di tim Anda untuk menjalankannya. Seiring berkembangnya prosedur, perbarui runbook Anda sesuai dengan proses manajemen perubahan Anda.

Runbook teks Anda harus diotomatiskan seiring semakin matangnya organisasi Anda. Dengan menggunakan layanan-layanan seperti AWS Systems Manager Automation, Anda dapat mentransformasikan teks biasa menjadi otomatisasi yang dapat dijalankan dengan beban kerja Anda. Otomatisasi ini dapat dijalankan sebagai respons terhadap peristiwa, sehingga mengurangi beban operasional untuk memelihara beban kerja Anda. AWS Systems Manager Automation juga menyediakan pengalaman desain visual kode rendah untuk membuat runbook otomatisasi dengan lebih mudah.

Contoh pelanggan

AnyCompany Retail harus melakukan pembaruan skema basis data selama deployment perangkat lunak. Tim Operasi Cloud bekerja sama dengan Tim Administrasi Basis Data untuk membuat sebuah runbook guna menerapkan perubahan ini secara manual. Runbook ini mencantumkan setiap langkah yang ada dalam prosesnya dalam bentuk daftar periksa. Runbook ini berisi sebuah bagian yang menjelaskan cara menangani kesalahan saat terjadi kesalahan. Mereka memublikasikan runbook di wiki internal mereka bersama dengan runbook mereka yang lain. Tim Operasi Cloud berencana untuk mengotomatiskan runbook dalam sprint mendatang.

Langkah-langkah implementasi

Jika Anda belum memiliki repositori dokumen, repositori kontrol versi bisa menjadi tempat yang tepat untuk mulai membangun pustaka runbook Anda. Anda dapat membangun runbook Anda dengan menggunakan Markdown. Kami telah menyediakan contoh templat runbook yang dapat Anda gunakan untuk mulai membangun runbook.

```
# Runbook Title
## Runbook Info
| Runbook ID | Description | Tools Used | Special Permissions | Runbook Author | Last
 Updated | Escalation POC |
|-------|-------|-------|-------|-------|-------|-------|
| RUN001 | What is this runbook for? What is the desired outcome? | Tools | Permissions
 | Your Name | 2022-09-21 | Escalation Name |
## Steps
1. Step one
2. Step two
```

1. Jika Anda belum memiliki repositori atau wiki dokumentasi, buatlah repositori kontrol versi baru di sistem kontrol versi Anda.
2. Identifikasi proses yang tidak memiliki runbook. Proses yang ideal adalah proses yang dilakukan secara semi-reguler, sedikit jumlah langkahnya, dan memiliki kegagalan yang dampaknya rendah.

3. Di repositori dokumen Anda, buat draf dokumen Markdown baru dengan menggunakan templat tersebut. Isi Judul Runbook dan bidang-bidang yang wajib diisi di bawah Info Runbook.
4. Dimulai dengan langkah pertama, isi bagian Langkah-langkah dalam runbook.
5. Berikan runbook kepada anggota tim. Minta mereka menggunakan runbook ini untuk memvalidasi langkah-langkahnya. Jika ada sesuatu yang belum dimasukkan atau memerlukan kejelasan, perbarui runbook ini.
6. Publikasikan runbook ini ke bagian penyimpanan dokumentasi internal Anda. Setelah dipublikasikan, beri tahu tim Anda dan para pemangku kepentingan lainnya.
7. Seiring waktu, Anda akan membangun sebuah pustaka runbook. Saat pustaka tersebut bertambah besar, mulailah bekerja untuk mengotomatiskan runbook.

Tingkat upaya untuk rencana implementasi: Rendah. Standar minimum untuk runbook adalah panduan teks langkah demi langkah. Mengotomatiskan runbook dapat meningkatkan upaya penerapan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi
- OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah
- OPS10-BP01 Menggunakan proses untuk manajemen peristiwa, insiden, dan masalah
- OPS10-BP02 Menjalankan proses untuk setiap peringatan
- OPS11-BP04 Menjalankan manajemen pengetahuan

Dokumen terkait:

- Mencapai Keunggulan Operasional menggunakan playbook dan runbook otomatis
- AWS Systems Manager: Bekerja dengan runbook
- Playbook migrasi untuk migrasi besar AWS - Tugas 4: Meningkatkan runbook migrasi Anda
- Gunakan runbook AWS System Manager Automation untuk menyelesaikan tugas-tugas operasional

Video terkait:

- AWS re:Invent 2019: Panduan mandiri untuk runbook, laporan insiden, dan respons insiden
- Cara melakukan otomatisasi Operasi IT di AWS | Amazon Web Services
- Integrasikan Skrip ke dalam AWS Systems Manager

Contoh terkait:

- Lab Well-Architected: Melakukan otomatisasi operasi dengan Playbook dan Runbook
- AWS Postingan Blog : Membangun Praktik Otomatisasi Cloud untuk Keunggulan Operasional: Praktik Terbaik dari AWS Managed Services
- AWS Systems Manager: Panduan otomatisasi
- AWS Systems Manager: Pulihkan volume root dari snapshot runbook terbaru
- Membangun runbook respons insiden AWS dengan menggunakan notebook Jupyter dan CloudTrail Lake
- Gitlab - Runbook
- Rubix – Pustaka Python untuk membuat runbook di Notebook Jupyter
- Menggunakan Document Builder untuk membuat runbook kustom

Layanan terkait:

- AWS Systems Manager Automation

OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah

Playbook adalah panduan mendetail yang digunakan untuk menyelidiki insiden. Ketika terjadi sebuah insiden, playbook digunakan untuk menyelidiki, membuat cakupan dampak, dan mengidentifikasi akar masalah penyebabnya. Playbook digunakan untuk berbagai skenario, dari deployment yang gagal hingga insiden keamanan. Dalam banyak kasus, playbook mengidentifikasi akar masalah yang dimitigasi dengan menggunakan runbook. Playbook adalah komponen pokok dalam rencana respons insiden organisasi Anda.

Playbook yang baik memiliki sejumlah fitur utama. Playbook memberikan panduan secara mendetail bagi pengguna, dalam proses penemuan. Dengan berpikir secara menyeluruh, langkah apa saja yang sebaiknya diikuti seseorang untuk mendiagnosis sebuah insiden? Tetapkan secara jelas di dalam playbook apakah alat-alat khusus atau izin yang lebih tinggi diperlukan di dalam playbook.

Membuat sebuah rencana komunikasi untuk memberikan informasi terbaru kepada para pemangku kepentingan mengenai status penyelidikan adalah komponen utama. Dalam situasi ketika akar penyebab masalah tidak dapat diidentifikasi, playbook harus memiliki rencana eskalasi. Jika akar penyebab masalah sudah diidentifikasi, playbook harus mengarah ke sebuah runbook yang menjelaskan cara menyelesaikannya. Playbook harus disimpan secara terpusat dan dipelihara secara rutin. Jika playbook digunakan untuk pemberitahuan khusus, bekali tim Anda dengan penunjuk ke playbook yang ada di dalam pemberitahuan tersebut.

Otomatisasi playbook Anda seiring dengan kematangan organisasi. Mulailah dengan playbook yang mencakup insiden-insiden berisiko rendah. Gunakan penulisan skrip untuk mengotomatiskan langkah-langkah penemuan. Pastikan Anda memiliki runbook pendamping untuk memitigasi akar masalah umum.

Hasil yang diinginkan: Organisasi Anda memiliki playbook untuk insiden umum. Playbook disimpan di lokasi terpusat dan tersedia untuk anggota tim Anda. Playbook harus sering diperbarui. Runbook pendamping dibuat untuk akar masalah apa pun yang diketahui.

Anti-pola umum:

- Tidak ada cara standar untuk menyelidiki sebuah insiden.
- Anggota tim mengandalkan memori otot atau pengetahuan kelembagaan untuk memecahkan masalah kegagalan deployment.
- Anggota tim baru mempelajari cara menyelidiki permasalahan melalui coba-coba (trial and error).
- Praktik terbaik untuk menyelidiki permasalahan tidak dibagikan ke seluruh tim.

Manfaat menjalankan praktik terbaik ini:

- Playbook meningkatkan upaya Anda untuk memitigasi insiden.
- Anggota tim yang berbeda-beda dapat menggunakan playbook yang sama untuk mengidentifikasi akar penyebab masalah secara konsisten.
- Setelah akar penyebab masalah diketahui, kemudian bisa dikembangkan runbook untuk masalah tersebut, sehingga dapat mempercepat waktu pemulihan.
- Playbook akan membantu anggota tim untuk mulai berkontribusi lebih cepat.
- Tim dapat menskalakan proses mereka dengan playbook yang dapat diulangi langkah-langkahnya.

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

Panduan implementasi

Cara Anda membangun dan menggunakan playbook bergantung pada kematangan organisasi Anda. Jika Anda baru mengenal cloud, buatlah playbook dalam bentuk teks di dalam repositori dokumen pusat. Seiring dengan kematangan organisasi, playbook bisa dibuat menjadi semi-otomatis dengan bahasa skrip seperti Python. Skrip-skrip ini dapat dijalankan di dalam notebook Jupyter untuk mempercepat penemuan. Organisasi tingkat lanjut memiliki playbook yang sepenuhnya otomatis untuk permasalahan-permasalahan umum yang diperbaiki secara otomatis dengan runbook.

Mulailah membangun playbook Anda dengan mengidentifikasi insiden-insiden umum yang terjadi pada beban kerja Anda. Untuk mengawali, pilihlah playbook untuk insiden-insiden dengan risiko rendah dan dengan akar penyebab masalah yang telah dipersempit menjadi beberapa permasalahan. Setelah Anda memiliki playbook untuk skenario yang lebih sederhana, beralihlah ke skenario yang memiliki risiko lebih tinggi atau skenario dengan akar penyebab masalah yang tidak diketahui dengan baik.

Playbook teks Anda harus diotomatiskan seiring dengan kematangan organisasi Anda. Menggunakan layanan seperti AWS Systems Manager Automation, teks datar dapat ditransformasi menjadi otomatisasi. Otomatisasi ini dapat dijalankan terhadap beban kerja untuk mempercepat penyelidikan. Otomatisasi ini dapat diaktifkan untuk merespons peristiwa, sehingga akan mengurangi rata-rata waktu untuk menemukan dan menyelesaikan insiden.

Pelanggan dapat menggunakan AWS Systems Manager Incident Manager untuk menanggapi insiden. Layanan ini menyediakan satu antarmuka untuk memeriksa insiden, memberikan informasi kepada para pemangku kepentingan saat proses penemuan dan mitigasi, dan berkolaborasi selama berlangsungnya insiden. Layanan ini menggunakan AWS System Manager Automation untuk mempercepat deteksi dan pemulihan.

Contoh pelanggan

Insiden produksi memberikan dampak pada AnyCompany Retail. Rekayasawan yang siap dipanggil kapan saja (on-call) menggunakan playbook untuk menyelidiki permasalahan. Seiring mereka mengikuti langkah-langkahnya, mereka terus memutakhirkan para pemangku kepentingan utama yang diidentifikasi di dalam playbook. Rekayasawan mengidentifikasi akar penyebab masalah sebagai kondisi pacu di dalam layanan backend. Dengan menggunakan runbook, rekayasawan meluncurkan ulang layanan, sehingga AnyCompany Retail dapat kembali online.

Langkah-langkah implementasi

Jika Anda belum memiliki repositori dokumen, kami menyarankan Anda untuk membuat repositori kontrol versi untuk pustaka playbook Anda. Anda dapat membangun playbook Anda dengan menggunakan Markdown, yang kompatibel dengan sebagian besar sistem otomatisasi playbook. Jika Anda memulai dari nol, gunakan contoh templat playbook berikut ini.

```
# Playbook Title
## Playbook Info
| Playbook ID | Description | Tools Used | Special Permissions | Playbook Author | Last
 Updated | Escalation POC | Stakeholders | Communication Plan |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| RUN001 | What is this playbook for? What incident is it used for? | Tools |
 Permissions | Your Name | 2022-09-21 | Escalation Name | Stakeholder Name | How will
 updates be communicated during the investigation? |
## Steps
1. Step one
2. Step two
```

1. Jika Anda belum memiliki repositori dokumen atau wiki, buatlah repositori kontrol versi baru untuk playbook Anda di sistem kontrol versi Anda.
2. Identifikasi permasalahan umum yang memerlukan penyelidikan. Ini sebaiknya adalah skenario dengan akar penyebab masalah yang dibatasi ke beberapa permasalahan dan penyelesaian yang memiliki risiko rendah.
3. Menggunakan templat Markdown, isilah bagian Nama Playbook dan bidang di bawah Info Playbook.
4. Lengkapi langkah-langkah pemecahan masalah. Sampaikan dengan sejelas mungkin tindakan-tindakan yang akan dilakukan atau area apa saja yang harus Anda selidiki.
5. Berikan playbook tersebut kepada anggota tim dan minta mereka mempelajari dan memvalidasinya. Jika terdapat hal yang terlewatkan atau tidak jelas, lakukan pembaruan playbook.
6. Terbitkan playbook di dalam repositori dokumen Anda dan informasikan kepada tim dan pemangku kepentingan.
7. Pustaka playbook ini akan tumbuh seiring dengan semakin banyaknya playbook yang Anda tambahkan. Setelah Anda memiliki beberapa playbook, mulailah mengotomatiskannya menggunakan alat seperti AWS Systems Manager Automation untuk terus menyinkronkan otomatisasi dan playbook.

Tingkat upaya untuk rencana implementasi: Rendah. Playbook Anda harus berupa dokumen teks yang disimpan di sebuah lokasi terpusat. Organisasi yang lebih matang akan beralih ke playbook otomatis.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi
- OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur
- OPS10-BP01 Menggunakan proses untuk manajemen peristiwa, insiden, dan masalah
- OPS10-BP02 Menjalankan proses untuk setiap peringatan
- OPS11-BP04 Menjalankan manajemen pengetahuan

Dokumen terkait:

- Mencapai Keunggulan Operasional menggunakan playbook dan runbook otomatis
- AWS Systems Manager: Bekerja dengan runbook
- Gunakan runbook AWS System Manager Automation untuk menyelesaikan tugas-tugas operasional

Video terkait:

- AWS re:Invent 2019: Panduan mandiri untuk runbook, laporan insiden, dan respons insiden (SEC318-R1)
- Incident Manager AWS Systems Manager - Lokakarya AWS Virtual
- Integrasikan Skrip ke dalam AWS Systems Manager

Contoh terkait:

- AWS Kerangka Kerja Playbook Pelanggan
- AWS Systems Manager: Panduan otomatisasi
- Membangun runbook respons insiden AWS dengan menggunakan notebook Jupyter dan CloudTrail Lake
- Rubix – Pustaka Python untuk membuat runbook di Notebook Jupyter

- Menggunakan Document Builder untuk membuat runbook kustom

Layanan terkait:

- AWS Systems Manager Automation
- AWS Peluncuran Systems Manager Incident Manager

OPS07-BP05 Membuat keputusan yang tepat untuk melakukan deployment sistem dan perubahan

Miliki proses untuk perubahan yang sukses dan tidak sukses pada beban kerja Anda. Pre-mortem adalah latihan simulasi tim terhadap terjadinya kegagalan untuk mengembangkan strategi mitigasi. Gunakan pre-mortem untuk mengantisipasi kegagalan dan menciptakan prosedur ketika diperlukan. Evaluasi manfaat dan risiko dari deployment perubahan ke beban kerja Anda. Pastikan apakah semua perubahan sudah mematuhi tata kelola atau tidak.

Hasil yang diinginkan:

- Anda mengambil keputusan yang tepat ketika melakukan deployment perubahan ke beban kerja Anda.
- Perubahan mematuhi tata kelola.

Anti-pola umum:

- Melakukan deployment perubahan ke beban kerja tanpa proses untuk menangani deployment yang gagal.
- Membuat perubahan pada lingkungan produksi Anda yang tidak mematuhi persyaratan-persyaratan tata kelola.
- Melakukan deployment versi beban kerja baru Anda tanpa menetapkan garis dasar untuk pemanfaatan sumber daya.

Manfaat menjalankan praktik terbaik ini:

- Anda siap untuk menangani perubahan-perubahan yang tidak berhasil dilakukan pada beban kerja Anda.
- Perubahan pada beban kerja Anda mematuhi kebijakan-kebijakan tata kelola.

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

Panduan implementasi

Gunakan pre-mortem untuk mengembangkan proses untuk perubahan yang tidak berhasil. Buatlah dokumentasi dari proses-proses Anda untuk perubahan yang tidak berhasil. Pastikan semua perubahan mematuhi tata kelola. Evaluasi manfaat dan risiko melakukan deployment perubahan ke beban kerja Anda.

Contoh pelanggan

AnyCompany Retail melakukan pre-mortem secara teratur untuk melakukan validasi terhadap proses-proses mereka untuk perubahan yang tidak berhasil. Mereka mendokumentasikan proses mereka di Wiki bersama dan sering kali memperbaruinya. Semua perubahan mematuhi persyaratan-persyaratan tata kelola.

Langkah-langkah implementasi

1. Ambil keputusan yang tepat ketika melakukan deployment perubahan ke beban kerja Anda. Tetapkan dan tinjau kriteria untuk deployment yang berhasil. Kembangkan skenario atau kriteria yang akan menginisiasi pengembalian perubahan ke versi sebelumnya. Pikirkan manfaat dari deployment perubahan dibandingkan dengan risiko perubahan yang tidak berhasil.
2. Pastikan bahwa semua perubahan mematuhi kebijakan tata kelola.
3. Gunakan pre-mortem guna membuat rencana untuk perubahan-perubahan yang tidak berhasil dan mendokumentasikan strategi mitigasi. Jalankan sesi latihan table-top untuk memperagakan perubahan yang tidak berhasil dan melakukan validasi terhadap prosedur pengembalian ke versi sebelumnya (roll-back).

Tingkat upaya untuk rencana implementasi: Sedang. Mengimplementasikan praktik pre-mortem memerlukan koordinasi dan upaya dari para pemangku kepentingan dalam seluruh organisasi Anda

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP03 Mengevaluasi persyaratan tata kelola - Persyaratan tata kelola merupakan faktor kunci dalam menentukan apakah akan melakukan deployment perubahan.
- OPS06-BP01 Rencana untuk perubahan yang gagal - Buat rencana untuk memitigasi deployment yang gagal dan gunakan pre-mortem untuk memvalidasinya.

- OPS06-BP02 Uji penerapan - Setiap perubahan perangkat lunak harus diuji dengan tepat sebelum deployment untuk mengurangi kecacatan dalam produksi.
- OPS07-BP01 Memastikan kemampuan personel - Memiliki cukup banyak personel yang terlatih untuk mendukung beban kerja sangat penting dalam mengambil keputusan yang tepat dalam hal deployment perubahan sistem.

Dokumen terkait:

- Amazon Web Services: Risiko dan Kepatuhan
- Model Tanggung Jawab Bersama AWS
- Tata Kelola dalam AWS Cloud: Keseimbangan yang Tepat Antara Kelincahan dan Keamanan

OPS07-BP06 Membuat rencana dukungan untuk beban kerja produksi

Aktifkan dukungan untuk perangkat lunak dan layanan yang diandalkan beban kerja produksi Anda. Pilih tingkat dukungan yang sesuai untuk memenuhi kebutuhan tingkat layanan produksi Anda. Rencana dukungan untuk dependensi ini diperlukan untuk berjaga-jaga jika ada gangguan layanan atau masalah perangkat lunak yang terjadi. Buatlah dokumentasi dari rencana-rencana dukungan dan cara meminta dukungan untuk semua vendor perangkat lunak dan layanan. Implementasikan mekanisme yang memastikan bahwa titik kontak dukungan selalu yang terbaru.

Hasil yang diinginkan:

- Implementasikan rencana dukungan untuk perangkat lunak dan layanan yang diandalkan oleh beban kerja produksi.
- Pilih rencana dukungan yang sesuai berdasarkan kebutuhan tingkat layanan.
- Buatlah dokumentasi dari rencana dukungan, tingkat dukungan, dan cara meminta dukungan.

Anti-pola umum:

- Anda tidak memiliki rencana dukungan untuk vendor perangkat lunak yang penting. Beban kerja Anda terkena dampaknya dan Anda tidak dapat melakukan apa-apa untuk mempercepat perbaikan atau mendapatkan informasi terbaru dari vendor secara tepat waktu.
- Seorang pengembang yang merupakan titik utama kontak untuk vendor perangkat lunak tidak lagi bekerja di perusahaan. Anda tidak dapat menghubungi dukungan vendor secara langsung. Anda

harus meluangkan waktu menelusuri dan mencari-cari dalam sistem kontak generik, sehingga menambah waktu yang diperlukan untuk memberikan respons ketika diperlukan.

- Penghentian (outage) produksi terjadi pada vendor perangkat lunak. Tidak ada dokumentasi tentang cara mengajukan kasus dukungan.

Manfaat menjalankan praktik terbaik ini:

- Dengan tingkat dukungan yang sesuai, Anda dapat memperoleh respons dalam kerangka waktu yang diperlukan untuk memenuhi kebutuhan-kebutuhan tingkat layanan.
- Sebagai pelanggan yang didukung, Anda dapat menyampaikan masalah, jika terjadi masalah produksi.
- Vendor layanan dan perangkat lunak dapat membantu menyelesaikan masalah saat terjadi insiden.

