



Panduan Pengguna

AWS DevOps Agen



AWS DevOps Agen: Panduan Pengguna

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

Tentang AWS DevOps Agen	1
Fitur utama	1
Always-on, respon insiden otonom	1
Mencegah insiden future	2
Dapatkan lebih banyak dari DevOps alat Anda	2
Bagaimana AWS DevOps Agen bekerja	2
Manfaat	3
Apa itu Aplikasi Web DevOps Agen?	3
Konsol	3
Kemampuan aplikasi web	4
Autentikasi	4
Apa itu Ruang DevOps Agen?	5
Bagaimana Ruang Agen diisolasi	5
Aplikasi Web Ruang Agen	6
Kapan menggunakan beberapa Ruang Agen	6
Apa itu DevOps Topologi Agen?	7
Bagaimana grafik topologi dibuat	7
Kemampuan kunci	7
Tampilan topologi	8
Penemuan sumber daya	8
Ruang lingkup investigasi di luar topologi	9
Topologi dan keterampilan Pemahaman Ruang Agen	9
DevOps Keterampilan Agen	9
Apa itu Keterampilan	9
Mengapa menggunakan Keterampilan	10
Bagaimana Keterampilan bekerja	10
Struktur keterampilan	11
Contoh: Keterampilan lengkap	12
Contoh: Keterampilan penyaringan insiden	14
Menciptakan Keterampilan	14
Keterampilan Mengelola	17
Migrasi dari Runbook	19
Keterampilan yang Dipelajari	19
Apa Keterampilan yang Dipelajari?	19

Mengelola Keterampilan yang Dipelajari	21
Instruksi agen	22
Apa itu instruksi agen	22
Mengapa menggunakan instruksi agen	23
Bagaimana instruksi agen bekerja	23
Pelingkupan jenis agen	24
Panduan ukuran konten	24
Contoh	25
Mengatur instruksi agen	25
Mengelola instruksi agen	26
Wilayah yang Didukung	26
Pemantauan sumber daya lintas wilayah	26
Wilayah yang Didukung	27
Titik akhir layanan	27
Pertimbangan-pertimbangan	28
Memulai dengan AWS DevOps Agen	29
Topik:	29
Membuat Ruang Agen	29
Membuat Ruang Agen	29
Memverifikasi pengaturan Ruang Agen Anda	32
Langkah selanjutnya	32
AWS DevOps Panduan orientasi Agen CLI	32
Ikhtisar	32
Prasyarat	33
Pengaturan peran IAM	33
Langkah-langkah orientasi	37
Verifikasi	46
Langkah selanjutnya	32
Catatan	47
Menciptakan lingkungan pengujian	47
Prasyarat	33
Ikhtisar biaya dan keamanan	47
Siapkan AWS akun untuk pengujian	48
Pilih tes Anda	48
Opsi uji A: Uji kapasitas CPU EC2	48
Opsi uji B: Tes tingkat kesalahan Lambda	49

Validasi AWS DevOps Deteksi agen	59
Instruksi pembersihan	61
Pemecahan masalah	61
Uji validasi	62
Memulai AWS DevOps Agen menggunakan AWS CDK	62
Ikhtisar	32
Prasyarat	33
Apa yang dicakup oleh panduan ini	63
Sumber daya dibuat	63
Pengaturan	64
Bagian 1: Menyebarkan ruang agen	65
Bagian 2 (Opsional): Tambahkan pemantauan lintas akun	66
Pemecahan masalah	61
Pembersihan	69
Pertimbangan keamanan	69
Langkah selanjutnya	32
Sumber daya tambahan	70
Memulai dengan AWS DevOps Agen menggunakan AWS CloudFormation	70
Ikhtisar	32
Prasyarat	33
Apa yang dicakup oleh panduan ini	63
Bagian 1: Menyebarkan ruang agen	65
Bagian 2 (Opsional): Tambahkan pemantauan lintas akun	66
Verifikasi	46
Pemecahan masalah	61
Pembersihan	69
Langkah selanjutnya	32
Memulai dengan AWS DevOps Agen menggunakan Terraform	80
Ikhtisar	32
Prasyarat	33
Apa yang dicakup oleh panduan ini	63
Sumber daya dibuat	63
Pengaturan	64
Bagian 1: Menyebarkan ruang agen	65
Bagian 2 (Opsional): Tambahkan pemantauan lintas akun	66
Pemecahan masalah	61