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

Panduan implementasi

Aktifkan rencana dukungan untuk vendor perangkat lunak dan layanan yang diandalkan oleh beban kerja produksi Anda. Atur rencana dukungan yang sesuai untuk memenuhi kebutuhan-kebutuhan tingkat layanan Anda. Untuk pelanggan AWS, ini artinya Anda mengaktifkan Business Support AWS atau yang lebih tinggi pada setiap akun yang memiliki beban kerja produksi. Temui para vendor dukungan secara teratur untuk mendapatkan informasi terbaru mengenai penawaran dukungan, proses, dan kontak. Buatlah dokumentasi tentang cara meminta dukungan dari para vendor perangkat lunak dan layanan, termasuk cara menyampaikan masalah jika ada penghentian (outage). Implementasikan mekanisme-mekanisme untuk menjaga agar kontak selalu yang terbaru.

Contoh pelanggan

Di AnyCompany Retail, semua dependensi layanan dan perangkat lunak komersial memiliki rencana dukungan. Contohnya, mereka mengaktifkan Dukungan Perusahaan AWS di semua akun yang memiliki beban kerja produksi. Semua pengembang dapat membuka kasus dukungan bila ada masalah yang terjadi. Ada satu halaman wiki yang memuat informasi tentang cara meminta dukungan, siapa yang harus diberi tahu, dan praktik-praktik terbaik untuk mempercepat penanganan kasus.

Langkah-langkah implementasi

1. Bekerjasamalah dengan para pemangku kepentingan yang ada di organisasi Anda untuk mengidentifikasi para vendor perangkat lunak dan layanan yang diandalkan oleh beban kerja Anda. Buatlah dokumentasi mengenai dependensi ini.
2. Tentukan kebutuhan tingkat layanan untuk beban kerja Anda. Pilih rencana dukungan yang selaras dengannya.
3. Untuk layanan-layanan dan perangkat lunak komersial, tetapkan rencana dukungan dengan para vendornya.
    a. Dengan berlangganan AWS Business Support atau yang lebih tinggi untuk semua akun produksi, waktu respons AWS Dukungan akan lebih cepat dan hal ini sangat disarankan. Jika Anda tidak memiliki dukungan premium, Anda harus memiliki rencana tindakan untuk menangani masalah, yang memerlukan bantuan dari AWS Dukungan. AWS Dukungan memberikan kombinasi alat dan teknologi, orang, dan program yang dirancang untuk secara proaktif membantu Anda mengoptimalkan performa, menurunkan biaya, dan berinovasi dengan lebih cepat. Selain itu, AWS Business Support memberikan manfaat tambahan, termasuk akses API ke AWS Trusted Advisor dan AWS Health untuk integrasi programatis dengan sistem Anda, bersama metode akses lainnya seperti Konsol Manajemen AWS dan saluran Amazon EventBridge.
4. Buatlah dokumentasi tentang rencana dukungan di alat manajemen pengetahuan Anda. Sertakan cara untuk meminta dukungan, siapa yang harus diberi tahu jika ada kasus dukungan diajukan, dan cara untuk menyampaikan masalah saat terjadi insiden. Wiki bisa menjadi sebuah mekanisme yang bagus untuk memungkinkan semua orang membuat pembaruan yang diperlukan pada dokumentasi ketika mereka mengetahui tentang adanya perubahan yang dilakukan untuk mendukung proses atau perubahan kontak.

Tingkat upaya untuk rencana implementasi: Rendah. Sebagian besar vendor perangkat lunak dan layanan menawarkan pilihan penyertaan rencana dukungan. Mendokumentasikan dan berbagi praktik terbaik terkait dukungan di sistem manajemen pengetahuan Anda akan memastikan tim Anda mengetahui tindakan-tindakan yang harus dilakukan jika ada masalah produksi.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP02 Proses dan Prosedur memiliki pemilik teridentifikasi

Dokumen terkait:

- Rencana AWS Dukungan

Layanan terkait:

- AWS Business Support
- Dukungan Perusahaan AWS

Jalankan

Observabilitas memungkinkan Anda fokus pada data yang bermakna serta memahami interaksi dan output beban kerja Anda. Dengan berkonsentrasi pada wawasan penting dan menghilangkan data yang tidak perlu, Anda mempertahankan pendekatan langsung untuk memahami kinerja beban kerja. Hal ini sangat penting tidak hanya untuk mengumpulkan data tetapi juga untuk menafsirkannya dengan benar. Menentukan garis acuan yang jelas, menetapkan ambang batas peringatan yang sesuai, dan memantau secara aktif setiap penyimpangan. Pergeseran metrik kunci, terutama ketika berkorelasi dengan data lain, dapat menunjukkan dengan tepat area masalah tertentu. Dengan observabilitas, Anda lebih siap untuk memperkirakan dan mengatasi tantangan potensial, memastikan bahwa beban kerja Anda beroperasi dengan lancar dan memenuhi kebutuhan bisnis.

Keberhasilan operasi beban kerja diukur dengan pencapaian hasil bisnis dan pelanggan. Tetapkan hasil yang diharapkan, tentukan bagaimana keberhasilan akan diukur, dan identifikasi metrik yang akan digunakan pada perhitungan tersebut untuk menentukan apakah beban kerja dan operasi Anda berhasil. Kondisi operasional meliputi kondisi beban kerja serta kondisi dan keberhasilan aktivitas operasi yang dilakukan dalam dukungan beban kerja (misalnya, deployment dan respons insiden). Tetapkan baris acuan metrik untuk peningkatan, investigasi, serta intervensi, kumpulkan dan analisis metrik Anda, kemudian validasi pemahaman Anda tentang keberhasilan operasi dan bagaimana hal tersebut berubah seiring waktu. Gunakan metrik yang dikumpulkan untuk menentukan apakah Anda memenuhi kebutuhan pelanggan dan bisnis, serta mengidentifikasi area yang perlu ditingkatkan.

Manajemen peristiwa operasional yang efektif dan efisien diperlukan untuk mencapai keunggulan operasional. Hal ini berlaku untuk peristiwa operasional baik yang terencana maupun tidak terencana. Gunakan runbook yang telah dibuat untuk peristiwa yang dipahami dengan baik, dan gunakan buku panduan untuk membantu investigasi dan resolusi masalah. Prioritaskan respons terhadap peristiwa berdasarkan dampaknya pada bisnis dan pelanggan. Pastikan bahwa jika muncul peringatan sebagai respons terhadap suatu peristiwa, ada proses terkait untuk dijalankan, dengan pemilik yang diidentifikasi secara spesifik. Tentukan terlebih dulu personel yang dibutuhkan untuk menyelesaikan suatu peristiwa dan sertakan proses eskalasi agar dapat melibatkan personel tambahan, jika diperlukan, berdasarkan urgensi dan dampaknya. Identifikasi dan libatkan individu yang memiliki wewenang untuk membuat keputusan mengenai tindakan yang akan menimbulkan dampak bisnis dari respons peristiwa yang belum ditangani sebelumnya.

Komunikasikan status operasional beban kerja melalui dasbor dan pemberitahuan yang disesuaikan dengan audiens target (misalnya, pelanggan, bisnis, pengembang, operasi) sehingga mereka bisa

mengambil tindakan yang sesuai, ekspektasi mereka terkelola, serta mereka mendapatkan informasi ketika operasi kembali normal.

Di AWS, Anda dapat membuat tampilan dasbor metrik Anda yang dikumpulkan dari beban kerja dan secara native dari AWS. Anda dapat memanfaatkan CloudWatch atau aplikasi pihak ketiga untuk menggabungkan dan mempresentasikan tampilan tingkat bisnis, beban kerja, dan operasi terkait aktivitas operasi. AWS menyediakan wawasan beban kerja melalui kemampuan pencatatan yang mencakup AWS X-Ray, CloudWatch, CloudTrail, dan Log Alur VPC untuk mengidentifikasi masalah beban kerja dalam mendukung analisis akar masalah dan perbaikan.

Semua metrik yang Anda kumpulkan harus selaras dengan kebutuhan bisnis dan hasil yang didukung. Kembangkan respons dalam skrip untuk memahami peristiwa dengan baik dan otomatiskan respons tersebut saat ada peristiwa yang dikenali.

Topik

- Memanfaatkan observabilitas beban kerja
- Memahami kesehatan operasional
- Merespons peristiwa

Memanfaatkan observabilitas beban kerja

Memastikan kondisi beban kerja yang optimal dengan memanfaatkan observabilitas. Memanfaatkan metrik, log, dan jejak yang relevan untuk mendapatkan pandangan komprehensif tentang kinerja beban kerja Anda dan mengatasi masalah secara efisien.

Observabilitas memungkinkan Anda fokus pada data yang bermakna serta memahami interaksi dan output beban kerja Anda. Dengan berkonsentrasi pada wawasan penting dan menghilangkan data yang tidak perlu, Anda mempertahankan pendekatan langsung untuk memahami kinerja beban kerja.

Hal ini sangat penting tidak hanya untuk mengumpulkan data tetapi juga untuk menafsirkannya dengan benar. Menentukan garis acuan yang jelas, menetapkan ambang batas peringatan yang sesuai, dan memantau secara aktif setiap penyimpangan. Pergeseran metrik kunci, terutama ketika berkorelasi dengan data lain, dapat menunjukkan dengan tepat area masalah tertentu.

Dengan observabilitas, Anda lebih siap untuk memperkirakan dan mengatasi tantangan potensial, memastikan bahwa beban kerja Anda beroperasi dengan lancar dan memenuhi kebutuhan bisnis.

AWS menawarkan alat-alat khusus seperti Amazon CloudWatch untuk pemantauan dan pencatatan log, AWS X-Ray dan untuk penelusuran terdistribusi. Layanan ini terintegrasi dengan mudah dengan

berbagai sumber daya AWS, memungkinkan pengumpulan data yang efisien, menyiapkan peringatan berdasarkan ambang batas yang telah ditentukan, dan menyajikan data di dasbor untuk interpretasi yang mudah. Dengan memanfaatkan wawasan ini, Anda dapat membuat keputusan berdasarkan data yang matang yang sesuai dengan tujuan operasional Anda.

Praktik terbaik

- OPS08-BP01 Menganalisis metrik beban kerja
- OPS08-BP02 Menganalisis log beban kerja
- OPS08-BP03 Menganalisis jejak beban kerja
- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti
- OPS08-BP05 Membuat dasbor

OPS08-BP01 Menganalisis metrik beban kerja

Setelah mengimplementasikan telemetri aplikasi, lakukan analisis terhadap metrik yang dikumpulkan secara rutin. Latensi, permintaan, kesalahan, dan kapasitas (atau kuota) memang memberikan wawasan tentang performa sistem, tetapi memprioritaskan peninjauan terhadap metrik hasil bisnis adalah hal yang sangat penting. Ini akan memastikan Anda mengambil keputusan berbasis data yang selaras dengan tujuan-tujuan bisnis Anda.

Hasil yang diharapkan: Wawasan akurat tentang performa beban kerja yang mendorong keputusan berdasarkan informasi data, sehingga memastikan keselarasan dengan tujuan bisnis.

Anti-pola umum:

- Menganalisis metrik secara terpisah tanpa mempertimbangkan dampak-dampak yang ditimbulkannya terhadap hasil bisnis.
- Ketergantungan berlebihan pada metrik teknis sekaligus mengesampingkan metrik bisnis.
- Peninjauan metrik jarang dilakukan, sehingga peluang pengambilan keputusan waktu nyata terlewatkan.

Manfaat menjalankan praktik terbaik ini:

- Peningkatan pemahaman tentang korelasi antara performa teknis dan hasil bisnis.
- Perbaikan proses pengambilan keputusan yang berlandaskan data waktu nyata.

- Melakukan identifikasi dan mitigasi masalah secara proaktif sebelum hasil bisnis terkena dampaknya.

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

Panduan implementasi

Manfaatkan alat seperti Amazon CloudWatch untuk melakukan analisis metrik. AWS layanan seperti deteksi CloudWatch anomali dan Amazon DevOps Guru dapat digunakan untuk mendeteksi anomali, terutama ketika ambang batas statis tidak diketahui atau ketika pola perilaku lebih cocok untuk deteksi anomali.

Langkah-langkah implementasi

1. Lakukan analisis dan peninjauan: Tinjau dan tafsirkan metrik beban kerja Anda secara rutin.
   a. Memprioritaskan metrik hasil bisnis daripada metrik teknis murni.
   b. Memahami arti penting dari lonjakan, penurunan, atau pola dalam data Anda.
2. Manfaatkan Amazon CloudWatch: Gunakan Amazon CloudWatch untuk tampilan terpusat dan analisis mendalam.
   a. Konfigurasikan CloudWatch dasbor untuk memvisualisasikan metrik Anda dan membandingkannya dari waktu ke waktu.
   b. Gunakan persentil CloudWatch untuk mendapatkan pandangan yang jelas tentang distribusi metrik, yang dapat membantu dalam mendefinisikan SLAs dan memahami outlier.
   c. Siapkan deteksi CloudWatch anomali untuk mengidentifikasi pola yang tidak biasa tanpa bergantung pada ambang batas statis.
   d. Menerapkan observabilitas CloudWatch lintas akun untuk memantau dan memecahkan masalah aplikasi yang menjangkau beberapa akun dalam suatu Wilayah.
   e. Gunakan Wawasan CloudWatch Metrik untuk menanyakan dan menganalisis data metrik di seluruh akun dan Wilayah, mengidentifikasi tren dan anomali.
   f. Terapkan CloudWatch Metric Math untuk mengubah, menggabungkan, atau melakukan perhitungan pada metrik Anda untuk wawasan yang lebih dalam.
3. Mempekerjakan Amazon DevOps Guru: Gabungkan Amazon DevOps Guru untuk deteksi anomali yang ditingkatkan pembelajaran mesin untuk mengidentifikasi tanda-tanda awal masalah operasional untuk aplikasi tanpa server Anda dan memperbaikinya sebelum berdampak pada pelanggan Anda.

4. Lakukan optimalisasi berdasarkan wawasan: Ambil keputusan cerdas berdasarkan analisis metrik Anda untuk menyesuaikan dan meningkatkan beban kerja Anda.

Tingkat upaya untuk Rencana Implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi

Dokumen terkait:

- The Wheel Blog - Menekankan pentingnya peninjauan metrik secara terus-menerus
- Persentil itu penting
- Menggunakan AWS Cost Anomaly Detection
- CloudWatch observabilitas lintas akun
- Kueri metrik Anda dengan Wawasan CloudWatch Metrik

Video terkait:

- Aktifkan Observabilitas Lintas Akun di Amazon CloudWatch
- Pengantar Amazon DevOps Guru
- Terus Menganalisis Metrik menggunakan AWS Cost Anomaly Detection

Contoh terkait:

- Lokakarya Satu Observabilitas
- Mendapatkan wawasan operasi dengan AIOps menggunakan Amazon Guru DevOps

OPS08-BP02 Menganalisis log beban kerja

Melakukan analisis log beban kerja secara rutin merupakan hal yang sangatlah penting untuk mendapatkan pemahaman yang lebih mendalam tentang aspek-aspek operasional aplikasi Anda.

Dengan memilah-milah, memvisualisasikan, dan menafsirkan data log secara efisien, Anda akan dapat terus mengoptimalkan performa dan keamanan aplikasi.

Hasil yang diinginkan: Wawasan yang kaya tentang perilaku dan operasi aplikasi yang berasal dari analisis log yang dilakukan secara menyeluruh, sehingga akan memastikan deteksi dan mitigasi masalah yang proaktif.

Anti-pola umum:

- Mengabaikan analisis log sampai ada masalah kritis yang muncul.
- Tidak menggunakan rangkaian alat lengkap yang tersedia untuk melakukan analisis log, sehingga ada wawasan kritis yang terlewatkan.
- Hanya mengandalkan tinjauan log manual tanpa memanfaatkan kemampuan-kemampuan otomatisasi dan kueri.

Manfaat menjalankan praktik terbaik ini:

- Lakukan identifikasi kemacetan operasional, ancaman keamanan, dan masalah-masalah potensial lain secara proaktif.
- Pemanfaatan data log yang efisien untuk optimalisasi aplikasi yang berkelanjutan.
- Peningkatan pemahaman tentang perilaku aplikasi, sehingga itu akan membantu Anda dalam melakukan upaya debugging dan pemecahan masalah.

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

Panduan implementasi

Amazon CloudWatch Logs adalah alat yang ampuh untuk analisis log. Fitur terintegrasi seperti Wawasan CloudWatch Log dan Wawasan Kontributor membuat proses memperoleh informasi yang bermakna dari log menjadi intuitif dan efisien.

Langkah-langkah implementasi

1. Siapkan CloudWatch Log: Konfigurasikan aplikasi dan layanan untuk mengirim CloudWatch log ke Log.
2. Gunakan deteksi anomali log: Manfaatkan deteksi anomali CloudWatch Amazon Logs untuk secara otomatis mengidentifikasi dan memperingatkan pola log yang tidak biasa. Alat ini akan

membantu Anda secara proaktif mengelola anomali-anomali yang terjadi dalam log Anda dan mendeteksi setiap potensi masalah sejak dini.

3. Siapkan Wawasan CloudWatch Log: Gunakan Wawasan CloudWatch Log untuk mencari dan menganalisis data log Anda secara interaktif.
   a. Buat kueri untuk mengekstrak pola, memvisualisasikan data log, dan memperoleh wawasan yang dapat Anda tindaklanjuti.
   b. Gunakan analisis pola Wawasan CloudWatch Log untuk menganalisis dan memvisualisasikan pola log yang sering. Fitur ini akan membantu Anda memahami tren operasional umum dan setiap potensi penyimpangan yang ada dalam data log Anda.
   c. Gunakan perbandingan CloudWatch Log (diff) untuk melakukan analisis diferensial antara periode waktu yang berbeda atau di seluruh grup log yang berbeda. Gunakan kemampuan ini untuk mendeteksi perubahan-perubahan yang terjadi dan menilai dampaknya terhadap kinerja atau perilaku sistem Anda.
4. Pantau log secara real-time dengan Live Tail: Gunakan Amazon CloudWatch Logs Live Tail untuk melihat data log secara real-time. Anda dapat secara aktif memantau aktivitas operasional aplikasi Anda saat sedang berlangsung, yang memberikan visibilitas langsung kepada Anda mengenai kinerja sistem dan potensi masalah.
5. Manfaatkan Wawasan Kontributor: Gunakan Wawasan CloudWatch Kontributor untuk mengidentifikasi pembicara teratas dalam dimensi kardinalitas tinggi seperti alamat IP atau agen pengguna.
6. Menerapkan filter metrik CloudWatch Log: Konfigurasikan filter metrik CloudWatch Log untuk mengonversi data log menjadi metrik yang dapat ditindaklanjuti. Ini memungkinkan Anda untuk mengatur alarm atau melakukan analisis pola lebih lanjut.
7. Menerapkan observabilitas CloudWatch lintas akun: Pantau dan pecahkan masalah aplikasi yang menjangkau beberapa akun dalam suatu Wilayah.
8. Lakukan peninjauan dan penyempurnaan secara rutin: Tinjau strategi analisis log Anda secara berkala untuk menangkap semua informasi yang relevan dan terus mengoptimalkan performa aplikasi.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS08-BP01 Menganalisis metrik beban kerja

Dokumen terkait:

- Menganalisis Data Log dengan Wawasan CloudWatch Log
- Menggunakan CloudWatch Wawasan Kontributor
- Membuat dan Mengelola Filter Metrik CloudWatch Log

Video terkait:

- Menganalisis Data Log dengan Wawasan CloudWatch Log
- Gunakan Wawasan CloudWatch Kontributor untuk Menganalisis Data Kardinalitas Tinggi

Contoh terkait:

- CloudWatch Pertanyaan Contoh Log
- Lokakarya Satu Observabilitas

OPS08-BP03 Menganalisis jejak beban kerja

Menganalisis data jejak sangatlah penting untuk mencapai pandangan yang komprehensif tentang perjalanan operasional aplikasi. Dengan memvisualisasikan dan memahami interaksi antara berbagai komponen, performa dapat disesuaikan, kemacetan dapat diidentifikasi, dan pengalaman pengguna dapat ditingkatkan.

Hasil yang dinginkan: Dapatkan visibilitas yang jelas tentang operasi terdistribusi yang dimiliki aplikasi Anda, sehingga memungkinkan penyelesaian masalah yang lebih cepat dan pengalaman pengguna yang disempurnakan.

Anti-pola umum:

- Mengabaikan data jejak, dan hanya mengandalkan log serta metrik.
- Tidak melakukan korelasi antara data jejak dengan log terkait.
- Mengabaikan metrik-metrik yang berasal dari jejak, seperti latensi dan tingkat kesalahan.

Manfaat menjalankan praktik terbaik ini:

- Tingkatkan pemecahan masalah dan kurangi waktu rata-rata ke resolusi ()MTTR.
- Mendapatkan wawasan tentang dependensi dan dampaknya.
- Identifikasi dan perbaikan masalah performa secara cepat.
- Memanfaatkan metrik-metrik yang berasal dari jejak untuk pengambilan keputusan yang tepat berdasarkan informasi.
- Pengalaman pengguna yang ditingkatkan melalui interaksi komponen yang dioptimalkan.

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

Panduan implementasi

AWS X-Ray menawarkan serangkaian fitur komprehensif untuk melakukan analisis data jejak, yang dapat menyediakan pandangan yang menyeluruh tentang interaksi layanan, memantau aktivitas pengguna, dan mendeteksi masalah-masalah performa. Fitur seperti ServiceLens, X-Ray Insights, X-Ray Analytics, dan Amazon DevOps Guru meningkatkan kedalaman wawasan yang dapat ditindaklanjuti yang berasal dari data jejak.

Langkah-langkah implementasi

Langkah-langkah berikut menawarkan pendekatan terstruktur untuk menerapkan analisis data jejak secara efektif menggunakan AWS layanan:

1. Integrasikan AWS X-Ray: Pastikan X-Ray terintegrasi dengan aplikasi Anda untuk menangkap data jejak.
2. Analisis metrik X-Ray: Selidiki metrik yang berasal dari jejak X-Ray, seperti latensi, tingkat permintaan, tingkat kesalahan, dan distribusi waktu respons, dengan menggunakan peta layanan untuk memantau kesehatan aplikasi.
3. Gunakan ServiceLens: Manfaatkan ServiceLenspeta untuk meningkatkan observabilitas layanan dan aplikasi Anda. Fitur ini memungkinkan Anda untuk menampilkan jejak, metrik, log, alarm, dan informasi kondisi lainnya secara terpadu.
4. Aktifkan Wawasan X-Ray:
   a. Aktifkan Wawasan X-Ray untuk deteksi anomali otomatis dalam jejak.
   b. Periksa wawasan untuk menentukan pola dan memastikan akar masalah, misalnya peningkatan tingkat kesalahan atau latensi.

   c. Pelajari lini waktu wawasan untuk mendapatkan analisis kronologis dari masalah-masalah yang terdeteksi.
5. Gunakan Analitik X-Ray: Analitik X-Ray akan memungkinkan Anda menjelajahi data jejak secara menyeluruh, menentukan pola, dan mengekstrak wawasan.
6. Gunakan grup di X-Ray: Buat grup di X-Ray untuk memfilter jejak berdasarkan kriteria seperti latensi tinggi, sehingga memungkinkan analisis yang lebih tertarget.
7. Menggabungkan Amazon DevOps Guru: Libatkan Amazon DevOps Guru untuk mendapatkan manfaat dari model pembelajaran mesin yang menunjukkan dengan tepat anomali operasional dalam jejak.
8. Gunakan CloudWatch Synthetics: Gunakan CloudWatchSynthetics untuk membuat kenari untuk terus memantau titik akhir dan alur kerja Anda. Canary ini dapat terintegrasi dengan X-Ray untuk menyediakan data jejak untuk analisis aplikasi yang sedang diuji secara mendalam.
9. Gunakan Real User Monitoring (RUM): Dengan AWS X-Ray dan CloudWatch RUM, Anda dapat menganalisis dan men-debug jalur permintaan mulai dari pengguna akhir aplikasi Anda melalui layanan AWS terkelola hilir. Ini akan membantu Anda untuk mengidentifikasi tren latensi dan kesalahan yang berdampak pada pengguna akhir Anda.
10. Berkorelasi dengan log: Korelasikan data jejak dengan log terkait dalam tampilan jejak X-Ray untuk perspektif mendetail tentang perilaku aplikasi. Ini memungkinkan Anda untuk melihat peristiwa log yang terkait langsung dengan transaksi-transaksi yang dilacak.
11. Menerapkan observabilitas CloudWatch lintas akun: Pantau dan pecahkan masalah aplikasi yang menjangkau beberapa akun dalam suatu Wilayah.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS08-BP01 Menganalisis metrik beban kerja
- OPS08-BP02 Menganalisis log beban kerja

Dokumen terkait:

- Menggunakan ServiceLens untuk Memantau Kesehatan Aplikasi
- Menjelajahi Data Jejak dengan Analitik X-Ray

- Mendeteksi Anomali di dalam Jejak dengan Wawasan X-Ray
- Pemantauan Berkelanjutan dengan CloudWatch Synthetics

Video terkait:

- Analisis dan Debug Aplikasi Menggunakan Amazon CloudWatch Synthetics & AWS X-Ray
- Gunakan Wawasan AWS X-Ray

Contoh terkait:

- Lokakarya Satu Observabilitas
- Menerapkan X-Ray dengan AWS Lambda
- CloudWatchTemplat Canary Synthetics

OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti

Sangat penting untuk mendeteksi dan merespons penyimpangan dalam perilaku aplikasi Anda segera. Lebih penting lagi adalah mengenali ketika hasil yang didasarkan pada indikator kinerja utama (KPI) berisiko atau ketika muncul anomali yang tak terduga. Mendasarkan peringatan pada KPI akan memastikan bahwa sinyal yang Anda terima berkaitan langsung dengan dampak bisnis atau operasional. Pendekatan terhadap peringatan yang dapat ditindaklanjuti ini mempromosikan respons proaktif dan akan membantu Anda untuk mempertahankan performa dan keandalan sistem.

Hasil yang diinginkan: Menerima peringatan yang tepat waktu, relevan, dan dapat ditindaklanjuti untuk identifikasi dan mitigasi potensi masalah dengan cepat, terutama ketika hasil KPI berisiko.

Anti-pola umum:

- Menyiapkan terlalu banyak peringatan non-kritis, yang mengakibatkan kewalahan.
- Tidak memprioritaskan peringatan berdasarkan KPI, sehingga dampak masalah terhadap bisnis menjadi sulit dipahami.
- Mengabaikan penanganan akar masalah, yang berimbas pada munculnya peringatan berulang untuk masalah yang sama.

Manfaat menjalankan praktik terbaik ini:

- Berkurangnya kewalahan akibat peringatan dengan memusatkan perhatian pada peringatan-peringatan yang dapat ditindaklanjuti dan relevan.
- Waktu aktif dan keandalan sistem yang lebih baik melalui deteksi dan mitigasi masalah yang proaktif.
- Kolaborasi tim yang disempurnakan dan penyelesaian masalah yang lebih cepat dengan melakukan integrasi alat-alat peringatan dan komunikasi populer.

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

Panduan implementasi

Untuk membuat sebuah mekanisme peringatan yang efektif, Anda harus menggunakan metrik, log, dan data jejak yang menandai kapan hasil yang didasarkan pada KPI berisiko atau terdapat anomali yang terdeteksi.

Langkah-langkah implementasi

1. Tentukan indikator kinerja utama (KPI): Identifikasi KPI yang dimiliki aplikasi Anda. Peringatan harus dikaitkan dengan KPI ini agar mencerminkan dampak bisnis secara akurat.
2. Implementasikan deteksi anomali:
   - Gunakan deteksi anomali Amazon CloudWatch: Siapkan deteksi anomali Amazon CloudWatch untuk mendeteksi pola yang tidak biasa secara otomatis, yang membantu Anda hanya menghasilkan peringatan untuk anomali yang asli.
   - Gunakan Wawasan AWS X-Ray:
     a. Siapkan Wawasan X-Ray untuk mendeteksi anomali dalam data jejak.
     b. Konfigurasikan notifikasi untuk Wawasan X-Ray agar Anda mendapat notifikasi tentang masalah yang terdeteksi.
   - Integrasikan dengan Amazon DevOps Guru:
     a. Manfaatkan Amazon DevOps Guru untuk kemampuan machine learning-nya dalam mendeteksi anomali operasional pada data yang ada.
     b. Arahkan ke pengaturan notifikasi di DevOps Guru untuk menyiapkan peringatan anomali.
3. Implementasikan peringatan yang dapat ditindaklanjuti: Rancang peringatan yang menyediakan informasi yang memadai untuk tindakan cepat.
   1. Pantau peristiwa AWS Health dengan aturan Amazon EventBridge, atau integrasikan secara terprogram dengan API AWS Health untuk mengotomatiskan tindakan saat Anda menerima

peristiwa AWS Health. Ini bisa berupa tindakan-tindakan umum, seperti mengirimkan semua pesan peristiwa siklus hidup yang direncanakan ke antarmuka obrolan, atau tindakan tertentu, seperti inisiasi alur kerja di alat manajemen layanan IT.

4. Kurangi kelelahan karena peringatan: Minimalkan peringatan non-kritis. Ketika tim kewalahan dengan banyaknya peringatan yang tidak penting, mereka bisa jadi melewatkan masalah-masalah kritis, sehingga mengurangi efektivitas mekanisme peringatan secara keseluruhan.
5. Siapkan alarm komposit: Gunakan alarm komposit Amazon CloudWatch untuk mengkonsolidasikan beberapa alarm.
6. Integrasikan dengan alat peringatan: Gabungkan alat seperti Ops Genie dan PagerDuty.
7. Gunakan kemampuan Amazon Q Developer dalam aplikasi obrolan: Integrasikan Amazon Q Developer dalam aplikasi obrolan untuk menyampaikan peringatan ke Amazon Chime, Microsoft Teams, dan Slack.
8. Peringatan berdasarkan log: Gunakan filter metrik log di CloudWatch untuk membuat alarm berdasarkan peristiwa log tertentu.
9. Tinjau dan ulangi: Tinjau ulang dan sempurnakan konfigurasi peringatan secara rutin.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS04-BP03 Menerapkan telemetri pengalaman pengguna
- OPS04-BP04 Mengimplementasikan telemetri dependensi
- OPS04-BP05 Melaksanakan penelusuran terdistribusi
- OPS08-BP01 Menganalisis metrik beban kerja
- OPS08-BP02 Menganalisis log beban kerja
- OPS08-BP03 Menganalisis jejak beban kerja

Dokumen terkait:

- Menggunakan alarm Amazon CloudWatch

- Membuat sebuah alarm gabungan
- Membuat sebuah alarm CloudWatch berdasarkan pada deteksi anomali
- Notifikasi DevOps Guru
- Notifikasi wawasan X-ray
- Pantau, operasikan, dan pecahkan masalah sumber daya AWS Anda dengan ChatOps interaktif
- Panduan Integrasi Amazon CloudWatch | PagerDuty
- Integrasikan Opsgenie dengan Amazon CloudWatch

Video terkait:

- Membuat Alarm Gabungan di Amazon CloudWatch
- Ikhtisar Amazon Q Developer dalam aplikasi obrolan
- AWS On Air dengan Perintah Mutatif di Amazon Q Developer dalam aplikasi obrolan

Contoh terkait:

- Alarm, manajemen insiden, dan remediasi di cloud dengan Amazon CloudWatch
- Tutorial: Membuat aturan Amazon EventBridge yang mengirimkan notifikasi ke Amazon Q Developer dalam aplikasi obrolan
- Lokakarya Satu Observabilitas

OPS08-BP05 Membuat dasbor

Dasbor adalah tampilan yang berpusat pada manusia tentang data telemetri beban kerja Anda. Meskipun menyediakan antarmuka visual yang vital, dasbor tidak boleh menggantikan mekanisme peringatan, melainkan hanya melengkapinya. Ketika dibuat dengan cermat, dasbor tidak hanya dapat menawarkan wawasan yang disajikan dengan cepat tentang kondisi dan kinerja sistem, tetapi juga dapat menyajikan informasi waktu nyata kepada para pemangku kepentingan tentang hasil bisnis dan dampak dari masalah yang ditimbulkannya.

Hasil yang diinginkan:

Wawasan yang jelas dan dapat ditindaklanjuti tentang kondisi sistem dan bisnis menggunakan representasi visual.

Anti-pola umum:

- Dasbor yang terlalu rumit yang mempunyai terlalu banyak metrik.
- Mengandalkan dasbor tanpa peringatan untuk deteksi anomali.
- Tidak memperbarui dasbor seiring perkembangan beban kerja.

Manfaat praktik terbaik ini:

- Visibilitas langsung tentang metrik sistem penting dan KPI.
- Komunikasi dan pemahaman para pemangku kepentingan yang ditingkatkan.
- Wawasan yang disajikan dengan cepat tentang dampak masalah operasional.

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

Panduan implementasi

Dasbor yang berpusat pada bisnis

Dasbor yang disesuaikan dengan KPI bisnis melibatkan lebih banyak pemangku kepentingan. Meskipun orang-orang ini mungkin tidak tertarik pada metrik sistem, namun mereka tertarik untuk memahami implikasi bisnis dari angka-angka ini. Dasbor yang berpusat pada bisnis memastikan semua metrik teknis dan operasional yang dipantau dan dianalisis selaras dengan tujuan-tujuan bisnis secara keseluruhan. Penyelarasan ini memberikan kejelasan, memastikan semua orang memiliki pemahaman yang sama mengenai hal-hal yang penting dan hal-hal yang tidak penting. Selain itu, dasbor yang menyoroti KPI bisnis cenderung lebih mudah untuk ditindaklanjuti.
Para pemangku kepentingan dapat dengan cepat memahami kondisi operasi, area yang perlu diperhatikan, dan dampak yang mungkin ditimbulkan terhadap hasil bisnis.

Dengan mempertimbangkan hal ini, saat membuat dasbor Anda, pastikan ada keseimbangan antara metrik-metrik teknis dan KPI bisnis. Keduanya penting, tetapi melayani audiens yang berbeda. Idealnya, Anda harus memiliki dasbor yang memberikan pandangan menyeluruh tentang kondisi dan performa sistem sekaligus menekankan hasil bisnis utama serta implikasinya.

Dasbor Amazon CloudWatch adalah halaman beranda yang dapat disesuaikan di konsol CloudWatch yang dapat digunakan untuk memantau sumber daya Anda dalam satu tampilan, bahkan sumber daya yang tersebar di berbagai Wilayah AWS dan akun yang berbeda.

Langkah-langkah implementasi

1. Buat dasbor dasar: Buatlah sebuah dasbor baru di CloudWatch, berikan nama yang deskriptif.

2. Gunakan widget Markdown: Sebelum menggunakan metrik, gunakan widget Markdown untuk menambahkan konteks tekstual di bagian atas dasbor Anda. Widget ini akan menjelaskan cakupan dasbor, tingkat pentingnya metrik yang ditampilkan, dan juga dapat diisi dengan tautan-tautan ke dasbor serta alat-alat pemecahan masalah lainnya.
3. Buat variabel dasbor: Gabungkan variabel dasbor jika sesuai agar dasbor mempunyai tampilan yang dinamis dan fleksibel.
4. Buat widget metrik: Tambahkan widget metrik untuk memberikan visualisasi dari berbagai metrik yang dihasilkan oleh aplikasi Anda, lalu sesuaikan semua widget agar efektif menampilkan kondisi sistem dan hasil bisnis.
5. Kueri Wawasan Log: Manfaatkan Wawasan Log CloudWatch untuk mendapatkan metrik yang dapat ditindaklanjuti dari log Anda dan menampilkan wawasan ini di dasbor Anda.
6. Siapkan alarm: Integrasikan Alarm CloudWatch ke dasbor Anda untuk melihat sekilas metrik apa pun yang melanggar ambang batas mereka.
7. Gunakan Wawasan Kontributor: Gabungkan Wawasan Kontributor CloudWatch untuk menganalisis bidang kardinalitas tinggi dan mendapatkan pemahaman yang lebih jelas tentang kontributor utama sumber daya Anda.
8. Desain widget kustom: Untuk kebutuhan spesifik yang tidak dipenuhi oleh widget standar, sebaiknya Anda membuat widget kustom. Widget kustom ini dapat menarik dari berbagai sumber data atau menyajikan data dengan cara yang unik.
9. Gunakan AWS Health: AWS Health adalah sumber informasi otoritatif tentang kondisi sumber daya AWS Cloud Anda. Gunakan Dasbor AWS Health langsung, atau gunakan data AWS Health di dasbor dan alat Anda sendiri sehingga Anda memiliki informasi tepat yang tersedia untuk membuat keputusan yang tepat.
10. Ulangi dan sempurnakan: Saat aplikasi Anda berkembang, tinjau kembali dasbor Anda secara teratur untuk memastikan relevansinya.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS08-BP01 Menganalisis metrik beban kerja
- OPS08-BP02 Menganalisis log beban kerja
- OPS08-BP03 Menganalisis jejak beban kerja

- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti

Dokumen terkait:

- Membangun Dasbor untuk Visibilitas Operasional
- Menggunakan Dasbor Amazon CloudWatch

Video terkait:

- Membuat Dasbor CloudWatch Lintas Akun & Lintas Wilayah
- AWS re:Invent 2021 - Mendapatkan visibilitas korporasi dengan dasbor operasional AWS Cloud)

Contoh terkait:

- Lokakarya Satu Observabilitas
- Pemantauan Aplikasi dengan Amazon CloudWatch
- Dasbor dan Wawasan Intelijen Peristiwa AWS Health
- Visualisasikan peristiwa AWS Health menggunakan Amazon Managed Grafana

Memahami kesehatan operasional

Tetapkan, rekam, dan analisis metrik operasi untuk mendapatkan visibilitas terhadap aktivitas tim operasi sehingga Anda dapat mengambil tindakan yang tepat.

Organisasi Anda harus mampu memahami kesehatan operasi Anda dengan mudah. Sebaiknya tentukan sasaran bisnis tim operasi Anda, dengan mengidentifikasi indikator performa utama yang mencerminkan sasaran tersebut. Setelah itu, kembangkan dan gunakan metrik berdasarkan hasil operasi untuk mendapatkan wawasan yang berguna. Anda sebaiknya menggunakan metrik-metrik ini untuk mengimplementasikan dasbor dan laporan dengan sudut pandang bisnis dan teknis yang akan membantu para pemimpin dan pemangku kepentingan mengambil keputusan yang matang.

AWS memudahkan penggabungan dan analisis log operasi sehingga Anda dapat menghasilkan metrik, mengetahui status operasi, dan mendapatkan wawasan dari operasi seiring waktu.

Praktik terbaik

- OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik

- OPS09-BP02 Mengomunikasikan status dan tren untuk memastikan visibilitas beroperasi
- OPS09-BP03 Meninjau metrik-metrik operasi dan memprioritaskan perbaikan

OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik

Dapatkan sasaran dan KPI yang menentukan keberhasilan operasi dari organisasi Anda dan pastikan metrik-metrik tersebut mencerminkan hal ini. Tetapkan garis acuan sebagai titik referensi dan lakukan evaluasi ulang secara rutin. Kembangkan mekanisme untuk mengumpulkan metrik-metrik tersebut dari tim untuk dievaluasi. Metrik DevOps Research and Assessment (DORA) menyediakan metode populer untuk mengukur progres penerapan praktik DevOps dalam pengiriman perangkat lunak.

Hasil yang diinginkan:

- Organisasi menerbitkan dan membagikan sasaran dan KPI untuk tim operasi.
- Anda menetapkan metrik-metrik yang mencerminkan KPI ini. Di antara contohnya adalah:
  - Kedalaman antrean tiket atau rata-rata umur tiket
  - Jumlah tiket yang dikelompokkan berdasarkan jenis masalah
  - Waktu yang dihabiskan untuk mengurusi masalah dengan atau tanpa prosedur operasi standar (SOP)
  - Jumlah waktu yang dihabiskan untuk pulih dari push kode yang gagal
  - Volume panggilan

Anti-pola umum:

- Tenggat waktu deployment tidak terpenuhi karena developer disibukkan dengan tugas-tugas pemecahan masalah. Tim pengembangan menuntut lebih banyak personel, tetapi tidak dapat mengukur berapa orang yang mereka butuhkan karena waktu yang tersita tidak dapat diukur.
- Meja Tingkat 1 disiapkan untuk menangani panggilan pengguna. Seiring waktu, makin banyak beban kerja yang ditambahkan, tetapi tidak ada personel yang dialokasikan ke meja Tingkat 1 tersebut. Kepuasan pelanggan sangat rendah karena waktu panggilan semakin meningkat dan masalah berlarut-larut tanpa penyelesaian, tetapi manajemen tidak melihat indikator permasalahan ini, sehingga tidak ada tindakan yang dilakukan.
- Beban kerja yang bermasalah diserahkan kepada tim operasi terpisah untuk dilakukan pemeliharaan. Tidak seperti beban kerja lainnya, beban kerja tersebut tidak dilengkapi dengan

dokumentasi dan runbook yang baik. Akibatnya, tim menghabiskan waktu lebih lama untuk memecahkan masalah dan mengurusi kegagalan. Namun demikian, tidak ada metrik yang mendokumentasikan hal ini, sehingga akuntabilitas menjadi sulit.

Manfaat menjalankan praktik terbaik ini: Ketika pemantauan beban kerja menunjukkan status aplikasi dan layanan kita, tim operasi pemantauan memberi pemilik wawasan tentang perubahan yang terjadi di antara para pemakai beban kerja tersebut, misalnya perubahan-perubahan kebutuhan bisnis. Ukur efektivitas tim-tim tersebut dan evaluasi mereka berdasarkan sasaran bisnis dengan membuat metrik-metrik yang dapat mencerminkan status operasi. Metrik dapat menyoroti masalah dukungan atau mengidentifikasi penyimpangan ketika terjadi pergeseran dari target tingkat layanan.

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

Panduan implementasi

Jadwalkan waktu dengan para pemimpin bisnis dan pemangku kepentingan untuk menentukan apa sasaran layanan secara keseluruhan. Tentukan tugas apa saja yang seharusnya dijalankan oleh berbagai tim operasi dan tantangan apa yang dapat mereka tangani. Dengan menggunakan hal ini, lakukan curah pendapat tentang indikator kinerja utama (KPI) yang mungkin mencerminkan semua sasaran operasi ini. Indikator tersebut mungkin berupa kepuasan pelanggan, waktu dari konsepsi fitur hingga deployment, waktu penyelesaian masalah rata-rata, atau efisiensi biaya.