Pembersihan	69
Pertimbangan keamanan	69
Langkah selanjutnya	32
Sumber daya tambahan	70
Bekerja dengan DevOps Agen	89
Bekerja dengan DevOps Agen	89
Respon insiden otonom	89
Tugas sesuai permintaan DevOps	89
Pencegahan insiden proaktif	89
Interface dengan Agen DevOps	90
Respon insiden otonom	90
Memulai Investigasi	90
Triase insiden	92
Minta dukungan manusia	93
Pencegahan insiden proaktif	96
Cara kerja pencegahan insiden proaktif	96
Manfaat	3
Ringkasan agen	97
Mengontrol evaluasi	97
Mengelola rekomendasi	98
Prioritas rekomendasi	98
Agent-ready spesifikasi	100
Menerapkan rekomendasi	100
DevOps Tugas Sesuai Permintaan	101
Kemampuan tugas	101
Mengakses Obrolan	102
Context-aware tanggapan	103
Mengelola percakapan	104
Menghasilkan artefak	104
Mengirim lampiran file	105
Kueri Sampel	107
Mengaktifkan Obrolan di Ruang Agen Anda	110
Interface dengan Agen DevOps	111
DevOps Aplikasi web agen	112
Integrasi Model Context Protocol (MCP)	112
Integrasi Protokol Klien Agen (ACP)	113

Webhook	113
AWS DevOps Agen API	113
Mengkonfigurasi kemampuan untuk AWS DevOps Agen	114
Migrasi dari pratinjau publik ke ketersediaan umum	115
Apa yang berubah	115
Riwayat obrolan sesuai permintaan dari pratinjau publik	115
Kebijakan Terkelola Baru	115
Hubungkan kembali Pusat Identitas IAM (jika ada)	120
Verifikasi	46
Pemecahan masalah	61
AWS Pengaturan akses EKS	123
Prasyarat	33
Pengaturan	64
Pemecahan masalah	61
Menghubungkan Azure	124
Metode pendaftaran	124
Keterbatasan yang Sudah Diketahui	125
Topik	29
Menghubungkan Sumber Daya Azure	125
Menghubungkan Azure DevOps	132
Menghubungkan ke CI/CD jaringan pipa	136
CI/CD Penyedia yang didukung	137
Menghubungkan GitHub	137
Menghubungkan GitLab	142
Menghubungkan Server MCP	144
Persyaratan	144
Pertimbangan keamanan	69
Mendaftarkan server MCP (tingkat akun)	145
Mengkonfigurasi alat MCP di Ruang Agen	149
Mengelola koneksi server MCP	149
Membuat peran IAM untuk otentikasi SiGv4	150
Topik terkait	151
Menghubungkan beberapa AWS Akun	151
Prasyarat	33
Menambahkan AWS akun sekunder	152
Memahami kebijakan yang diperlukan	153

Mengelola akun sekunder	154
Menghubungkan sumber telemetri	154
Integrasi 2 arah bawaan	154
Integrasi 1 arah bawaan	155
Bring-your-own sumber telemetri	156
Menghubungkan Dynatrace	156
Menghubungkan DataDog	160
Menghubungkan Grafana	164
Menghubungkan Relik Baru	168
Menghubungkan Splunk	171
Menghubungkan ke tiket dan obrolan	174
Menghubungkan PagerDuty	174
Menghubungkan ServiceNow	177
Menghubungkan Slack	187
Memanggil DevOps Agen melalui Webhook	189
Prasyarat	33
Jenis webhook	189
Metode otentikasi Webhook	190
Mengkonfigurasi akses webhook	192
Mengelola kredensial webhook	193
Menggunakan webhook	193
Memecahkan masalah webhook	198
Topik terkait	151
Mengintegrasikan AWS DevOps Agen dengan Amazon EventBridge	199
Bagaimana EventBridge rute AWS DevOps Acara agen	199
AWS DevOps Acara agen	200
Membuat pola acara yang cocok AWS DevOps Acara agen	201
EventBridge Izin Amazon	202
EventBridge Sumber daya tambahan	203
AWS DevOps Referensi detail acara agen	203
Log dan Metrik Terjual	210
Metrik yang dijual CloudWatch	210
Prasyarat	33
Log yang dipasok	213
Harga	223
Menghubungkan ke alat yang dihosting secara pribadi	223