Berpatokan pada KPI, identifikasi metrik dan sumber data yang mungkin paling mencerminkan semua sasaran ini. Kepuasan pelanggan dapat berupa kombinasi dari berbagai metrik seperti waktu tunggu atau respons panggilan, skor kepuasan, dan jenis-jenis masalah yang disampaikan. Waktu deployment mungkin merupakan jumlah waktu yang diperlukan untuk pengujian dan deployment, serta perbaikan pasca-deployment yang perlu ditambahkan. Statistik yang menunjukkan waktu yang dihabiskan untuk berbagai jenis masalah (atau jumlah masalah tersebut) dapat memberikan wawasan tentang bagian-bagian yang memerlukan upaya tertarget.

Sumber daya

Dokumen terkait:

- Quick - Menggunakan KPI
- Amazon CloudWatch - Menggunakan Metrik
- Membangun Dasbor
- Cara melacak KPI pengoptimalan biaya Anda dengan Dasbor KPI

- Panduan DevOps AWS

Contoh terkait:

- Memantau kinerja pengiriman perangkat lunak Anda menggunakan alat pemantauan dan observabilitas bawaan AWS
- Menyeimbangkan kecepatan dan stabilitas deployment dengan metrik DORA
- Contoh metrik operasional MLOps dalam industri jasa keuangan
- Cara melacak KPI optimalisasi biaya Anda dengan Dasbor KPI

OPS09-BP02 Mengomunikasikan status dan tren untuk memastikan visibilitas beroperasi

Anda perlu mengetahui keadaan operasi Anda dan arah trennya untuk mengidentifikasi kapan hasil mungkin berisiko, apakah pekerjaan tambahan dapat didukung, atau mengidentifikasi efek perubahan terhadap tim Anda. Selama peristiwa operasi, halaman status yang dapat dijadikan acuan oleh para pengguna dan tim operasi untuk mendapatkan informasi dapat mengurangi tekanan pada saluran komunikasi dan menyebarkan informasi secara proaktif.

Hasil yang diinginkan:

- Pimpinan operasi memiliki wawasan sekilas untuk melihat volume panggilan seperti apa yang sedang dioperasikan oleh tim mereka dan upaya apa yang mungkin sedang dilakukan, seperti deployment.
- Peringatan disebarkan kepada para pemangku kepentingan dan komunitas pengguna ketika terjadi dampak terhadap operasi normal.
- Pimpinan dan pemangku kepentingan organisasi dapat memeriksa halaman status sebagai respons terhadap peringatan atau dampak, dan memperoleh informasi seputar peristiwa operasional yang terjadi, seperti titik kontak, informasi tiket, dan perkiraan waktu pemulihan.
- Laporan tersedia bagi para pimpinan dan pemangku kepentingan lainnya untuk menunjukkan statistik operasi seperti volume panggilan selama periode waktu tertentu, skor kepuasan pengguna, jumlah tiket tertunda, dan usia mereka.

Anti-pola umum:

- Terdapat beban kerja yang tidak aktif, sehingga sebuah layanan menjadi tidak tersedia. Volume panggilan melonjak karena para pengguna ingin mengetahui apa yang terjadi. Manajer menambah volume tersebut dengan permintaan informasinya tentang siapa yang mengurusi masalah. Berbagai tim operasi melipatgandakan upaya untuk melakukan penyelidikan.
- Keinginan untuk kemampuan baru menyebabkan beberapa personel dialihkan ke upaya rekayasa. Tidak ada pengisian ulang (backfill) yang disediakan, dan waktu penyelesaian masalah semakin lama. Informasi ini tidak direkam, dan pimpinan baru menyadari hal ini setelah beberapa minggu dan pengguna menyampaikan ketidakpuasan.

Manfaat menjalankan praktik terbaik ini: Selama peristiwa operasional yang berdampak pada bisnis, banyak waktu dan tenaga yang bisa terbuang untuk meminta informasi dari berbagai tim yang sedang berusaha memahami situasinya. Dengan membuat halaman status dan dasbor yang disebarluaskan, para pemangku kepentingan dapat dengan cepat memperoleh informasi mengenai hal-hal seperti apakah ada masalah yang sudah terdeteksi, siapa yang memimpin penanganan masalah tersebut, atau kapan operasi diperkirakan akan kembali normal. Dengan begitu, anggota tim terhindar dari membuang-buang waktu untuk mengomunikasikan status kepada orang lain dan lebih bisa berkonsentrasi untuk menangani masalah.

Selain itu, dasbor dan laporan dapat memberikan wawasan kepada para pembuat keputusan dan pemangku kepentingan untuk melihat bagaimana tim operasi dapat menanggapi kebutuhan bisnis dan bagaimana sumber daya mereka dialokasikan. Hal ini sangat penting untuk menentukan apakah sumber daya yang memadai tersedia untuk mendukung bisnis.

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

Panduan implementasi

Buatlah dasbor yang menunjukkan metrik-metrik utama saat ini untuk tim operasi Anda, dan buat dasbor tersebut mudah diakses oleh para pemimpin operasi serta manajemen.

Buat halaman status yang dapat diperbarui dengan cepat untuk menunjukkan apabila insiden atau peristiwa sedang berlangsung, yang mencantumkan siapa yang bertanggung jawab, dan siapa yang mengoordinasikan respons. Bagikan langkah atau solusi apa pun yang harus dipertimbangkan oleh para pengguna di halaman ini, dan sebarkan luaskan lokasinya. Imbau para pengguna untuk memeriksa lokasi ini terlebih dahulu ketika mereka dihadapkan dengan masalah yang tidak diketahui.

Kumpulkan dan sediakan laporan yang menunjukkan kondisi operasi dari waktu ke waktu, dan distribusikan hal ini kepada para pimpinan dan pengambil keputusan untuk menggambarkan pekerjaan operasi beserta tantangan dan kebutuhan.

Bagikan kepada tim metrik dan laporan yang paling mencerminkan sasaran dan KPI dan bagian yang paling menerima pengaruhnya dalam mendorong perubahan. Luangkan waktu khusus untuk aktivitas ini untuk meningkatkan pentingnya operasi di dalam tim dan antar-tim.

Gunakan AWS Health bersama dasbor Anda sendiri, atau integrasikan peristiwa AWS Health ke dalamnya, sehingga tim Anda dapat mengorelasikan masalah aplikasi dengan status layanan AWS.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik

Dokumen terkait:

- Mengukur Kemajuan
- Membangun dasbor untuk visibilitas operasi

Contoh terkait:

- Operasi Data
- Cara melacak KPI pengoptimalan biaya Anda dengan Dasbor KPI
- Pentingnya Indikator Kinerja Utama (KPI) untuk Migrasi Cloud Skala Besar

OPS09-BP03 Meninjau metrik-metrik operasi dan memprioritaskan perbaikan

Menyisihkan waktu dan sumber daya khusus untuk meninjau keadaan operasi memastikan bahwa pelayanan lini bisnis sehari-hari tetap menjadi prioritas. Kumpulkan para pemimpin operasi dan pemangku kepentingan untuk secara rutin meninjau metrik, menegaskan kembali atau memodifikasi sasaran dan tujuan, dan memprioritaskan perbaikan.

Hasil yang diinginkan:

- Para pemimpin operasi dan staf secara rutin bertemu untuk meninjau metrik selama periode pelaporan tertentu. Tantangan dikomunikasikan, keberhasilan dirayakan, dan pelajaran yang dipetik dibagikan.

- Para pemangku kepentingan dan pemimpin bisnis secara rutin diberi pengarahan tentang keadaan operasi dan diminta untuk memberikan masukan mengenai sasaran, KPI, dan inisiatif masa depan. Kompromi antara pemberian layanan, operasi, dan pemeliharaan dibahas dan dimasukkan ke dalam konteks.

Anti-pola umum:

- Sebuah produk baru diluncurkan, tetapi tim operasi Tingkat 1 dan Tingkat 2 tidak mendapatkan pelatihan yang memadai untuk mendukung atau tidak mendapatkan staf tambahan. Metrik-metrik yang menunjukkan penurunan waktu resolusi tiket dan peningkatan volume insiden tidak terlihat oleh para pimpinan. Tindakan diambil beberapa minggu kemudian ketika jumlah langganan mulai turun karena para pengguna yang tidak puas dan beralih ke platform lain.
- Proses manual untuk melakukan pemeliharaan pada beban kerja telah berlangsung sejak lama. Meskipun sudah ada keinginan untuk melakukan otomatisasi, prioritas yang diberikan rendah mengingat rendahnya nilai penting sistem. Namun seiring waktu, sistem menjadi semakin penting dan sekarang proses manual ini menyita sebagian besar waktu operasional. Tidak ada sumber daya yang dijadwalkan untuk menyediakan peningkatan peralatan untuk operasi, sehingga menyebabkan kelelahan pada staf saat terjadi peningkatan beban kerja. Para pimpinan menyadari hal ini setelah ada laporan bahwa para staf beralih ke kompetitor.

Manfaat menerapkan praktik terbaik ini: Beberapa organisasi mengalami kesulitan untuk mengalokasikan waktu dan perhatian yang sama untuk pemberian layanan dan produk atau penawaran baru. Ketika masalah ini terjadi, lini bisnis dapat mengalami hal buruk karena tingkat layanan yang diharapkan perlahan-lahan memburuk. Alasannya adalah karena operasi tidak berubah dan berkembang sesuai dengan perkembangan bisnis, dan bisa segera tertinggal. Tanpa melakukan peninjauan rutin terhadap wawasan yang dikumpulkan oleh operasi, risiko terhadap bisnis mungkin baru terlihat ketika semua sudah terlambat. Dengan pengalokasian waktu untuk meninjau metrik dan prosedur, baik di antara staf operasi maupun dengan pimpinan, peran penting yang dimiliki oleh operasi akan terus dapat dilihat, dan risiko dapat diidentifikasi jauh sebelum mencapai tingkat kritis. Tim operasi mendapatkan wawasan yang lebih baik tentang perubahan dan inisiatif bisnis yang akan datang, sehingga upaya-upaya proaktif dapat dilakukan. Visibilitas para pimpinan ke dalam metrik-metrik operasi menunjukkan peran penting yang dimiliki oleh tim operasional dalam hal kepuasan pelanggan, baik internal maupun eksternal, dan memungkinkan mereka mempertimbangkan pilihan prioritas dengan lebih baik, atau memastikan bahwa operasional memiliki waktu dan sumber daya untuk berubah dan berkembang seiring munculnya inisiatif bisnis dan beban kerja baru.

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

Panduan implementasi

Luangkan waktu khusus untuk meninjau metrik-metrik operasi antara para pemangku kepentingan dan tim operasional dan meninjau data laporan. Pertimbangkan laporan-laporan berdasarkan tujuan dan sasaran organisasi untuk menentukan apakah semuanya terpenuhi. Identifikasi sumber-sumber ambiguitas yang membuat sasaran menjadi tidak jelas, atau di mana mungkin ada ketidaksesuaian antara apa yang diminta dan apa yang diberikan.

Identifikasi di mana waktu, personel, dan alat dapat membantu mencapai hasil operasi yang diharapkan. Tentukan KPI mana yang akan menerima dampaknya dan target kesuksesan apa yang harus dimiliki. Lakukan peninjauan ulang secara rutin untuk memastikan operasi memiliki sumber daya yang memadai untuk mendukung lini bisnis.

Sumber daya

Dokumen terkait:

- Amazon Athena
- Referensi metrik dan dimensi Amazon CloudWatch
- Amazon Quick
- AWS Glue
- AWS Glue Data Catalog
- Mengumpulkan metrik dan log dari server instans Amazon EC2 dan server on-premise dengan Agen Amazon CloudWatch
- Menggunakan metrik Amazon CloudWatch

Merespons peristiwa

Anda harus mengantisipasi peristiwa operasional, baik yang terencana (seperti promo penjualan, deployment, dan uji kegagalan) dan yang tidak terencana (seperti lonjakan pemanfaatan dan kegagalan komponen). Anda harus menggunakan runbook dan buku pedoman yang Anda miliki untuk menghadirkan hasil yang konsisten ketika merespons pemberitahuan. Pemberitahuan yang ditetapkan harus dimiliki oleh sebuah peran atau tim yang bertanggung jawab atas respons dan eskalasi. Anda juga perlu mengetahui dampak komponen sistem Anda terhadap bisnis dan gunakan

untuk menargetkan upaya saat diperlukan. Anda harus melakukan analisis akar masalah (RCA) setelah peristiwa, lalu mencegah kembali terjadinya kegagalan atau mendokumentasikan pemecahan masalah.

AWS menyederhanakan respons peristiwa Anda dengan menyediakan alat-alat yang mendukung semua aspek beban kerja dan operasi Anda dalam bentuk kode. Alat-alat ini memungkinkan Anda untuk membuat skrip respons terhadap peristiwa operasional dan memulai inisiasi skrip tersebut ketika merespons data pemantauan.

Di AWS, Anda dapat mempercepat waktu pemulihan dengan mengganti komponen yang gagal dengan versi komponen yang diketahui baik, alih-alih mencoba untuk memperbaikinya. Lalu Anda dapat menjalankan analisis terhadap sumber daya yang gagal tersebut di luar jaringan.

Praktik terbaik

- OPS10-BP01 Menggunakan proses untuk manajemen peristiwa, insiden, dan masalah
- OPS10-BP02 Menjalankan proses untuk setiap peringatan
- OPS10-BP03 Memprioritaskan peristiwa operasional berdasarkan dampaknya terhadap bisnis
- OPS10-BP04 Tetapkan jalur eskalasi
- OPS10-BP05 Menentukan rencana komunikasi pelanggan untuk peristiwa yang berdampak pada layanan
- OPS10-BP06 Mengomunikasikan status melalui dasbor
- OPS10-BP07 Otomatiskan tanggapan terhadap acara

OPS10-BP01 Menggunakan proses untuk manajemen peristiwa, insiden, dan masalah

Kemampuan untuk mengelola peristiwa, insiden, dan masalah secara efisien adalah kunci untuk menjaga kondisi kesehatan dan kinerja beban kerja. Sangat penting untuk mengenali dan memahami perbedaan antara elemen-elemen ini untuk mengembangkan sebuah strategi respons dan resolusi yang efektif. Dengan membentuk dan mengikuti proses yang ditentukan dengan baik untuk setiap aspek, tim Anda dapat dengan cepat dan efektif menangani setiap tantangan operasional yang muncul.

Hasil yang diinginkan: Organisasi Anda mengelola peristiwa-peristiwa operasional, insiden, dan masalah secara efektif melalui proses yang terdokumentasi dengan baik dan tersimpan secara terpusat. Proses-proses tersebut diperbarui secara konsisten untuk mencerminkan setiap perubahan,

merampingkan proses penanganan, dan mempertahankan keandalan layanan serta kinerja beban kerja yang tinggi.

Anti-pola umum:

- Anda merespons peristiwa secara reaktif, bukan proaktif.
- Pendekatan-pendekatan yang tidak konsisten diambil untuk berbagai jenis peristiwa atau insiden yang berbeda.
- Organisasi Anda tidak menganalisis dan belajar dari insiden-insiden yang terjadi untuk mencegah kejadian di masa mendatang.

Manfaat menjalankan praktik terbaik ini:

- Proses respons yang efisien dan terstandardisasi.
- Berkurangnya dampak insiden pada layanan dan pelanggan.
- Resolusi masalah yang lebih cepat.
- Perbaikan berkelanjutan dalam proses operasional.

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

Panduan implementasi

Menerapkan praktik terbaik ini berarti Anda melacak peristiwa-peristiwa beban kerja. Anda memiliki proses untuk menangani insiden dan masalah. Proses ini didokumentasikan, dibagikan, dan sering diperbarui. Masalah diidentifikasi, diprioritaskan, dan diperbaiki.

Memahami peristiwa, insiden, dan masalah

- Peristiwa: Sebuah peristiwa adalah sebuah pengamatan atas suatu tindakan, kejadian, atau perubahan status. Peristiwa dapat direncanakan atau tidak direncanakan dan dapat berasal dari dalam atau luar beban kerja.
- Insiden: Insiden adalah peristiwa-peristiwa yang memerlukan respons, seperti gangguan yang tidak terencana atau penurunan kualitas layanan. Insiden-insiden tersebut mewakili gangguan yang membutuhkan perhatian cepat untuk memulihkan operasi beban kerja yang normal.
- Masalah: Masalah adalah penyebab-penyebab yang mendasari satu atau beberapa insiden. Mengidentifikasi dan menyelesaikan masalah mencakup langkah-langkah untuk menyelidiki insiden dengan lebih mendalam untuk mencegah kejadian di masa mendatang.

Langkah-langkah implementasi

Peristiwa

1. Memantau peristiwa:
   - Menerapkan observabilitas dan memanfaatkan observabilitas beban kerja.
   - Tindakan pemantauan yang dilakukan oleh pengguna, peran, atau layanan AWS dicatat sebagai peristiwa di dalam AWS CloudTrail.
   - Respons perubahan operasional di dalam aplikasi Anda dalam waktu nyata dengan Amazon EventBridge.
   - Lakukan penilaian, pemantauan, dan pencatatan perubahan konfigurasi sumber daya secara berkelanjutan dengan AWS Config.
2. Ciptakan proses:
   - Kembangkan sebuah proses untuk menilai peristiwa mana yang signifikan dan memerlukan pemantauan. Langkah ini melibatkan pengaturan ambang batas dan parameter untuk aktivitas normal dan abnormal.
   - Tentukan kriteria eskalasi suatu peristiwa menjadi insiden. Kriteria ini dapat didasarkan pada tingkat keparahan, dampak yang ditimbulkan pada pengguna, atau penyimpangan dari perilaku yang diperkirakan.
   - Lakukan peninjauan terhadap proses pemantauan dan respons peristiwa secara rutin. Langkah ini mencakup analisis insiden masa lalu, penyesuaian ambang batas, dan penyempurnaan mekanisme pembuatan peringatan.

Insiden

1. Merespons insiden:
   - Gunakan wawasan dari alat-alat observabilitas untuk mengidentifikasi dan merespons insiden dengan cepat.
   - Implementasikan AWS Systems Manager Ops Center untuk mengagregasi, mengatur, dan memprioritaskan item dan insiden operasional.
   - Gunakan layanan-layanan seperti Amazon CloudWatch dan AWS X-Ray untuk analisis dan pemecahan masalah yang lebih dalam.
   - Pertimbangkan AWS Managed Services (AMS) untuk meningkatkan manajemen insiden, dengan memanfaatkan kemampuan proaktif, pencegahan, dan detektifnya. AMS akan

memperluas dukungan operasional dengan layanan-layanan seperti pemantauan, deteksi dan respons insiden, dan manajemen keamanan.

- Pelanggan Dukungan Perusahaan dapat menggunakan Deteksi dan Respons Insiden AWS, yang akan menyediakan pemantauan proaktif terus-menerus dan manajemen insiden untuk beban kerja produksi.

2. Buat proses manajemen insiden:
   - Tetapkan sebuah proses manajemen insiden yang terstruktur, termasuk peran yang jelas, protokol komunikasi, dan langkah-langkah penyelesaian masalah.
   - Integrasikan manajemen insiden dengan alat seperti Amazon Q Developer dalam aplikasi obrolan untuk mendapatkan respons dan koordinasi yang efisien.
   - Kategorikan insiden berdasarkan tingkat keparahan, dengan rencana respons insiden yang telah ditentukan sebelumnya untuk masing-masing kategori.
3. Pelajari dan tingkatkan:
   - Lakukan analisis pasca-insiden untuk memahami akar penyebab masalah dan efektivitas penyelesaian masalah.
   - Lakukan pembaruan dan peningkatan secara berkelanjutan terhadap rencana-rencana respons berdasarkan tinjauan dan praktik yang berkembang.
   - Buatlah dokumentasi dari dan bagikan pelajaran yang diperoleh ke seluruh tim untuk meningkatkan ketahanan operasional.
   - Pelanggan Dukungan Perusahaan dapat meminta Lokakarya Manajemen Insiden dari Manajer Akun Teknis mereka. Lokakarya terpandu ini akan menguji rencana respons insiden yang ada sekarang dan akan membantu Anda mengidentifikasi area-area yang perlu ditingkatkan.

Masalah

1. Identifikasi masalah:
   - Gunakan data dari insiden-insiden sebelumnya untuk mengidentifikasi pola-pola yang berulang yang mungkin menandakan adanya masalah sistemik yang lebih mendalam.
   - Manfaatkan alat-alat seperti AWS CloudTrail dan Amazon CloudWatch untuk melakukan analisis tren dan mengungkap masalah-masalah mendasar.
   - Libatkan tim lintas fungsi, termasuk tim operasional, pengembangan, dan unit bisnis, untuk mendapatkan perspektif yang beragam tentang akar penyebab masalah.
2. Buat proses manajemen masalah:

- Kembangkan sebuah proses terstruktur untuk manajemen masalah, dengan fokus pada penyelesaian masalah jangka panjang, bukan perbaikan-perbaikan cepat.
- Sertakan teknik-teknik analisis akar masalah (RCA) untuk menyelidiki dan memahami penyebab dasar terjadinya insiden.
- Perbarui kebijakan operasional, prosedur, dan infrastruktur berdasarkan temuan yang didapatkan untuk mencegah terulangnya kejadian.

3. Terus lakukan perbaikan:
   - Pupuk budaya pembelajaran dan perbaikan yang konstan, dengan mendorong tim untuk mengidentifikasi dan mengatasi setiap potensi masalah secara proaktif.
   - Tinjau dan revisi proses dan alat manajemen masalah agar selaras dengan lanskap bisnis dan teknologi yang berkembang.
   - Bagikan wawasan dan praktik terbaik ke seluruh organisasi untuk membangun sebuah lingkungan operasional yang lebih tangguh dan efisien.
4. Libatkan AWS Dukungan:
   - Gunakan sumber daya AWS dukungan, seperti AWS Trusted Advisor, untuk panduan proaktif dan rekomendasi pengoptimalan.
   - Pelanggan Dukungan Perusahaan dapat mengakses program khusus seperti AWS Countdown untuk mendapatkan dukungan saat terjadi peristiwa kritis.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS04-BP02 Melaksanakan telemetri aplikasi
- OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur
- OPS07-BP04 Menggunakan playbook untuk menyelidiki masalah
- OPS08-BP01 Menganalisis metrik beban kerja
- OPS11-BP02 Lakukan analisis pasca-insiden

Dokumen terkait:

- Panduan Respons Insiden Keamanan AWS
- Deteksi dan Respons Insiden AWS
- Kerangka Kerja Adopsi Cloud AWS: Perspektif Operasional - Manajemen insiden dan masalah
- Manajemen Insiden di Era DevOps dan SRE
- PagerDuty - Apa itu Manajemen Insiden?

Video terkait:

- Kiat respons insiden teratas dari AWS
- AWS re:Invent 2022 - Amazon Builders' Library: 25 tahun keunggulan operasional Amazon
- AWS re:Invent 2022 - Deteksi dan Respons Insiden AWS (SUP201)
- Memperkenalkan Incident Manager dari AWS Systems Manager

Contoh terkait:

- Layanan Proaktif AWS – Lokakarya Manajemen Insiden
- Cara Mengotomatiskan Respons Insiden dengan PagerDuty dan Manajer Insiden AWS Systems Manager
- Libatkan Perespons Insiden dengan Jadwal Personel Siaga di Manajer Insiden AWS Systems Manager
- Tingkatkan Visibilitas dan Kolaborasi selama Penanganan Insiden di Manajer Insiden AWS Systems Manager
- Laporan insiden dan permintaan layanan di AMS

Layanan terkait:

- Amazon EventBridge

OPS10-BP02 Menjalankan proses untuk setiap peringatan

Menetapkan proses yang jelas dan terdefinisi untuk setiap peringatan di dalam sistem Anda sangat penting untuk manajemen insiden yang efektif dan efisien. Praktik ini memastikan bahwa setiap peringatan menghasilkan respons spesifik yang dapat ditindaklanjuti, sehingga meningkatkan keandalan dan responsivitas operasi Anda.

Hasil yang diinginkan: Setiap peringatan memulai rencana respons spesifik dan terdefinisi dengan baik. Jika memungkinkan, respons dilakukan secara otomatis, dengan kepemilikan yang jelas dan jalur eskalasi yang sudah ditentukan. Peringatan ditautkan ke basis pengetahuan yang mutakhir sehingga setiap operator dapat memberikan respons secara konsisten dan efektif. Respons diberikan secara cepat dan seragam, sehingga meningkatkan efisiensi dan keandalan operasional.

Anti-pola umum:

- Peringatan tidak memiliki proses respons yang telah ditentukan sebelumnya, sehingga menyebabkan resolusi yang seadanya dan tertunda.
- Jumlah peringatan yang terlalu banyak dapat menyebabkan terabaikannya peringatan-peringatan penting.
- Peringatan-peringatan ditangani secara tidak konsisten karena tidak adanya kepemilikan dan tanggung jawab yang jelas.

Manfaat menjalankan praktik terbaik ini:

- Mengurangi kewalahan akibat peringatan dengan hanya memunculkan peringatan yang dapat ditindaklanjuti.
- Penurunan rata-rata waktu resolusi (MTTR) untuk masalah operasional.
- Penurunan rata-rata waktu untuk menyelidiki (MTTI), sehingga membantu mengurangi MTTR.
- Peningkatan kemampuan untuk menskalakan respons-respons operasional.
- Peningkatan konsistensi dan keandalan dalam menangani peristiwa-peristiwa operasional.

Misalnya, Anda memiliki proses yang ditentukan untuk peristiwa AWS Health untuk akun-akun penting, termasuk alarm aplikasi, masalah operasional, dan peristiwa siklus hidup terencana (seperti memperbarui versi Amazon EKS sebelum klaster diperbarui secara otomatis), dan Anda memberikan kemampuan bagi tim Anda untuk secara aktif memantau, berkomunikasi, dan merespons peristiwa-peristiwa ini. Tindakan ini membantu Anda mencegah gangguan layanan yang disebabkan oleh perubahan di sisi AWS atau memitigasinya lebih cepat ketika terjadi masalah yang tak terduga.

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

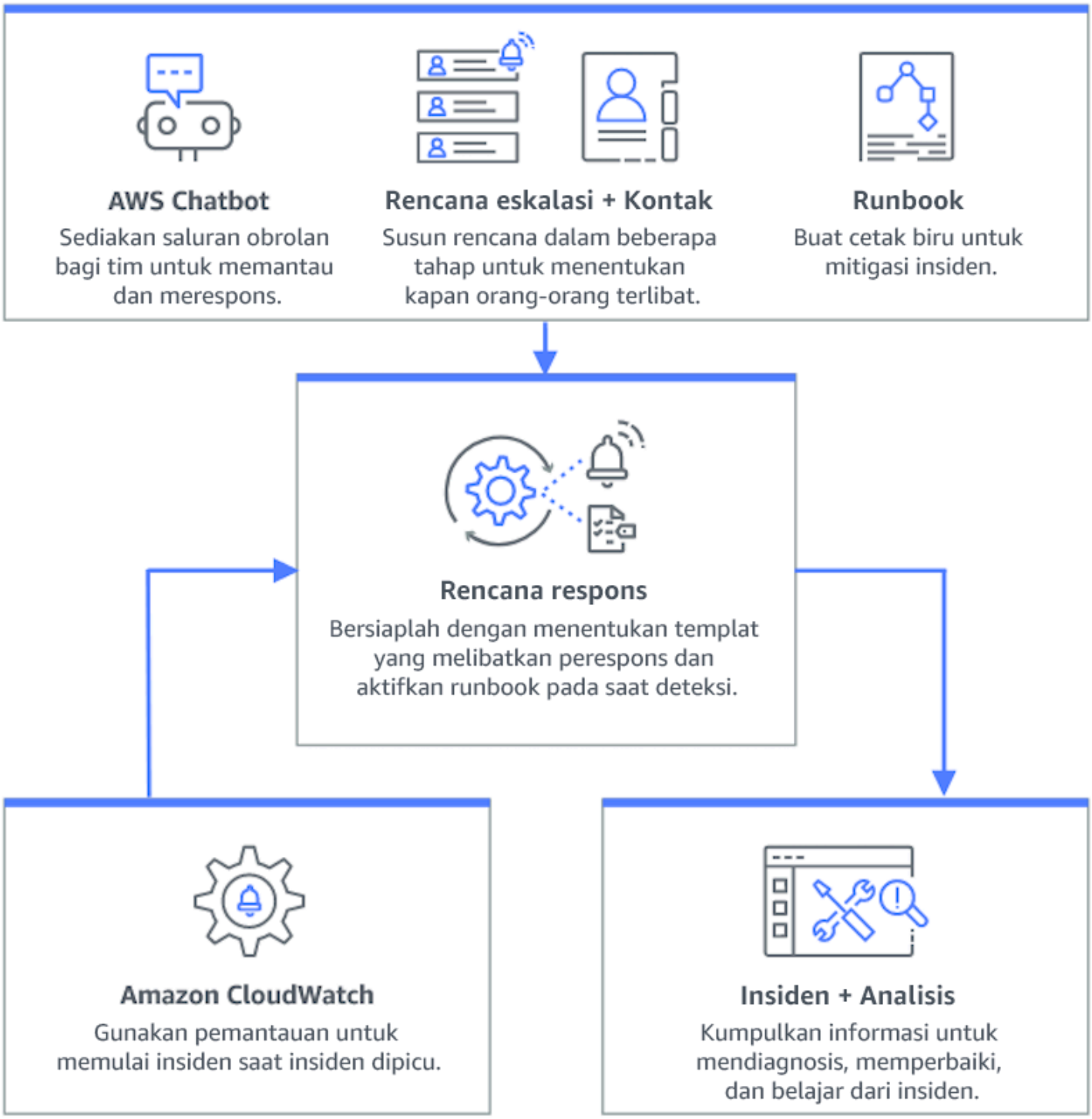
Panduan implementasi

Untuk membuat sebuah proses untuk setiap peringatan, diperlukan pembuatan rencana respons yang jelas untuk setiap peringatan, otomatisasi respons apabila memungkinkan, dan penyempurnaan

proses-proses ini secara berkelanjutan berdasarkan umpan balik operasional dan perubahan persyaratan.

Langkah-langkah implementasi

Diagram berikut ini menggambarkan alur kerja manajemen insiden di dalam Manajer Insiden AWS Systems Manager. Ini dirancang untuk merespons masalah operasional dengan cara menciptakan insiden secara otomatis sebagai respons terhadap peristiwa tertentu dari Amazon CloudWatch atau Amazon EventBridge. Ketika insiden dibuat, baik secara otomatis maupun manual, Manajer Insiden memusatkan manajemen insiden, mengatur informasi sumber daya AWS yang relevan, dan memulai rencana respons yang telah ditentukan sebelumnya. Ini mencakup menjalankan runbook Systems Manager Automation untuk tindakan cepat, serta membuat item kerja operasional induk di OpsCenter untuk melacak tugas dan analisis terkait. Proses yang efisien ini mempercepat dan mengoordinasikan respons insiden di seluruh lingkungan AWS Anda.



1. Gunakan alarm komposit: Buat alarm komposit di CloudWatch untuk mengelompokkan alarm terkait, mengurangi noise, dan memungkinkan respons yang lebih bermakna.
2. Terus dapatkan informasi dengan AWS Health. AWS Health adalah sumber informasi otoritatif tentang kondisi sumber daya AWS Cloud Anda. Gunakan AWS Health untuk memvisualisasikan

dan mendapatkan notifikasi tentang peristiwa layanan saat ini dan perubahan yang akan datang, seperti peristiwa siklus hidup yang direncanakan, sehingga Anda dapat mengambil langkah-langkah untuk mengurangi dampaknya.

a. Buat notifikasi peristiwa AWS Health sesuai keperluan yang dikirim ke saluran email dan obrolan melalui Notifikasi Pengguna AWS serta integrasikan secara programatis dengan alat pemantauan dan peringatan Anda melalui Amazon EventBridge atau API AWS Health.

b. Rencanakan dan lacak progres pada peristiwa kesehatan yang memerlukan tindakan dengan mengintegrasikan dengan manajemen perubahan atau alat ITSM (seperti Jira atau ServiceNow) yang mungkin sudah Anda gunakan melalui Amazon EventBridge atau API AWS Health.

c. Jika Anda menggunakan AWS Organizations, aktifkan tampilan organisasi untuk AWS Health guna menggabungkan peristiwa AWS Health di seluruh akun.

3. Integrasikan alarm Amazon CloudWatch dengan Manajer Insiden: Konfigurasikan alarm CloudWatch untuk membuat insiden secara otomatis di Manajer Insiden AWS Systems Manager.
4. Integrasikan Amazon EventBridge dengan Manajer Insiden: Buat aturan EventBridge untuk bereaksi terhadap peristiwa dan membuat insiden menggunakan rencana respons yang ditentukan.
5. Mempersiapkan insiden di Manajer Insiden:
   - Buat rencana respons terperinci di Manajer Insiden untuk setiap jenis peringatan.
   - Buat saluran obrolan melalui Amazon Q Developer dalam aplikasi obrolan yang terhubung ke rencana respons di Manajer Insiden, sehingga akan memfasilitasi komunikasi waktu nyata selama insiden di seluruh platform seperti Slack, Microsoft Teams, dan Amazon Chime.
   - Menggabungkan runbook Otomatisasi Systems Manager dalam Manajer Insiden untuk mendorong respons otomatis terhadap insiden.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS04-BP01 Identifikasi indikator kinerja utama
- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti

Dokumen terkait:

- Kerangka Kerja Adopsi Cloud AWS: Perspektif Operasional - Manajemen insiden dan masalah
- Menggunakan alarm Amazon CloudWatch

- Menyiapkan Manajer Insiden AWS Systems Manager
- Mempersiapkan insiden di Manajer Insiden

Video terkait:

- Kiat respons insiden teratas dari AWS
- re:Invent 2023 | Mengelola peristiwa siklus hidup sumber daya dalam skala besar dengan AWS Health

Contoh terkait:

- Lokakarya AWS - Manajer Insiden AWS Systems Manager - Mengotomatiskan respons insiden terhadap peristiwa keamanan

OPS10-BP03 Memprioritaskan peristiwa operasional berdasarkan dampaknya terhadap bisnis

Merespons peristiwa operasional dengan cepat adalah hal yang sangat penting, tetapi tidak semua peristiwa sama. Ketika Anda melakukan prioritas berdasarkan dampak bisnis, Anda juga memprioritaskan penanganan peristiwa yang berpotensi menimbulkan konsekuensi signifikan, seperti keamanan, kerugian finansial, pelanggaran peraturan, atau kerusakan reputasi.

Hasil yang diinginkan: Respons terhadap peristiwa operasional diprioritaskan berdasarkan dampak yang mungkin ditimbulkannya terhadap operasi dan tujuan bisnis. Hal ini membuat respons menjadi efisien dan efektif.

Anti-pola umum:

- Setiap peristiwa diperlakukan dengan tingkat urgensi yang sama, sehingga menyebabkan kebingungan dan ketertundaan dalam menangani masalah-masalah kritis.
- Anda gagal membedakan antara peristiwa berdampak tinggi dan rendah, sehingga menyebabkan kesalahan alokasi sumber daya.
- Organisasi Anda tidak memiliki kerangka prioritas yang jelas, sehingga menghasilkan respons-respons yang tidak konsisten terhadap peristiwa-peristiwa operasional.
- Peristiwa diprioritaskan berdasarkan urutan pelaporannya, bukan dampaknya terhadap hasil bisnis.

Manfaat menjalankan praktik terbaik ini:

- Memastikan fungsi-fungsi bisnis penting mendapatkan perhatian terlebih dahulu, sehingga akan meminimalkan potensi kerugian.
- Memperbaiki alokasi sumber daya selama saat terjadi peristiwa secara serentak.
- Meningkatkan kemampuan organisasi untuk mempertahankan kepercayaan dan memenuhi persyaratan-persyaratan berdasarkan peraturan.

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

Panduan implementasi

Ketika dihadapkan dengan beberapa peristiwa operasional, sebuah pendekatan prioritas yang terstruktur berdasarkan dampak dan urgensi sangatlah penting. Pendekatan ini akan membantu Anda dalam mengambil keputusan tepat berdasarkan informasi, mengerahkan upaya pada hal-hal yang paling membutuhkan, dan mengurangi risiko terhadap kelangsungan bisnis.

Langkah-langkah implementasi

1. Lakukan penilaian dampak: Kembangkan sebuah sistem klasifikasi untuk mengevaluasi tingkat keparahan peristiwa dalam hal dampak yang mungkin ditimbulkannya terhadap operasi dan tujuan bisnis. Contoh berikut ini menunjukkan kategori-kategori dampak:

Tingkat dampak	Deskripsi
Tinggi	Memengaruhi banyak staf atau pelanggan, dampak keuangan tinggi, kerusakan reputasi tinggi, atau cedera.
Sedang	Memengaruhi sekelompok staf atau pelanggan, dampak keuangan sedang, atau kerusakan reputasi sedang.
Rendah	Memengaruhi staf atau pelanggan per individu, dampak keuangan rendah, atau kerusakan reputasi rendah.

2. Lakukan penilaian tingkat urgensi: Tentukan tingkat urgensi dalam hal seberapa cepat suatu peristiwa membutuhkan respons, dengan mempertimbangkan faktor-faktor seperti keamanan,

implikasi keuangan, dan perjanjian tingkat layanan (SLA). Contoh berikut ini menunjukkan kategori-kategori urgensi:

Tingkat urgensi	Deskripsi
Tinggi	Kerusakan meningkat secara eksponensial, ada dampak pada pekerjaan sensitif waktu, eskalasi yang tidak terelakkan, atau berdampak pada pengguna atau kelompok VIP.
Sedang	Kerusakan meningkat seiring waktu, atau berdampak pada satu pengguna atau kelompok VIP.
Rendah	Kerusakan kecil meningkat seiring waktu, atau ada dampak pada pekerjaan yang tidak sensitif waktu.

3. Buat matriks prioritas:
   - Gunakan matriks untuk melakukan referensi silang antara dampak dan urgensi, sehingga tingkat prioritas dapat ditetapkan ke berbagai kombinasi.
   - Buat agar matriks tersebut mudah diakses dan dipahami oleh semua anggota tim yang bertanggung jawab untuk memberikan respons atas peristiwa-peristiwa operasional.
   - Contoh matriks berikut ini menampilkan tingkat keparahan insiden berdasarkan urgensi dan dampak:

Urgensi dan dampak	Tinggi	Sedang	Rendah
Tinggi	Kritis	Mendesak	Tinggi
Sedang	Mendesak	Tinggi	Normal
Rendah	Tinggi	Normal	Rendah

4. Latih dan komunikasikan: Latih tim-tim respons tentang matriks prioritas dan pentingnya mengikuti matriks tersebut saat terjadi insiden. Komunikasikan proses penyusunan prioritas kepada semua pemangku kepentingan untuk menetapkan harapan-harapan yang jelas.

5. Integrasikan dengan respons insiden:
    - Sertakan matriks prioritas ke dalam rencana dan alat respons insiden Anda.
    - Lakukan otomatisasi terhadap klasifikasi dan penyusunan prioritas peristiwa jika memungkinkan untuk mempercepat waktu respons.
    - Pelanggan Dukungan Perusahaan dapat memanfaatkan Deteksi dan Respons Insiden AWS, yang menyediakan pemantauan proaktif dalam 24x7 dan manajemen insiden untuk beban kerja produksi.
6. Tinjau dan adaptasi: Lakukan peninjauan secara rutin terhadap efektivitas proses penyusunan prioritas dan lakukan penyesuaian berdasarkan umpan balik dan perubahan dalam lingkungan bisnis.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS03-BP03 Eskalasi didorong
- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti
- OPS09-BP01 Mengukur sasaran operasi dan KPI dengan metrik

Dokumen terkait:

- Atlassian - Memahami tingkat keparahan insiden
- Peta Proses IT - Daftar Periksa Prioritas Insiden

OPS10-BP04 Tetapkan jalur eskalasi

Tetapkan jalur eskalasi yang jelas di dalam protokol respons insiden Anda untuk memfasilitasi tindakan yang tepat waktu dan efektif. Ini mencakup penentuan perintah untuk eskalasi, memberikan detail proses eskalasi, dan memberikan persetujuan tindakan di awal untuk mempercepat pengambilan keputusan dan mengurangi waktu rata-rata resolusi (MTTR).

Hasil yang diinginkan: Proses terstruktur dan efisien yang meneruskan insiden ke personel yang tepat, sehingga waktu respons dan dampak menjadi minimum.

Anti-pola umum:

- Kurangnya kejelasan tentang prosedur pemulihan menyebabkan muncul respons seadanya selama insiden kritis.
- Tidak adanya penentuan izin dan kepemilikan yang mengakibatkan ketertundaan ketika diperlukan tindakan mendesak.
- Para pemangku kepentingan dan pelanggan tidak menerima informasi sesuai dengan harapan.
- Keputusan-keputusan penting tertunda.

Manfaat menjalankan praktik terbaik ini:

- Respons insiden yang efisien melalui prosedur-prosedur eskalasi yang telah ditentukan sebelumnya.
- Mengurangi waktu henti dengan tindakan-tindakan yang telah disetujui sebelumnya dan penanggung jawab yang jelas.
- Alokasi sumber daya yang lebih baik dan penyesuaian tingkat dukungan berdasarkan tingkat keparahan insiden.
- Komunikasi yang lebih baik dengan para pemangku kepentingan dan pelanggan.

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

Panduan implementasi

Jalur eskalasi yang ditentukan dengan benar sangatlah penting untuk memberikan respons insiden yang cepat. Manajer Insiden AWS Systems Manager mendukung penyusunan rencana eskalasi terstruktur dan jadwal personel siaga, yang memberikan peringatan kepada personel yang tepat sehingga mereka siap bertindak ketika insiden terjadi.

Langkah-langkah implementasi

1. Siapkan perintah eskalasi: Siapkan alarm CloudWatch untuk membuat sebuah insiden di Manajer Insiden AWS Systems Manager.
2. Siapkan jadwal panggilan: Buat jadwal panggilan di Incident Manager yang selaras dengan jalur eskalasi Anda. Bekali personel siaga dengan izin dan alat yang diperlukan untuk bertindak cepat.
3. Detail prosedur eskalasi:
   - Tentukan kondisi-kondisi spesifik yang membuat insiden harus dieskalasi.
   - Buat rencana eskalasi di Incident Manager.

- Saluran eskalasi harus terdiri dari suatu kontak atau jadwal personel siaga.
- Tentukan peran dan tanggung jawab tim di setiap tingkat eskalasi.