Ikhtisar koneksi pribadi	223
Buat koneksi pribadi	226
Gunakan koneksi pribadi dengan penyedia kemampuan	230
Verifikasi koneksi pribadi	233
Hapus koneksi pribadi	234
Pengaturan lanjutan menggunakan sumber daya VPC Lattice yang ada	234
Topik terkait	151
AWS DevOps Agen Keamanan	236
Multi-layered keamanan	236
Agen Spaces	236
Pemrosesan regional dan aliran data	236
Penggunaan Amazon Bedrock dan inferensi lintas wilayah	237
Manajemen identitas dan akses	237
Metode autentikasi	237
Peran IAM	238
Perlindungan data	238
Enkripsi data	238
Penyimpanan dan retensi data	239
Informasi identitas pribadi (PII)	239
Jurnal agen dan pencatatan audit	239
Jurnal agen	239
AWS CloudTrail integrasi	239
Perlindungan injeksi yang cepat	240
Keamanan integrasi	241
Penyedia pendaftaran	242
Konektivitas jaringan	242
Lalu lintas masuk dari AWS DevOps Agen untuk sistem Anda	243
Lalu lintas keluar dari VPC Anda ke AWS DevOps Agen	244
Model tanggung jawab bersama	245
AWS tanggung jawab	245
Tanggung jawab pelanggan	245
Penggunaan data	245
DevOps Izin Agen IAM	246
Tindakan manajemen Ruang Agen	246
Investigasi dan tindakan eksekusi	246
Tindakan manajemen obrolan	246

Topologi dan tindakan penemuan	247
Tindakan pencegahan dan rekomendasi	247
Tindakan manajemen tugas backlog	247
Tindakan manajemen pengetahuan	248
AWS Support tindakan integrasi	248
Tindakan penggunaan dan pemantauan	248
Contoh kebijakan IAM umum	249
Menggunakan peran terkait layanan untuk AWS DevOps Agen	251
AWS Kebijakan terkelola untuk AWS DevOps Agen	252
Membatasi Akses Agen dalam AWS Akun	279
Memahami peran IAM untuk AWS DevOps Agen	279
Memahami pagar pembatas izin	279
Memilih batas sumber daya Anda	282
Membatasi akses layanan	283
Membatasi akses sumber daya	284
Membatasi akses regional	285
Membuat kebijakan IAM khusus	286
Praktik terbaik kebijakan kustom	286
Menyiapkan Autentikasi Pusat Identitas IAM	287
Prasyarat	33
Opsi otentikasi	287
Mengkonfigurasi Pusat Identitas IAM selama pembuatan Ruang Agen	287
Menambahkan Pengguna dan Grup	289
Cara pengguna mengakses aplikasi web Agent Space	290
Mengelola akses pengguna	290
Manajemen sesi	291
Memutuskan Pusat Identitas	291
Menyiapkan Otentikasi Penyedia Identitas Eksternal (IDP)	291
Prasyarat	33
Cara kerjanya	93
Mengkonfigurasi otentikasi iDP eksternal	292
Memperbarui konfigurasi iDP	296
Cara pengguna mengakses aplikasi web Agent Space	290
Manajemen sesi	291
Pertimbangan keamanan	69
Memutuskan sambungan iDP eksternal	299