4. Tindakan mitigasi sebelum persetujuan: Lakukan kerja sama dengan pengambil keputusan untuk menyetujui tindakan di awal untuk skenario yang diantisipasi. Gunakan runbook Systems Manager Automation yang terintegrasi dengan Incident Manager untuk mempercepat resolusi insiden.
5. Tentukan kepemilikan: Identifikasi dengan jelas pemilik internal untuk setiap langkah jalur eskalasi.
6. Sediakan detail eskalasi pihak ketiga:
   - Buatlah dokumentasi perjanjian tingkat layanan (SLA) pihak ketiga, dan selaraskan dengan tujuan-tujuan internal.
   - Tetapkan protokol yang jelas untuk komunikasi vendor selama terjadi insiden.
   - Integrasikan kontak vendor ke dalam alat-alat manajemen insiden sehingga bisa diakses langsung.
   - Lakukan latihan rutin yang menyertakan skenario respons pihak ketiga.
   - Jaga agar informasi eskalasi vendor terdokumentasi dengan baik dan mudah diakses.
7. Latih dan latih rencana eskalasi: Latih tim Anda menjalankan proses eskalasi dan lakukan latihan respons insiden rutin atau hari permainan. Pelanggan Dukungan Perusahaan dapat meminta Lokakarya Manajemen Insiden.
8. Lanjutkan untuk perbaikan: Tinjau efektivitas jalur eskalasi Anda secara rutin. Perbarui proses Anda berdasarkan pelajaran yang dipetik dari insiden yang sudah lewat (post-mortem) dan umpan balik berkelanjutan.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti
- OPS10-BP02 Menjalankan proses untuk setiap peringatan
- OPS11-BP02 Lakukan analisis pasca-insiden

Dokumen terkait:

- Rencana Eskalasi Manajer Insiden AWS Systems Manager

- Bekerja dengan jadwal panggilan di Incident Manager
- Membuat dan Mengelola Runbook
- Manajemen peningkatan akses sementara dengan AWS IAM Identity Center
- Atlassian - Kebijakan eskalasi untuk manajemen insiden yang efektif

OPS10-BP05 Menentukan rencana komunikasi pelanggan untuk peristiwa yang berdampak pada layanan

Komunikasi yang efektif selama peristiwa yang berdampak pada layanan sangat penting untuk menjaga kepercayaan dan transparansi dengan pelanggan. Rencana komunikasi yang terdefinisi dengan baik membantu organisasi Anda berbagi informasi dengan cepat dan jelas, baik secara internal maupun eksternal, selama insiden.

Hasil yang diinginkan:

- Rencana komunikasi yang solid sebagai pedoman yang efektif bagi para pelanggan dan pemangku kepentingan selama peristiwa yang berdampak pada layanan.
- Transparansi dalam komunikasi untuk membangun kepercayaan dan mengurangi kecemasan pelanggan.
- Meminimalkan dampak peristiwa yang berdampak pada layanan terhadap pengalaman pelanggan dan operasional bisnis.

Anti-pola umum:

- Komunikasi yang tidak memadai atau tertunda menyebabkan kebingungan dan ketidakpuasan pada pelanggan.
- Pesan yang terlalu teknis atau tidak jelas akan gagal menyampaikan dampak sebenarnya pada pengguna.
- Tidak ada strategi komunikasi yang telah ditentukan sebelumnya, sehingga menghasilkan pesan yang tidak konsisten dan reaktif.

Manfaat menjalankan praktik terbaik ini:

- Meningkatkan kepercayaan dan kepuasan pelanggan dengan melakukan komunikasi yang proaktif dan jelas.

- Mengurangi beban pada tim dukungan dengan menangani kekhawatiran pelanggan terlebih dahulu.
- Meningkatkan kemampuan untuk mengelola dan memulihkan insiden secara efektif.

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

Panduan implementasi

Pembuatan rencana komunikasi yang komprehensif untuk peristiwa-peristiwa yang berdampak pada layanan melibatkan banyak aspek, mulai dari pemilihan saluran yang tepat hingga penyusunan pesan dan nada pesan. Rencana harus dapat disesuaikan, dapat diskalakan, dan memenuhi skenario pemadaman (outage) yang berbeda-beda.

Langkah-langkah implementasi

1. Menentukan peran dan tanggung jawab:
   - Tugaskan manajer insiden utama untuk mengawasi aktivitas respons insiden.
   - Tunjuk seorang manajer komunikasi yang bertanggung jawab untuk mengoordinasikan semua komunikasi eksternal dan internal.
   - Libatkan manajer dukungan untuk menyediakan komunikasi yang konsisten melalui tiket dukungan.
2. Identifikasi saluran komunikasi: Pilih saluran seperti obrolan di tempat kerja, email, SMS, media sosial, pemberitahuan dalam aplikasi, dan halaman status. Saluran-saluran tersebut harus tangguh dan mampu beroperasi secara independen selama terjadi peristiwa yang berdampak pada layanan.
3. Lakukan komunikasi dengan cepat, jelas, dan rutin kepada pelanggan:
   - Kembangkan templat-templat untuk berbagai skenario gangguan layanan, dengan menekankan kesederhanaan dan detail-detail penting. Sertakan informasi tentang gangguan layanan, waktu penyelesaian yang diharapkan, dan dampak.
   - Gunakan Amazon Pinpoint untuk memberi tahu para pelanggan menggunakan notifikasi push, notifikasi dalam aplikasi, email, pesan teks, pesan suara, dan pesan melalui saluran khusus.
   - Gunakan Amazon Simple Notification Service (Amazon SNS) untuk memberi tahu pelanggan (subscriber) secara terprogram atau melalui email, notifikasi push seluler, dan pesan teks.
   - Komunikasikan status melalui dasbor dengan membagikan dasbor Amazon CloudWatch kepada publik.
   - Dorong keterlibatan media sosial:

    - Pantau media sosial secara aktif untuk memahami sentimen pelanggan.
    - Buat postingan di platform media sosial untuk menyampaikan informasi terbaru kepada publik dan menciptakan keterlibatan komunitas.
    - Siapkan templat untuk melakukan komunikasi media sosial yang konsisten dan jelas.
4. Koordinasikan komunikasi internal: Implementasikan protokol internal menggunakan alat seperti Amazon Q Developer dalam aplikasi obrolan untuk koordinasi dan komunikasi tim. Gunakan dasbor CloudWatch untuk mengkomunikasikan status.
5. Orkestrasi komunikasi dengan alat dan layanan-layanan khusus:
    - Gunakan Manajer Insiden AWS Systems Manager dengan Amazon Q Developer dalam aplikasi obrolan untuk menyiapkan saluran-saluran obrolan khusus untuk komunikasi internal waktu nyata dan koordinasi selama insiden.
    - Gunakan runbook Manajer Insiden AWS Systems Manager untuk mengotomatiskan notifikasi pelanggan melalui Amazon Pinpoint, Amazon SNS, atau alat-alat pihak ketiga seperti platform media sosial selama terjadi insiden.
    - Integrasikan alur kerja persetujuan di dalam runbook untuk meninjau dan mengotorisasi semua komunikasi eksternal secara opsional sebelum dikirim.
6. Latih dan tingkatkan:
    - Lakukan pelatihan tentang penggunaan alat dan strategi komunikasi. Berdayakan tim untuk mengambil keputusan secara tepat waktu selama terjadi insiden.
    - Uji rencana komunikasi dengan menjalankan latihan rutin atau game day. Gunakan pengujian ini untuk menyempurnakan perpesanan dan mengevaluasi efektivitas saluran.
    - Implementasikan mekanisme umpan balik untuk menilai efektivitas komunikasi selama terjadi insiden. Terus kembangkan rencana komunikasi berdasarkan umpan balik dan perubahan kebutuhan.

Tingkat upaya untuk rencana implementasi: Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS07-BP03 Menggunakan runbook untuk menjalankan prosedur
- OPS10-BP06 Mengomunikasikan status melalui dasbor
- OPS11-BP02 Lakukan analisis pasca-insiden

Dokumen terkait:

- Atlassian - Praktik terbaik komunikasi insiden
- Atlassian - Cara menulis pembaruan status yang baik
- PagerDuty - Panduan Komunikasi Insiden

Video terkait:

- Atlassian - Buat rencana komunikasi insiden Anda sendiri: Templat insiden

Contoh terkait:

- Dasbor AWS Health

OPS10-BP06 Mengomunikasikan status melalui dasbor

Gunakan dasbor sebagai alat strategis untuk menyampaikan status operasional waktu nyata dan metrik utama kepada audiens yang berbeda, termasuk tim teknis internal, pimpinan, dan pelanggan. Dasbor ini menawarkan representasi visual tersentralisasi tentang kesehatan sistem dan kinerja bisnis, sehingga meningkatkan transparansi dan efisiensi pengambilan keputusan.

Hasil yang diinginkan:

- Dasbor Anda memberikan gambaran yang komprehensif tentang sistem dan metrik-metrik bisnis yang relevan untuk berbagai pemangku kepentingan.
- Para pemangku kepentingan dapat mengakses informasi operasional secara proaktif, sehingga mengurangi kebutuhan permintaan status yang harus sering kali dilakukan.
- Pengambilan keputusan waktu nyata disempurnakan selama operasi dan insiden normal.

Anti-pola umum:

- Rekayasawan yang bergabung dengan panggilan manajemen insiden mengharuskan adanya pembaruan status untuk mengejar ketertinggalan.
- Mengandalkan pelaporan manual untuk manajemen, yang menyebabkan ketertundaan dan potensi ketidakakuratan.
- Tim operasi sering terganggu dengan permintaan pembaruan status selama terjadi insiden.

Manfaat menjalankan praktik terbaik ini:

- Memberdayakan para pemangku kepentingan dengan akses langsung ke informasi penting, sehingga mendorong pengambilan keputusan tepat yang berdasar informasi.
- Mengurangi inefisiensi operasional dengan meminimalkan pelaporan manual dan permintaan status yang sering dilakukan.
- Meningkatkan transparansi dan kepercayaan melalui visibilitas waktu nyata pada kinerja sistem dan metrik-metrik bisnis.

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

Panduan implementasi

Dasbor menyampaikan status sistem dan metrik-metrik bisnis Anda secara efektif dan dapat disesuaikan dengan kebutuhan kelompok audiens yang berbeda. Alat-alat seperti dasbor Amazon CloudWatch dan Amazon Quick dapat membantu Anda untuk membuat dasbor interaktif waktu nyata untuk pemantauan sistem dan kecerdasan bisnis.

Langkah-langkah implementasi

1. Identifikasi kebutuhan pemangku kepentingan: Tentukan kebutuhan informasi khusus untuk kelompok audiens yang berbeda-beda, seperti tim teknis, pimpinan, dan pelanggan.
2. Pilih alat yang tepat: Pilih alat yang sesuai seperti dasbor Amazon CloudWatch untuk pemantauan sistem dan Amazon Quick untuk kecerdasan bisnis interaktif. AWS Health memberikan pengalaman siap pakai di Dasbor AWS Health, atau Anda dapat menggunakan peristiwa Health di Amazon EventBridge atau melalui API AWS Health untuk melengkapi dasbor Anda sendiri.
3. Rancang dasbor yang efektif:
    - Rancang dasbor yang menyajikan metrik dan KPI yang relevan secara jelas, sehingga dasbor menjadi mudah dimengerti dan dapat ditindaklanjuti.
    - Gabungkan tampilan tingkat sistem dan tingkat bisnis sesuai kebutuhan.
    - Sertakan dasbor tingkat tinggi (untuk gambaran umum) dan dasbor tingkat rendah (untuk analisis mendetail).
    - Integrasikan alarm otomatis di dalam dasbor untuk menyoroti masalah-masalah kritis.
    - Buatlah anotasi dasbor dengan sasaran dan ambang batas metrik-metrik penting untuk visibilitas langsung.
4. Integrasikan sumber data:

- Gunakan Amazon CloudWatch untuk menggabungkan dan menampilkan metrik dari berbagai layanan AWS dan metrik kueri dari sumber data lain, yang akan membuat tampilan terpadu dari metrik kesehatan dan bisnis sistem Anda.
- Gunakan fitur seperti Wawasan Log CloudWatch untuk melakukan kueri dan memvisualisasikan data log dari berbagai aplikasi dan layanan.
- Gunakan peristiwa AWS Health untuk terus mendapatkan informasi tentang status operasional dan masalah operasional yang dikonfirmasi dari layanan AWS melalui API AWS Health atau peristiwa AWS Health di Amazon EventBridge.

5. Berikan akses mandiri:
   - Bagikan dasbor CloudWatch dengan para pemangku kepentingan yang relevan untuk memberikan akses informasi mandiri dengan menggunakan fitur berbagi dasbor.
   - Pastikan dasbor mudah diakses dan menyediakan informasi terkini dalam waktu nyata.
6. Perbarui dan perbaiki secara rutin:
   - Lakukan pembaruan dan penyempurnaan secara terus-menerus pada dasbor agar selaras dengan kebutuhan bisnis yang terus berkembang dan umpan balik yang diberikan para pemangku kepentingan.
   - Tinjau dasbor secara rutin agar tetap relevan dan efektif untuk menyampaikan informasi yang diperlukan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS08-BP05 Membuat dasbor

Dokumen terkait:

- Membangun dasbor untuk visibilitas operasional
- Menggunakan dasbor Amazon CloudWatch
- Membuat dasbor fleksibel dengan variabel dasbor
- Berbagi dasbor CloudWatch
- Metrik kueri dari sumber data lain
- Menambahkan widget kustom ke sebuah dasbor CloudWatch

Contoh terkait:

- Lokakarya Satu Observabilitas - Dasbor

OPS10-BP07 Otomatiskan tanggapan terhadap acara

Mengotomatiskan respons peristiwa sangatlah penting untuk penanganan operasional yang cepat, konsisten, dan bebas kesalahan. Ciptakan proses yang efisien dan gunakan alat untuk mengelola dan merespons peristiwa secara otomatis, sehingga meminimalkan intervensi manual dan meningkatkan efektivitas operasional.

Hasil yang diinginkan:

- Mengurangi kesalahan manusia dan waktu resolusi yang lebih cepat melalui otomatisasi.
- Penanganan peristiwa operasional yang konsisten dan andal.
- Peningkatan efisiensi operasional dan keandalan sistem.

Anti-pola umum:

- Penanganan peristiwa secara manual menyebabkan terjadinya penundaan dan kesalahan.
- Otomatisasi diabaikan dalam tugas-tugas penting yang repetitif.
- Tugas manual yang repetitif menyebabkan kewalahan akibat peringatan dan terlewatkannya masalah-masalah kritis.

Manfaat menjalankan praktik terbaik ini:

- Respons peristiwa yang lebih cepat, sehingga mengurangi waktu henti sistem.
- Operasi yang andal dengan penanganan peristiwa yang otomatis dan konsisten.

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

Panduan implementasi

Integrasikan otomatisasi untuk menciptakan alur kerja operasional yang efisien dan meminimalkan intervensi manual.

Langkah-langkah implementasi

1. Identifikasi peluang otomatisasi: Tentukan tugas-tugas repetitif untuk diotomatisasi, seperti remediasi masalah, pengayaan tiket, manajemen kapasitas, penskalaan, deployment, dan pengujian.
2. Identifikasi perintah-perintah otomatisasi:
   - Menilai dan menentukan kondisi atau metrik tertentu yang memulai respons otomatis menggunakan tindakan CloudWatch alarm Amazon.
   - Gunakan Amazon EventBridge untuk merespons peristiwa dalam AWS layanan, beban kerja khusus, dan aplikasi SaaS.
   - Pertimbangkan peristiwa inisiasi seperti entri log tertentu, ambang metrik kinerja, atau perubahan status sumber daya. AWS
3. Implementasikan otomatisasi berbasis peristiwa:
   - Gunakan runbook AWS Systems Manager Otomasi untuk menyederhanakan tugas pemeliharaan, penerapan, dan remediasi.
   - Membuat insiden di Manajer Insiden secara otomatis mengumpulkan dan menambahkan detail tentang AWS sumber daya yang terlibat ke insiden tersebut.
   - Secara proaktif memantau kuota menggunakan Monitor Kuota untuk AWS.
   - Secara otomatis menyesuaikan kapasitas dengan AWS Auto Scaling untuk menjaga ketersediaan dan kinerja.
   - Otomatiskan jaringan pipa pengembangan dengan Amazon. CodeCatalyst
   - Uji asap atau terus memantau titik akhir dan APIs menggunakan pemantauan sintetis.
4. Lakukan mitigasi risiko melalui otomatisasi:
   - Menerapkan respons keamanan otomatis untuk mengatasi risiko dengan cepat.
   - Gunakan AWS Systems Manager State Manager untuk mengurangi penyimpangan konfigurasi.
   - Memperbaiki sumber daya yang tidak sesuai dengan. Aturan AWS Config

Tingkat upaya untuk rencana implementasi: Tinggi

Sumber daya

Praktik-praktik terbaik terkait:

- OPS08-BP04 Membuat peringatan yang dapat ditindaklanjuti

- OPS10-BP02 Menjalankan proses untuk setiap peringatan

Dokumen terkait:

- Menggunakan runbook Systems Manager Automation dengan Incident Manager
- Membuat insiden di Incident Manager
- AWS kuota layanan
- Pantau penggunaan sumber daya dan kirim notifikasi saat mendekati kuota
- AWS Auto Scaling
- Apa itu Amazon CodeCatalyst?
- Menggunakan CloudWatch alarm Amazon
- Menggunakan tindakan CloudWatch alarm Amazon
- Memediasi Sumber Daya yang Tidak Sesuai dengan Aturan AWS Config
- Membuat metrik dari peristiwa log dengan menggunakan filter
- AWS Systems Manager State Manager

Video terkait:

- Buat Runbook Otomasi dengan AWS Systems Manager
- Cara mengotomatiskan Operasi TI di AWS
- AWS Security Hub CSPM aturan otomatisasi
- Mulai proyek perangkat lunak Anda dengan cepat dengan CodeCatalyst cetak biru Amazon

Contoh terkait:

- Amazon CodeCatalyst Tutorial: Membuat proyek dengan cetak biru aplikasi web tiga tingkat modern
- Lokakarya Satu Observabilitas
- Menanggapi insiden menggunakan Incident Manager

Kembangkan

Belajar, berbagi, dan terus melakukan peningkatan untuk mempertahankan keunggulan operasional. Dedikasikan siklus kerja untuk melakukan perbaikan bertahap yang hampir terus-menerus. Lakukan analisis pasca-insiden tentang semua peristiwa yang memengaruhi pelanggan. Identifikasi faktor kontribusi dan tindakan preventif untuk meminimalkan atau mencegah kemungkinan terjadi lagi. Komunikasikan faktor penyebab kepada komunitas yang terpengaruh jika perlu. Evaluasi secara teratur dan prioritaskan peluang untuk peningkatan (misalnya, permintaan fitur, perbaikan masalah, serta persyaratan kepatuhan), termasuk beban kerja dan prosedur operasi.

Sertakan loop umpan balik dalam prosedur Anda untuk mengidentifikasi dengan cepat area yang perlu ditingkatkan serta menangkap pembelajaran dari operasi yang berjalan.

Bagikan pelajaran yang didapatkan ke seluruh tim. Analisis tren dalam pelajaran yang didapatkan dan lakukan analisis metrik operasi secara retrospektif lintas tim untuk mengidentifikasi peluang dan metode untuk peningkatan. Implementasikan perubahan yang ditujukan untuk menghadirkan peningkatan dan evaluasi hasil untuk menentukan keberhasilan.

Di AWS, Anda dapat mengekspor data log Anda ke Amazon S3 atau mengirim log secara langsung ke Amazon S3 untuk penyimpanan jangka panjang. Dengan AWS Glue, Anda dapat menemukan dan menyiapkan data log Anda di Amazon S3 untuk analitik, dan menyimpan metadata terkait di AWS Glue Data Catalog. Amazon Athena, melalui integrasi native-nya dengan AWS Glue, kemudian dapat digunakan untuk menganalisis data log Anda, yang mengkuerinya menggunakan SQL standar. Menggunakan alat kecerdasan bisnis seperti Amazon Quick, Anda dapat memvisualisasi, menelusuri, dan menganalisis data Anda. Menemukan tren dan peristiwa ketertarikan yang bisa mendorong peningkatan.

Keberhasilan pengembangan operasi terbentuk dari peningkatan kecil yang sering, penyediaan lingkungan yang aman dan waktu untuk bereksperimen, mengembangkan, serta menguji peningkatan, dan lingkungan yang mendukung untuk belajar dari kegagalan. Dukungan operasi untuk lingkungan sandbox, pengembangan, pengujian, dan produksi, dengan meningkatkan tingkat kontrol operasi, memfasilitasi pengembangan dan meningkatkan kemampuan prediksi hasil yang sukses dari perubahan yang diterapkan ke produksi.

Topik

- Belajar, berdiskusi, dan melakukan perbaikan

Belajar, berdiskusi, dan melakukan perbaikan

Menyisihkan waktu secara rutin untuk melakukan analisis aktivitas operasi, analisis kegagalan, bereksperimen, dan melakukan perbaikan, adalah hal yang penting. Ketika terjadi kegagalan, jadikan kegagalan itu sebagai pelajaran untuk tim Anda dan komunitas rekayasawan secara umum. Anda harus menganalisis kegagalan untuk mengambil pelajaran dan merencanakan perbaikan. Anda perlu meninjau hasil pengamatan Anda bersama tim lain untuk memvalidasi wawasan tersebut.