Pemecahan masalah	61
Enkripsi saat istirahat untuk AWS DevOps Agen	300
Kunci yang dikelola pelanggan	301
AWS DevOps Konteks enkripsi agen	307
Manajemen kunci	308
Memantau kunci enkripsi Anda	309
VPC Endpoint (AWS PrivateLink)	309
Pertimbangan untuk titik akhir AWS DevOps Agen VPC	309
Buat titik akhir antarmuka untuk Agen AWS DevOps	310
Buat kebijakan titik akhir untuk titik akhir antarmuka Anda	311
Validasi kepatuhan untuk Agen AWS DevOps	311
Kuota	313
Meminta peningkatan kuota	314
Riwayat dokumen	315
.....	cccxviii

Tentang AWS DevOps Agen

AWS DevOps Agen adalah agen perbatasan yang menyelesaikan dan secara proaktif mencegah insiden, terus meningkatkan keandalan dan kinerja.

AWS DevOps Agen menyelidiki insiden dan mengidentifikasi perbaikan operasional sebagai insinyur berpengalaman. DevOps

Agen bekerja dengan:

- Pelajari sumber daya Anda dan hubungan mereka.
- Bekerja dengan alat observabilitas, keterampilan, repositori kode, dan saluran pipa Anda. CI/CD
- Mengkorelasikan telemetri, kode, dan data penyebaran untuk memahami hubungan antara sumber daya aplikasi Anda.
- Mendukung aplikasi di lingkungan multicloud dan hybrid.

Fitur utama

AWS DevOps Agen memberikan respon insiden yang komprehensif dan kemampuan pencegahan melalui fitur-fitur berikut:

Always-on, respon insiden otonom

AWS DevOps Agen secara otonom menyelidiki masalah saat mereka terjadi:

- Investigasi insiden otomatis - Mulai menyelidiki segera ketika peringatan atau tiket dukungan masuk
- AWS DevOps Obrolan Agen - Kueri infrastruktur Anda, analisis kesehatan sistem, dan pandu investigasi menggunakan bahasa alami di seluruh aplikasi web DevOps Agent Space. Obrolan memberikan tanggapan sadar konteks berdasarkan halaman yang Anda lihat, apakah menanyakan tentang sumber daya dalam Topologi, mengarahkan penyelidikan, atau memfilter rekomendasi dalam Pencegahan.
- Rencana mitigasi terperinci - Menyediakan tindakan spesifik untuk menyelesaikan insiden, memvalidasi keberhasilan, dan mengembalikan perubahan jika diperlukan
- Koordinasi insiden otomatis — Rutekan pengamatan, temuan, dan langkah-langkah mitigasi melalui saluran komunikasi pilihan Anda seperti Slack dan ServiceNow

- AWS Integrasi Support - Buat kasus AWS Support langsung dari investigasi dengan konteks langsung yang diberikan kepada para ahli AWS Support

Mencegah insiden future

AWS DevOps Agen menganalisis pola di seluruh insiden historis untuk membantu Anda beralih dari pemadam kebakaran reaktif ke peningkatan operasional proaktif:

- Rekomendasi yang ditargetkan - Memberikan peningkatan spesifik dan dapat ditindaklanjuti yang memperkuat empat bidang utama: observabilitas (pemantauan, peringatan, pencatatan), pengoptimalan infrastruktur (penskalaan otomatis, penyetelan kapasitas), dan peningkatan pipa penyebaran (pengujian, validasi).
- Pembelajaran berkelanjutan - Menyempurnakan rekomendasi berdasarkan umpan balik tim Anda

Dapatkan lebih banyak dari DevOps alat Anda

AWS DevOps Agen terintegrasi dengan alat yang ada tanpa mengubah alur kerja Anda:

- Pemetaan sumber daya aplikasi - Membangun grafik topologi sumber daya aplikasi Anda dan hubungannya
- Built-in integrasi - Bekerja dengan alat observabilitas populer (Amazon CloudWatch, Dynatrace, Datadog, New Relic, dan Splunk), repositori kode, dan pipeline (Tindakan dan repositori, alur kerja, dan CI/CD repositori) GitHub GitLab
- Integrasi alat khusus - Memperluas kemampuan dengan menghubungkan ke server Model Context Protocol (MCP) Anda sendiri untuk alat tambahan
- Kueri infrastruktur percakapan — Gunakan bahasa alami untuk menanyakan AWS sumber daya, metrik sistem, dan status alarm tanpa menavigasi beberapa konsol. Obrolan memahami konteks dan memelihara riwayat percakapan untuk pertanyaan tindak lanjut.