Praktik terbaik

- OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan
- OPS11-BP02 Lakukan analisis pasca-insiden
- OPS11-BP03 Menerapkan loop umpan balik
- OPS11-BP04 Melakukan manajemen pengetahuan
- OPS11-BP05 Menetapkan pendorong untuk perbaikan
- OPS11-BP06 Validasi wawasan
- OPS11-BP07 Lakukan tinjauan metrik operasi
- OPS11-BP08 Mendokumentasikan dan membagikan pelajaran yang didapatkan
- OPS11-BP09 Alokasikan waktu untuk melakukan perbaikan

OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan

Evaluasi beban kerja Anda berdasarkan praktik terbaik arsitektur internal dan eksternal. Lakukan tinjauan beban kerja yang sering dan terencana. Prioritaskan peluang perbaikan ke dalam jadwal pengembangan perangkat lunak Anda.

Hasil yang diinginkan:

- Anda sering menganalisis beban kerja berdasarkan praktik-praktik terbaik arsitektur.
- Anda memberikan peluang perbaikan dengan prioritas yang setara pada fitur-fitur di dalam proses pengembangan perangkat lunak Anda.

Anti-pola umum:

- Anda belum menjalankan peninjauan arsitektur pada beban kerja Anda sejak melakukan deployment beberapa tahun lalu.

- Anda memberikan prioritas yang lebih rendah untuk peluang perbaikan. Dibandingkan dengan fitur-fitur baru, peluang ini tetap berada di backlog.
- Tidak ada standar untuk mengimplementasikan modifikasi terhadap praktik-praktik terbaik untuk organisasi.

Manfaat menjalankan praktik terbaik ini:

- Beban kerja Anda selalu dimutakhirkan dengan praktik terbaik arsitektur.
- Anda mengembangkan beban kerja Anda secara terencana.
- Anda dapat memanfaatkan praktik-praktik terbaik organisasi untuk meningkatkan semua beban kerja.
- Anda menghasilkan keuntungan stabil yang memberikan dampak kumulatif, yang mendorong efisiensi yang lebih menyeluruh.

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

Panduan implementasi

Lakukan tinjauan arsitektur dari beban kerja Anda dalam rentang waktu yang lebih pendek. Gunakan praktik-praktik terbaik internal dan eksternal, evaluasi beban kerja Anda, dan identifikasi peluang perbaikan. Prioritaskan peluang perbaikan ke dalam jadwal pengembangan perangkat lunak Anda.

Langkah-langkah implementasi

1. Lakukan peninjauan arsitektur secara berkala pada beban kerja produksi Anda dengan frekuensi yang sudah disepakati. Gunakan standar arsitektur terdokumentasi yang menyertakan praktik terbaik khusus AWS.
   a. Gunakan standar yang ditetapkan secara internal untuk peninjauan ini. Jika Anda tidak memiliki standar internal, gunakan Kerangka Kerja AWS Well-Architected.
   b. Gunakan AWS Well-Architected Tool untuk membuat lensa kustom praktik-praktik terbaik internal Anda dan lakukan peninjauan arsitektur Anda.
   c. Hubungi Manajer Akun Teknis atau Arsitek Solusi AWS Anda untuk melakukan Peninjauan Kerangka Kerja Well-Architected terpandu pada beban kerja Anda.
2. Prioritaskan peluang perbaikan yang diidentifikasi selama peninjauan ke dalam proses pengembangan perangkat lunak Anda.

Tingkat upaya untuk rencana implementasi: Rendah. Anda dapat menggunakan Kerangka Kerja AWS Well-Architected untuk melakukan peninjauan arsitektur tahunan Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP02 Menjalankan analisis setelah insiden
- OPS11-BP08 Mendokumentasikan dan membagikan pelajaran yang didapatkan
- OPS04 - Mengimplementasikan observabilitas

Dokumen terkait:

- AWS Well-Architected Tool - Lensa Kustom
- Laporan Resmi AWS Well-Architected - Proses peninjauan
- Sesuaikan Ulasan Well-Architected menggunakan Lensa Kustom dan AWS Well-Architected Tool
- Menerapkan siklus hidup Lensa Kustom AWS Well-Architected di organisasi Anda

Video terkait:

- AWS re:Invent 2023 - Menskalakan praktik terbaik AWS Well-Architected di seluruh organisasi Anda

Contoh terkait:

- AWS Well-Architected Tool

OPS11-BP02 Lakukan analisis pasca-insiden

Tinjau peristiwa yang memengaruhi pelanggan dan identifikasi faktor yang berkontribusi serta tindakan pencegahannya. Gunakan informasi ini untuk mengembangkan langkah-langkah mitigasi untuk meminimalkan atau mencegah kemungkinan terjadi lagi. Kembangkan prosedur untuk respons efektif dan cepat. Komunikasikan faktor-faktor yang berkontribusi dan tindakan-tindakan korektif yang diperlukan, yang disesuaikan dengan audiens target.

Hasil yang diinginkan:

- Anda telah menetapkan proses manajemen insiden yang mencakup analisis pasca-insiden.
- Anda menerapkan rencana observabilitas untuk mengumpulkan data tentang peristiwa.
- Dengan data ini, Anda memahami dan mengumpulkan metrik yang mendukung proses analisis pasca-insiden Anda.
- Anda belajar dari insiden untuk meningkatkan hasil di masa depan.

Anti-pola umum:

- Anda mengelola server aplikasi. Kira-kira setiap 23 jam 55 menit, semua sesi aktif Anda dihapus. Anda berupaya mengidentifikasi masalah yang terjadi di server aplikasi Anda. Anda menduga bahwa masalah ini mungkin masalah jaringan, tetapi tidak dapat memperoleh bantuan dari tim jaringan karena mereka terlalu sibuk. Anda tidak menetapkan proses di awal yang dapat Anda jadikan panduan untuk mendapatkan dukungan dan mengumpulkan informasi yang dibutuhkan guna mengetahui masalah yang sedang terjadi.
- Anda mengalami kehilangan data di dalam beban kerja Anda. Hal ini baru pertama kali terjadi dan penyebabnya belum jelas. Anda menganggap bahwa kejadian ini tidak penting karena Anda dapat membuat ulang data. Kehilangan data makin sering terjadi dan memengaruhi pelanggan Anda. Hal ini juga menambah beban operasional Anda karena harus memulihkan data yang hilang.

Manfaat menjalankan praktik terbaik ini:

- Anda memiliki proses yang telah ditetapkan di awal untuk menentukan komponen, kondisi, tindakan, dan peristiwa yang berkontribusi terhadap suatu insiden, yang membantu Anda mengidentifikasi peluang untuk perbaikan.
- Anda menggunakan data dari analisis pasca-insiden untuk melakukan perbaikan.

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

Panduan implementasi

Gunakan sebuah proses untuk menentukan faktor yang berkontribusi. Tinjau semua insiden yang memengaruhi pelanggan. Buatlah sebuah proses untuk mengidentifikasi dan membuat dokumentasi dari faktor-faktor yang berkontribusi terhadap sebuah insiden agar Anda dapat mengembangkan langkah-langkah mitigasi untuk membatasi atau mencegah kejadian serupa serta mengembangkan prosedur untuk merespons dengan cepat dan efektif. Komunikasikan akar masalah

insiden sebagaimana mestinya, dan sesuaikan komunikasi dengan audiens target Anda. Bagikan pembelajaran secara terbuka di dalam organisasi Anda.

Langkah-langkah implementasi

1. Kumpulkan meterik-metrik seperti perubahan deployment, perubahan konfigurasi, waktu mulai insiden, waktu alarm, waktu keterlibatan, waktu mulai mitigasi, dan waktu penyelesaian insiden.
2. Jelaskan titik-titik waktu utama pada kronologi (timeline) untuk memahami peristiwa insiden.
3. Ajukan pertanyaan-pertanyaan berikut:
    a. Apakah Anda dapat mempersingkat waktu deteksi?
    b. Apakah ada pembaruan metrik dan alarm yang dapat mendeteksi insiden lebih dini?
    c. Apakah Anda dapat mempersingkat waktu diagnosis?
    d. Apakah ada pembaruan pada rencana respons atau rencana eskalasi Anda yang melibatkan perespons yang tepat lebih dini?
    e. Apakah Anda dapat mempersingkat waktu mitigasi?
    f. Apakah ada langkah-langkah runbook atau panduan yang dapat Anda tambahkan atau tingkatkan?
    g. Apakah Anda dapat mencegah terjadinya insiden di masa mendatang?
4. Buat daftar periksa dan tindakan. Lacak dan selesaikan semua tindakan.

Tingkat upaya untuk rencana implementasi: Sedang

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP01 Buatlah suatu proses untuk peningkatan berkelanjutan
- OPS4 - Menerapkan observabilitas

Dokumen terkait:

- Menjalankan analisis pasca-insiden di Incident Manager
- Peninjauan Kesiapan Operasional

OPS11-BP03 Menerapkan loop umpan balik

Loop umpan balik menyediakan wawasan yang dapat ditindaklanjuti yang mendorong pengambilan keputusan. Masukkan loop umpan balik ke dalam prosedur dan beban kerja Anda. Ini akan membantu Anda mengidentifikasi permasalahan dan area yang memerlukan perbaikan. Loop umpan balik juga memvalidasi investasi yang dilakukan dalam upaya perbaikan. Loop umpan balik ini adalah landasan untuk meningkatkan beban kerja Anda secara berkelanjutan.

Loop umpan balik terbagi dalam dua kategori: umpan balik langsung dan analisis retrospektif. Umpan balik langsung (immediate feedback) dikumpulkan melalui peninjauan kinerja dan hasil dari aktivitas operasi. Umpan balik ini berasal dari anggota tim, pelanggan, atau output otomatis dari aktivitas. Umpan balik langsung diterima dari hal-hal seperti pengujian A/B dan pengiriman fitur baru, dan ini penting bagi gagal cepat (fail fast).

Analisis retrospektif dilakukan secara rutin untuk menangkap umpan balik dari peninjauan metrik dan hasil operasional dari waktu ke waktu. Retrospektif ini terjadi pada akhir sprint, secara terjadwal, atau setelah perilisan atau peristiwa besar. Tipe loop umpan balik ini memvalidasi investasi dalam operasi atau beban kerja Anda. Loop umpan balik ini akan membantu Anda mengukur keberhasilan dan memvalidasi strategi Anda.

Hasil yang diinginkan: Anda menggunakan umpan balik langsung dan analisis retrospektif untuk mendorong perbaikan. Terdapat sebuah mekanisme untuk mendapatkan umpan balik dari pengguna dan anggota tim. Analisis retrospektif digunakan untuk mengidentifikasi kecenderungan yang mendorong perbaikan.

Anti-pola umum:

- Anda meluncurkan fitur baru tetapi tidak ada cara untuk menerima umpan balik pelanggan tentangnya.
- Setelah berinvestasi dalam perbaikan operasi, Anda tidak melakukan analisis retrospektif untuk memvalidasinya.
- Anda mengumpulkan umpan balik pelanggan tetapi tidak meninjaunya secara rutin.
- Loop umpan balik mendatangkan item-item tindakan yang diajukan tetapi item-item tersebut tidak disertakan dalam proses pengembangan perangkat lunak.
- Pelanggan tidak menerima umpan balik tentang perbaikan yang mereka ajukan.

Manfaat menjalankan praktik terbaik ini:

- Anda dapat bekerja mundur (work backward) dari pelanggan untuk mendorong fitur-fitur baru.
- Budaya organisasi Anda dapat merespons perubahan lebih cepat.
- Tren digunakan untuk mengidentifikasi peluang perbaikan.
- Retrospektif memvalidasi investasi yang dilakukan pada beban kerja dan operasi Anda.

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

Panduan implementasi

Dengan mengimplementasikan praktik-praktik terbaik ini, Anda dapat menggunakan umpan balik langsung dan analisis retrospektif. Loop umpan balik ini mendorong perbaikan. Terdapat banyak mekanisme untuk umpan balik langsung, termasuk survei, jajak pendapat pelanggan, atau formulir umpan balik. Organisasi Anda juga menggunakan retrospektif untuk mengidentifikasi peluang-peluang perbaikan dan memvalidasi inisiatif.

Contoh pelanggan

AnyCompany Retail membuat formulir web di mana pelanggan dapat memberikan umpan balik atau melaporkan masalah. Selama melakukan scrum mingguan, umpan balik pengguna dievaluasi oleh tim pengembangan perangkat lunak. Umpan balik digunakan secara rutin sebagai landasan pengembangan platform mereka. Mereka melakukan analisis retrospektif di akhir setiap sprint untuk mengidentifikasi item yang ingin mereka tingkatkan.

Langkah-langkah implementasi

1. Umpan balik langsung
    - Anda memerlukan sebuah mekanisme untuk menjangkau umpan balik dari pelanggan dan anggota tim. Aktivitas operasi Anda juga dapat dikonfigurasi untuk menghadirkan umpan balik otomatis.
    - Organisasi Anda perlu sebuah proses untuk meninjau umpan balik ini, menentukan hal-hal yang harus ditingkatkan, dan menjadwalkan perbaikan.
    - Umpan balik harus ditambahkan ke dalam proses pengembangan perangkat lunak Anda.
    - Seiring Anda melakukan perbaikan, lakukan tindak lanjut dengan pemberi umpan balik.
        - Anda dapat menggunakan AWS Systems Manager OpsCenteruntuk membuat dan melacak peningkatan ini sebagai OpsItems.
2. Analisis retrospektif

- Lakukan retrospektif di akhir siklus pengembangan, pada jadwal yang ditetapkan, atau setelah perilisan besar.
- Kumpulkan para pemangku kepentingan yang terlibat dalam beban kerja untuk melakukan rapat retrospektif.
- Buat tiga kolom di papan tulis atau lembar kerja: Hentikan, Mulai, dan Pertahankan
  - Stop adalah untuk apa pun yang Anda ingin tidak dilakukan lagi oleh tim Anda.
  - Start adalah gagasan yang ingin mulai Anda lakukan.
  - Keep adalah untuk item-item yang ingin tetap Anda lakukan.
- Keliling ruangan dan kumpulkan umpan balik dari para pemangku kepentingan.
- Buat prioritas umpan balik. Tetapkan tindakan dan pemangku kepentingan untuk item-item Mulai atau Pertahankan.
- Tambahkan tindakan-tindakan ke proses pengembangan perangkat lunak Anda dan sampaikan pembaruan status kepada para pemangku kepentingan seiring Anda melakukan perbaikan.

Tingkat upaya untuk rencana implementasi: Sedang. Untuk mengimplementasikan praktik terbaik ini, Anda memerlukan cara untuk menyerap umpan balik langsung dan menganalisisnya. Selain itu, Anda perlu membangun sebuah proses analisis retrospektif.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP01 Mengevaluasi kebutuhan pelanggan eksternal: Loop umpan balik adalah sebuah mekanisme untuk mengumpulkan kebutuhan pelanggan eksternal.
- OPS01-BP02 Mengevaluasi kebutuhan nasabah internal: Para pemangku kepentingan internal dapat menggunakan loop umpan balik untuk menyampaikan kebutuhan dan persyaratan.
- OPS11-BP02 Lakukan analisis pasca-insiden: Analisis pasca-insiden adalah bentuk analisis retrospektif yang penting yang dilakukan setelah insiden.
- OPS11-BP07 Lakukan tinjauan metrik operasi: Peninjauan metrik operasi mengidentifikasi tren dan area perbaikan.

Dokumen terkait:

- 7 Jebakan yang Harus Dihindari Saat Membangun CCOE
- Playbook Tim Atlassian - Retrospektif

- Definisi Email: Loop Umpan Balik
- Membangun Loop Umpan Balik Berdasarkan Tinjauan Kerangka AWS Well-Architected
- IBMMetodologi Garasi - Pegang retrospektif
- Investopedia — Siklus PDCS
- Memaksimalkan Efektivitas Developer oleh Tim Cochran
- Ulasan Kesiapan Operasi (ORR) Whitepaper - Iterasi
- ITILCSI- Peningkatan Layanan Berkelanjutan
- Saat Toyota bertemu e-commerce: Bersandar pada Amazon

Video terkait:

- Membangun Loop Umpan Balik Pelanggan yang Efektif

Contoh terkait:

- Astuto - Alat umpan balik pelanggan sumber terbuka
- AWS Solusi - Q nABot on AWS
- Fider - Platform untuk mengatur umpan balik pelanggan

Layanan terkait:

- AWS Systems Manager OpsCenter

OPS11-BP04 Melakukan manajemen pengetahuan

Manajemen pengetahuan membantu anggota tim menemukan informasi untuk melakukan pekerjaan mereka. Di dalam organisasi yang mau belajar, informasi dibagikan secara bebas sehingga individu diberdayakan. Informasi dapat ditemukan atau dicari. Informasi bersifat akurat dan mutakhir. Ada mekanisme untuk membuat informasi baru, memperbarui informasi yang sudah ada, dan mengarsipkan informasi yang kedaluwarsa. Contoh paling umum dari platform manajemen pengetahuan adalah suatu sistem manajemen konten seperti wiki.

Hasil yang diinginkan:

- Anggota tim memiliki akses ke informasi yang akurat secara tepat waktu.

- Informasi dapat dicari.
- Ada mekanisme untuk menambahkan, memperbarui, dan mengarsipkan informasi.

Anti-pola umum:

- Tidak ada penyimpanan pengetahuan tersentralisasi. Anggota tim mengelola catatan mereka sendiri di mesin mereka secara lokal.
- Anda memiliki wiki yang di-hosting secara mandiri tetapi tidak ada mekanisme untuk mengelola informasi, yang mengakibatkan informasi menjadi kedaluwarsa.
- Seseorang melihat ada informasi yang kurang tetapi tidak ada proses untuk meminta penambahannya ke tim wiki. Mereka menambahkannya sendiri tetapi mereka melewatkan langkah yang penting, sehingga mengakibatkan terjadinya gangguan (outage).

Manfaat menjalankan praktik terbaik ini:

- Anggota tim diberdayakan karena informasi dibagikan secara bebas.
- Anggota tim baru menjalani masa orientasi dengan lebih cepat karena dokumentasinya mutakhir dan dapat dicari.
- Informasi bersifat tepat waktu, akurat, dan dapat ditindaklanjuti.

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

Panduan implementasi

Manajemen pengetahuan adalah segi penting dari organisasi yang mau belajar. Untuk memulai, Anda memerlukan tempat penyimpanan terpusat guna menyimpan pengetahuan Anda (contoh yang umum yakni wiki yang di-hosting secara mandiri). Anda harus membuat proses untuk menambahkan, memperbarui, dan mengarsipkan pengetahuan. Buatlah standar mengenai apa yang harus didokumentasikan dan izinkan semua orang memberi kontribusi.

Contoh pelanggan

AnyCompany Retail menghosting Wiki internal tempat semua pengetahuan disimpan. Anggota tim didorong untuk menambahkan pengetahuan seiring pengerjaan tugas sehari-hari mereka. Setiap tiga bulan sekali, tim lintas fungsi mengevaluasi halaman mana yang paling jarang diperbarui dan menentukan apakah halaman tersebut harus diarsipkan atau diperbarui.

Langkah-langkah implementasi

1. Mulailah dengan mengidentifikasi sistem manajemen konten tempat pengetahuan akan disimpan. Dapatkan kesepakatan dari para para pemangku kepentingan di seluruh organisasi Anda.
   a. Jika Anda belum memiliki sistem manajemen konten, pertimbangkan untuk menjalankan wiki yang di-hosting secara mandiri atau menggunakan tempat penyimpanan kontrol versi sebagai titik awal.
2. Kembangkan runbook untuk menambahkan, memperbarui, dan mengarsipkan informasi. Didik tim Anda tentang proses-proses ini.
3. Identifikasi pengetahuan apa yang harus disimpan di sistem manajemen konten. Mulailah dengan aktivitas harian (runbook dan playbook) yang dilakukan anggota tim. Bekerja samalah dengan para pemangku kepentingan untuk memprioritaskan pengetahuan yang akan ditambahkan.
4. Secara berkala, bekerja dengan pemangku kepentingan untuk mengidentifikasi out-of-date informasi dan mengarsipkannya atau memperbaruinya.

Tingkat upaya untuk rencana implementasi: Sedang. Jika Anda belum memiliki sistem manajemen konten, Anda dapat membuat wiki yang di-hosting secara mandiri atau menggunakan tempat penyimpanan dokumen dengan pengontrolan versi.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS11-BP08 Mendokumentasikan dan membagikan pelajaran yang didapatkan - Manajemen pengetahuan memfasilitasi pembagian informasi tentang pelajaran yang didapatkan.

Dokumen terkait:

- Atlassian - Manajemen Pengetahuan

Contoh terkait:

- DokuWiki
- Gollum
- MediaWiki
- Wiki.js

OPS11-BP05 Menetapkan pendorong untuk perbaikan

Identifikasi pendorong perbaikan untuk membantu Anda mengevaluasi dan memprioritaskan peluang berdasarkan data dan loop umpan balik. Jelajahi peluang perbaikan di dalam sistem dan proses Anda, dan otomatiskan jika sesuai.

Hasil yang diinginkan:

- Anda melacak data dari seluruh lingkungan Anda.
- Anda mengorelasikan peristiwa dan aktivitas dengan hasil bisnis.
- Anda dapat mencari kesamaan dan perbedaan di antara lingkungan dan sistem.
- Anda memelihara riwayat aktivitas mendetail untuk deployment dan hasil Anda.
- Anda mengumpulkan data untuk mendukung postur keamanan Anda.

Anti-pola umum:

- Anda mengumpulkan data dari seluruh lingkungan Anda tetapi tidak mengorelasikan peristiwa dan aktivitas.
- Anda mengumpulkan data mendetail dari seluruh estate Anda, dan hal tersebut mendorong aktivitas dan biaya Amazon CloudWatch dan AWS CloudTrail yang tinggi. Namun, Anda tidak menggunakan data ini secara bermakna.
- Anda tidak memperhitungkan hasil bisnis ketika menentukan pendorong untuk perbaikan.
- Anda tidak mengukur dampak fitur-fitur baru.

Manfaat menjalankan praktik terbaik ini:

- Anda meminimalkan dampak motivasi berbasis peristiwa atau investasi emosional dengan menentukan kriteria perbaikan.
- Anda merespons peristiwa-peristiwa bisnis, bukan hanya peristiwa teknis.
- Anda mengukur lingkungan Anda untuk mengidentifikasi area-area perbaikan.

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

Panduan implementasi

- Pahami pendorong perbaikan: Anda sebaiknya hanya melakukan perubahan pada suatu sistem ketika hasil-hasil yang diinginkan didukung.
  - Kemampuan yang diinginkan: Lakukan evaluasi terhadap fitur dan kemampuan yang diinginkan saat mengevaluasi peluang-peluang perbaikan.
    - Yang Baru dengan AWS
  - Masalah yang tidak dapat diterima: Lakukan evaluasi pada masalah, bug, dan kerentanan yang tidak dapat diterima saat mengevaluasi peluang-peluang perbaikan. Lacak opsi penyesuaian ukuran, dan cari peluang optimalisasi.
    - Buletin Keamanan Terkini AWS
    - AWS Trusted Advisor
    - Dasbor Inteligensi Cloud
  - Persyaratan kepatuhan: Lakukan evaluasi terhadap pembaruan dan perubahan yang diperlukan untuk mempertahankan kepatuhan Anda terhadap peraturan, kebijakan, atau agar tetap memperoleh dukungan pihak ketiga, saat meninjau peluang untuk perbaikan.
    - Kepatuhan AWS
    - Program Kepatuhan AWS
    - Berita Terbaru Kepatuhan AWS

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01 Prioritas organisasi
- OPS02 Hubungan dan Kepemilikan
- OPS04-BP01 Identifikasikan indikator performa utama
- OPS08 Memanfaatkan observabilitas beban kerja
- OPS09 Memahami Kesehatan Operasional
- OPS11-BP03 Mengimplementasikan loop umpan balik

Dokumen terkait:

- Amazon Athena

- Quick
- Kepatuhan AWS
- Berita Terbaru Kepatuhan AWS
- Program Kepatuhan AWS
- AWS Glue
- Buletin Keamanan Terkini AWS
- AWS Trusted Advisor
- Ekspor data log Anda ke Amazon S3
- Yang Baru dengan AWS
- Keharusan Inovasi yang Berpusat pada Pelanggan
- Transformasi Digital: Kehebohan Sementara atau Kebutuhan Strategis?

Video Terkait

- AWS re:Invent 2023 - Meningkatkan efisiensi operasional dan ketahanan dengan Dukungan (SUP310)

OPS11-BP06 Validasi wawasan

Tinjau respons dan hasil analisis Anda dengan tim lintas fungsi serta pemilik bisnis. Gunakan tinjauan tersebut untuk menetapkan pemahaman umum, mengidentifikasi dampak-dampak tambahan, dan menentukan alur tindakan. Sesuaikan respons sebagaimana mestinya.

Hasil yang diinginkan:

- Anda meninjau wawasan bersama dengan para pemilik bisnis secara rutin. Pemilik bisnis memberikan konteks tambahan untuk wawasan yang baru diperoleh.
- Anda meninjau wawasan dan meminta umpan balik dari rekan-rekan di bidang teknis, dan Anda membagikan pembelajaran Anda ke seluruh tim.
- Anda memublikasikan data dan wawasan untuk ditinjau oleh tim teknis dan bisnis lainnya. Anda memperhitungkan pembelajaran Anda untuk praktik-praktik baru oleh departemen lain.
- Ringkas dan tinjau wawasan baru bersama para pemimpin senior. Pemimpin senior menggunakan wawasan baru untuk menentukan strategi.

Anti-pola umum:

- Anda merilis fitur baru. Fitur ini mengubah beberapa perilaku pelanggan Anda. Observabilitas Anda tidak memperhitungkan perubahan ini. Anda tidak mengukur manfaat perubahan ini.
- Anda mendorong pembaruan baru dan mengabaikan penyegaran AndaCDN. CDNCache tidak lagi kompatibel dengan rilis terbaru. Anda mengukur persentase permintaan dengan kesalahan. Semua pengguna Anda melaporkan HTTP 400 kesalahan saat berkomunikasi dengan server backend. Anda menyelidiki kesalahan klien dan menemukan bahwa waktu Anda terbuang sia-sia karena Anda mengukur dimensi yang salah.
- Perjanjian tingkat layanan Anda menetapkan waktu aktif 99,9%, dan sasaran titik pemulihan Anda adalah empat jam. Pemilik layanan menyatakan bahwa sistem memiliki nol waktu henti (down time). Anda mengimplementasikan solusi replikasi yang mahal dan kompleks, yang menyita banyak waktu dan uang.

Manfaat menjalankan praktik terbaik ini:

- Ketika Anda memvalidasi wawasan bersama para pemilik bisnis dan orang yang ahli di bidangnya, Anda membangun pemahaman yang sama dan memandu perbaikan dengan lebih efektif.
- Anda menemukan masalah tersembunyi dan memperhitungkannya untuk keputusan masa depan.
- Fokus Anda beralih dari hasil teknis ke hasil bisnis.

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

Panduan implementasi

- Validasikan wawasan: Berinteraksi dengan para pemilik bisnis dan orang yang ahli di bidangnya untuk memastikan ada pemahaman dan kesepakatan bersama tentang makna data yang dikumpulkan. Identifikasi masalah-masalah tambahan, dampak potensial, dan tentukan alur tindakan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS01-BP06 Mengevaluasi pengorbanan sambil mengelola manfaat dan risiko
- OPS02-BP06 Tanggung jawab antar tim telah ditentukan sebelumnya atau dinegosiasikan

- OPS11-BP03 Menerapkan loop umpan balik

Dokumen terkait:

- Merancang Cloud Center of Excellence (CCOE)

Video terkait:

- Membangun observabilitas untuk meningkatkan ketahanan

OPS11-BP07 Lakukan tinjauan metrik operasi

Lakukan analisis retrospektif rutin terhadap metrik operasi dengan peserta lintas tim dari berbagai area bisnis. Gunakan tinjauan ini untuk mengidentifikasi peluang perbaikan, potensi pilihan tindakan, dan untuk membagikan pelajaran yang diperoleh. Cari peluang perbaikan di semua lingkungan Anda (misalnya pengembangan, pengujian, dan produksi).

Hasil yang diinginkan:

- Anda sering meninjau metrik yang memengaruhi bisnis
- Anda mendeteksi dan meninjau anomali melalui kemampuan observabilitas Anda
- Anda menggunakan data untuk mendukung hasil dan sasaran bisnis

Anti-pola umum:

- Jendela pemeliharaan Anda mengganggu promosi retail yang signifikan. Bisnis tidak tahu bahwa ada jadwal pemeliharaan standar yang dapat ditunda jika terdapat peristiwa lain yang memengaruhi bisnis.
- Anda mengalami pemadaman berkepanjangan karena Anda umumnya menggunakan pustaka yang sudah usang di organisasi Anda. Sejak saat itu Anda beralih ke pustaka yang didukung. Tim-tim lain yang ada di organisasi Anda tidak tahu bahwa mereka terpapar risiko.
- Anda tidak secara teratur meninjau pencapaian pelangganSLAs. Anda sedang tren untuk tidak bertemu pelanggan SLAs Anda. Ada hukuman finansial yang terkait dengan tidak bertemu pelanggan SLAs Anda.

Manfaat menjalankan praktik terbaik ini:

- Ketika Anda melakukan pertemuan rutin untuk meninjau-metrik metrik operasi, peristiwa, dan insiden, Anda dapat menjaga pemahaman bersama lintas tim.
- Tim Anda bertemu secara rutin untuk meninjau metrik dan insiden, yang memposisikan Anda untuk mengambil tindakan terhadap risiko dan mengenali pelanggan. SLAs
- Anda berbagi pelajaran yang diperoleh, yang menyediakan data untuk penyusunan prioritas dan perbaikan tertarget untuk hasil bisnis.

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

Panduan implementasi

- Lakukan analisis retrospektif rutin terhadap metrik operasi dengan peserta lintas tim dari berbagai area bisnis.
- Libatkan pemangku kepentingan, termasuk tim bisnis, pengembangan, dan operasi, untuk memvalidasi temuan dari umpan balik langsung dan analisis retrospektif, serta untuk membagikan pelajaran yang diperoleh.
- Gunakan wawasan mereka untuk mengidentifikasi peluang perbaikan dan potensi pilihan tindakan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS08-BP05 Buat dasbor
- OPS09-BP03 Meninjau metrik operasi dan memprioritaskan peningkatan
- OPS10-BP01 Gunakan proses untuk manajemen peristiwa, insiden, dan masalah

Dokumen terkait:

- Amazon CloudWatch
- CloudWatch Referensi metrik dan dimensi Amazon
- Menerbitkan metrik kustom
- Menggunakan CloudWatch metrik Amazon
- Dasbor dan visualisasi dengan CloudWatch

OPS11-BP08 Mendokumentasikan dan membagikan pelajaran yang didapatkan

Dokumentasikan dan bagikan pelajaran yang didapatkan dari aktivitas operasional sehingga Anda dapat menggunakannya secara internal dan di seluruh tim. Anda harus membagikan pelajaran yang didapatkan oleh tim Anda guna meningkatkan manfaat di seluruh organisasi Anda. Bagikan informasi dan sumber daya untuk mencegah kesalahan yang dapat dihindari dan memudahkan upaya pengembangan, dan berfokus pada pengiriman fitur-fitur yang diinginkan.

Gunakan AWS Identity and Access Management (IAM) untuk menetapkan izin yang memungkinkan akses terkontrol ke sumber daya yang ingin Anda bagikan di dalam dan antarakun.

Hasil yang diinginkan:

- Anda menggunakan repositori terkontrol versi untuk membagikan pustaka aplikasi, prosedur dalam skrip, dokumentasi prosedur, dan dokumentasi sistem lainnya.
- Anda membagikan standar infrastruktur Anda dalam bentuk templat AWS CloudFormation terkontrol versi.
- Anda meninjau pelajaran yang didapatkan di seluruh tim.

Anti-pola umum:

- Anda mengalami pemadaman berkepanjangan karena organisasi Anda umumnya menggunakan pustaka yang mengandung masalah. Sejak saat itu Anda beralih ke pustaka yang andal. Tim-tim lain di organisasi Anda tidak mengetahui bahwa mereka terpapar risiko. Tidak ada orang yang mendokumentasikan dan membagikan pengalaman dengan pustaka ini, dan mereka tidak menyadari risiko tersebut.
- Anda mengidentifikasi sebuah masalah edge di dalam layanan mikro yang digunakan bersama secara internal yang menyebabkan terganggunya sesi. Anda pun memperbarui panggilan Anda ke layanan guna menghindari masalah edge tersebut. Tim-tim lain yang ada di organisasi Anda tidak tahu bahwa mereka terpapar risiko.
- Anda menemukan cara untuk mengurangi secara signifikan persyaratan pemanfaatan CPU untuk salah satu layanan mikro Anda. Anda tidak tahu bahwa tim lain bisa memanfaatkan teknik ini.

Manfaat menjalankan praktik terbaik ini: Bagikan pelajaran yang didapatkan untuk mendukung perbaikan dan memaksimalkan manfaat pengalaman.

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

Panduan implementasi

- Dokumentasikan dan bagikan pelajaran yang didapatkan Miliki prosedur untuk mendokumentasikan pelajaran didapatkan dari aktivitas operasional dan analisis retrospektif agar dapat digunakan oleh tim lain.
- Bagikan pembelajaran: Miliki prosedur untuk membagikan pelajaran yang didapatkan serta artefak terkait ke seluruh tim. Sebagai contoh, bagikan prosedur, panduan, tata kelola, dan praktik terbaik yang telah diperbarui melalui wiki yang dapat diakses. Bagikan skrip, kode, dan pustaka melalui repositori umum.
  - Manfaatkan AWS re:Post Private sebagai layanan pengetahuan untuk merampingkan kolaborasi dan berbagi pengetahuan dalam organisasi Anda.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS02-BP06 Tanggung jawab antara tim telah dinegosiasikan atau ditetapkan sebelumnya
- OPS05-BP01 Menggunakan kontrol versi
- OPS05-BP06 Membagikan standar desain
- OPS11-BP03 Mengimplementasikan loop umpan balik
- OPS11-BP07 Melakukan peninjauan metrik operasi

Dokumen terkait:

- Tingkatkan kolaborasi dan bagikan pengetahuan cloud secara aman dengan AWS re:Post Private
- Kurangi penundaan proyek dengan solusi docs-as-code

Video terkait:

- AWS re:invent 2023 - Berkolaborasi dalam perusahaan Anda dan dengan AWS menggunakan AWS re:Post Private
- Dukungans You | Menjelajahi Latihan Diskusi Manajemen Insiden

OPS11-BP09 Alokasikan waktu untuk melakukan perbaikan

Dedikasikan waktu dan sumber daya dalam proses Anda untuk memungkinkan peningkatan bertahap yang berkelanjutan.

Hasil yang diinginkan:

- Anda dapat membuat duplikat lingkungan sementara, yang menurunkan risiko, usaha, serta biaya eksperimen dan pengujian.
- Lingkungan duplikat ini dapat digunakan untuk menguji kesimpulan dari analisis dan eksperimen Anda, serta mengembangkan dan menguji peningkatan terencana.
- Anda menjalankan gamedays, dan Anda menggunakan Fault Injection Service (FIS) untuk menyediakan kontrol dan pagar pembatas yang dibutuhkan tim untuk menjalankan eksperimen di lingkungan seperti produksi.

Anti-pola umum:

- Ada masalah performa yang diketahui dalam aplikasi Anda. Ini ditambahkan ke backlog di balik setiap implementasi fitur terencana. Jika peringkat fitur terencana yang ditambahkan tetap konstan, masalah performa tidak akan pernah tertangani.
- Untuk mendukung peningkatan berkelanjutan yang disetujui, administrator dan developer menggunakan seluruh waktu tambahan mereka untuk memilih dan mengimplementasikan peningkatan. Tidak ada peningkatan yang diselesaikan.
- Penerimaan operasional sudah selesai, dan Anda tidak menguji praktik operasional lagi.

Manfaat menerapkan praktik terbaik ini: Dengan mendedikasikan waktu dan sumber daya dalam proses, Anda memungkinkan peningkatan bertahap yang berkelanjutan.

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

Panduan implementasi

- Alokasikan waktu untuk membuat peningkatan: Dedikasikan waktu dan sumber daya dalam proses Anda untuk membuat peningkatan bertahap yang berkelanjutan.
- Implementasikan perubahan guna meningkatkan dan mengevaluasi hasil untuk menentukan keberhasilan.

- Jika hasilnya tidak memenuhi tujuan, dan peningkatan masih menjadi prioritas, lakukan tindakan alternatif.
- Simulasikan beban kerja produksi melalui game day, dan gunakan pembelajaran dari simulasi ini untuk melakukan peningkatan.

Sumber daya

Praktik-praktik terbaik terkait:

- OPS05-BP08 Gunakan beberapa lingkungan

Video terkait:

- AWS RE: invent 2023 - Meningkatkan ketahanan aplikasi dengan Fault Injection Service AWS

Kesimpulan

Keunggulan operasional adalah upaya berkelanjutan dan iteratif.

Raih keberhasilan organisasi Anda dengan adanya tujuan bersama. Pastikan semua orang memahami bagian masing-masing dalam mencapai hasil bisnis serta dampaknya terhadap kemampuan orang lain untuk sukses. Berikan dukungan kepada anggota tim untuk mendukung hasil bisnis Anda.

Setiap kegagalan dan peristiwa operasional harus dipandang sebagai peluang untuk meningkatkan operasi arsitektur Anda. Dengan memahami kebutuhan beban kerja Anda, menentukan runbook untuk aktivitas rutin, dan buku pedoman untuk memandu penyelesaian masalah, menggunakan operasi sebagai fitur kode, dan mempertahankan kesadaran situasional AWS, operasi Anda akan lebih siap dan mampu merespons lebih efektif ketika insiden terjadi.

Dengan berfokus pada peningkatan bertahap berdasarkan prioritas yang berubah-ubah, serta belajar dari respons peristiwa dan analisis retrospektif, Anda akan membantu menyukseskan bisnis dengan meningkatkan efisiensi dan efektivitas aktivitas Anda.

AWS berusaha untuk membantu Anda membangun dan mengoperasikan arsitektur yang memaksimalkan efisiensi saat Anda membangun penerapan yang sangat responsif dan adaptif. Untuk meningkatkan keunggulan operasional beban kerja, Anda harus menerapkan praktik terbaik yang dipaparkan dalam tulisan ini.

Kontributor

- Rich Boyd, Operational Excellence Pillar Lead, Well-Architected, Amazon Web Services
- Jon Steele, Solutions Architect Well-Architected, Amazon Web Services
- Ryan King, Sr. Technical Program Manager, Amazon Web Services
- Chris Kunselman, Advisory Consultant, Amazon Web Services
- Peter Mullen, Advisory Consultant, Amazon Web Services
- Brian Quinn, Sr. Advisory Consultant, Amazon Web Services
- David Stanley, Cloud Operating Model Lead, Amazon Web Services
- Chris Kozlowski, Spesialis Senior Manajer Akun Teknis, Dukungan Perusahaan, Amazon Web Services
- Alex Livingstone, Arsitek Solusi Spesialis Utama, Operasi Cloud, Amazon Web Services
- Paul Moran, Ahli Teknologi Utama, Dukungan Perusahaan, Amazon Web Services
- Peter Mullen, Advisory Consultant, Layanan Profesional, Amazon Web Services
- Chris Pates, Spesialis Senior Manajer Akun Teknis, Dukungan Perusahaan, Amazon Web Services
- Arvind Raghunathan, Spesialis Utama Manajer Akun Teknis, Dukungan Perusahaan, Amazon Web Services
- Fatih (Ben) Mergen, Senior Cost Lead Solutions Architect, Amazon Web Services

Sumber bacaan lebih lanjut

Untuk panduan tambahan, pelajari sumber berikut:

- Kerangka Kerja AWS Well-Architected
- Pusat Arsitektur AWS

Revisi dokumen

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

Perubahan	Deskripsi	Tanggal
Panduan praktik terbaik yang sudah diperbarui	Praktik terbaik diperbarui dengan panduan baru di area berikut: OPS 2, OPS 5, OPS 9, dan OPS 10. Panduan mencakup rekomendasi baru tentang layanan AWS dan AI generatif.	6 November 2024
Panduan praktik terbaik yang sudah diperbarui	Pembaruan praktik terbaik skala besar sudah dilakukan di seluruh pilar. Beberapa konsolidasi konten di OPS 1, OPS 2, dan OPS 3. Perubahan peringkat risiko di OPS 10.	27 Juni 2024
Pembaruan dan konsolidasi konten utama	Konten telah diperbarui dan dikonsolidasikan di beberapa area praktik terbaik. Dua area praktik terbaik (OPS 4 dan OPS 8) telah ditulis ulang dengan konten dan fokus baru. Praktik terbaik telah diperbaru i dan dikonsolidasikan di area berikut: Desain untuk operasi, Mengurangi risiko deploymen t, dan Memahami kesehatan operasional. Area praktik	3 Oktober 2023

	terbaik OPS 04 telah diperbarui untuk Menerapkan observabilitas. Area praktik terbaik OPS 08 telah diperbarui untuk Memanfaatkan observabilitas beban kerja.	
Pembaruan untuk Kerangka Kerja baru	Praktik terbaik diperbarui dengan panduan preskriptif dan praktik terbaik baru ditambahkan.	10 April 2023
Laporan resmi diperbarui	Praktik terbaik sudah diperbarui dengan panduan implementasi yang baru.	15 Desember 2022
Laporan resmi diperbarui	Praktik terbaik diperluas dan rencana pengembangan sudah ditambahkan.	20 Oktober 2022
Pembaruan kecil	Pembaruan redaksi kecil.	8 Agustus 2022
Laporan resmi diperbarui	Pembaruan untuk menggambarkan fitur dan layanan AWS baru, dan praktik terbaik terbaru.	2 Februari 2022
Pembaruan untuk Kerangka Kerja baru	Pembaruan untuk menggambarkan fitur dan layanan AWS baru, dan praktik terbaik terbaru.	8 Juli 2020
Laporan resmi diperbarui	Pembaruan untuk menggambarkan fitur dan layanan AWS baru, dan pembaruan referensi.	1 Juli 2018

Publikasi awal	Pilar Keunggulan Operasional - Kerangka Kerja AWS Well-Architected dipublikasikan	1 November 2017

Pemberitahuan

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

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

AWS Glosarium

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