Bagaimana AWS DevOps Agen bekerja

AWS DevOps Agen beroperasi melalui arsitektur dual-console. Administrator menggunakan Konsol AWS Manajemen untuk membuat dan mengelola Ruang Agen, mengonfigurasi integrasi, dan mengatur kontrol akses. Tim operasi menggunakan aplikasi web AWS DevOps Agen untuk respons insiden dan aktivitas investigasi sehari-hari. Aplikasi web adalah tempat operator dapat berinteraksi dengan investigasi agen, menelusuri topologi aplikasi lintas akun, dan belajar tentang peningkatan

pencegahan pada observabilitas, kode, saluran pipa, dan arsitektur infrastruktur. Untuk mempelajari selengkapnya, lihat [the section called “Pencegahan insiden proaktif”](#).

Layanan ini diatur di sekitar Agen Spaces, yang merupakan wadah logis yang menentukan apa yang dapat diakses dan diselidiki AWS DevOps Agen. Setiap Ruang Agen berisi konfigurasi AWS akun Anda, integrasi alat pihak ketiga, dan izin akses. Untuk mempelajari selengkapnya, lihat [the section called “Apa itu Ruang DevOps Agen?”](#).

AWS DevOps Agen secara otomatis membangun topologi aplikasi yang memetakan sumber daya Anda dan hubungannya. Topologi ini membantu layanan memahami arsitektur aplikasi Anda selama penyelidikan. Untuk mempelajari selengkapnya, lihat [the section called “Apa itu DevOps Topologi Agen?”](#).

Manfaat

- Reduce mean time to resolution (MTTR) — Investigasi otonom dimulai segera, mempercepat resolusi insiden dari jam ke menit
- Mencegah insiden berulang — Rekomendasi yang ditargetkan mengatasi akar penyebab dan memperkuat ketahanan sistem
- Tingkatkan efisiensi operasional — Bebaskan tim Anda dari tugas investigasi berulang untuk fokus pada inovasi
- Bekerja dalam alur kerja yang ada - Terintegrasi dengan alat dan proses yang ada tanpa gangguan

Apa itu Aplikasi Web DevOps Agen?

AWS DevOps Agen menggunakan arsitektur dual-console yang memisahkan fungsi administratif dari day-to-day kegiatan operasional. Desain ini memungkinkan administrator untuk mengonfigurasi layanan sementara tim operasi fokus pada respons dan pencegahan insiden.

Konsol

AWS DevOps Agen menyediakan dua antarmuka yang berbeda:

- AWS Management Console — Administrator menggunakan AWS Management Console untuk mengatur dan mengelola AWS DevOps Agen. Di konsol ini, Anda dapat [the section called “Membuat Ruang Agen”](#) menghubungkan AWS layanan dan alat pihak ketiga, dan mengelola izin akses untuk organisasi Anda.

- DevOps Aplikasi web agen - Tim operasi menggunakan aplikasi web DevOps Agent Space untuk aktivitas respons insiden harian. Aplikasi mandiri ini menyediakan antarmuka di mana insinyur panggilan dapat meluncurkan investigasi, berinteraksi dengan agen melalui obrolan bahasa alami, melihat topologi aplikasi, dan meninjau rekomendasi pencegahan insiden.

Kemampuan aplikasi web

Aplikasi web DevOps Agen menyediakan kemampuan utama berikut:

- Respons Insiden — Halaman ini adalah tempat Anda membuat dan melacak investigasi insiden serta menghasilkan rencana mitigasi untuk menyelesaikan insiden.
- Pencegahan Insiden — Ditemukan di halaman Pencegahan, di sinilah Anda akan menemukan rekomendasi untuk meningkatkan postur observabilitas, proses pengiriman, dan arsitektur infrastruktur Anda untuk mencegah insiden di masa depan.
- Topologi - Halaman Topologi menyediakan representasi visual interaktif dari sumber daya akun dan hubungannya di semua sumber daya di akun yang terhubung. Anda dapat melihat topologi dengan tingkat detail yang berbeda menggunakan dropdown “Tampilkan” untuk beralih antara tampilan System, Container, dan Resource.
- Keterampilan - Set instruksi modular yang memperluas AWS DevOps Agen dengan kemampuan khusus. Keterampilan berisi pengetahuan domain, metodologi investigasi, dan konfigurasi alat yang disesuaikan dengan infrastruktur Anda. Setiap keterampilan memungkinkan alat khusus dan memberikan pengungkapan instruksi secara progresif hanya jika relevan dengan penyelidikan.
- Antarmuka Obrolan bahasa alami — Tersedia di seluruh aplikasi web, Chat adalah asisten percakapan bertenaga AI yang memungkinkan Anda menanyakan infrastruktur, menganalisis kesehatan sistem, dan bekerja dengan investigasi menggunakan bahasa alami. Obrolan memberikan respons sadar konteks berdasarkan halaman yang Anda lihat.

Autentikasi

AWS DevOps Agen mendukung metode otentikasi fleksibel untuk mengakomodasi persyaratan organisasi yang berbeda:

- Integrasi Pusat Identitas IAM (Akses pengguna) — Organizations dapat menggunakan AWS Identity Center (IAM Identity Center) untuk mengelola akses pengguna secara terpusat ke aplikasi web DevOps Agent Space. IAM Identity Center dapat berfederasi dengan penyedia identitas

eksternal melalui protokol OIDC dan SAMP standar, termasuk penyedia seperti Okta, Ping Identity, dan Microsoft Entra ID. Metode ini mendukung otentikasi multi-faktor dari penyedia identitas Anda.

- Autentikasi penyedia identitas eksternal (iDP) — Organizations dapat menghubungkan penyedia identitas yang kompatibel dengan OIDC, seperti Okta atau Microsoft Entra ID, langsung ke aplikasi web Agent Space tanpa memerlukan Pusat Identitas IAM. Pengguna masuk dengan kredensi perusahaan mereka melalui iDP. Untuk petunjuk penyiapan, lihat [the section called “Menyiapkan Otentikasi Penyedia Identitas Eksternal \(IDP\)”](#).
- Tautan autentikasi IAM (Akses admin) — Metode alternatif menyediakan akses langsung ke aplikasi web dari Konsol AWS Manajemen menggunakan sesi konsol yang ada. Opsi ini berguna sebelum menerapkan integrasi Pusat Identitas penuh, tetapi sesi dibatasi hingga 10 menit.

Apa itu Ruang DevOps Agen?

Ruang DevOps Agen adalah wadah logis yang mendefinisikan alat dan infrastruktur yang dapat diakses AWS DevOps Agen. Setiap Ruang Agen beroperasi secara independen dengan akses AWS akunnya sendiri, integrasi pihak ketiga, dan izin pengguna.

Ruang Agen mewakili batas apa yang dapat diakses dan diselidiki AWS DevOps Agen selama respons insiden. Saat membuat Ruang Agen, Anda menentukan AWS akun mana yang dapat diakses agen, alat eksternal mana yang dapat dihubungkan, dan pengguna mana di organisasi Anda yang dapat berinteraksi dengan agen.

Setiap Ruang Agen berfungsi sebagai penyebaran AWS DevOps Agen yang independen. Anda mengonfigurasi Ruang Agen melalui Konsol AWS Manajemen, sementara tim operasi Anda menggunakan aplikasi web Agen Space untuk melakukan investigasi dan meninjau rekomendasi dalam ruang tersebut.

Bagaimana Ruang Agen diisolasi

Ruang Agen menjaga isolasi untuk memastikan keamanan dan mencegah akses yang tidak diinginkan di berbagai lingkungan atau tim:

- AWS isolasi akun — Setiap Ruang Agen menggunakan peran IAM khusus yang memberikan akses hanya ke AWS akun dan sumber daya tertentu. Agen tidak dapat mengakses AWS sumber daya di luar sumber daya yang dikonfigurasi secara eksplisit untuk Ruang Agen.

- Isolasi akses pengguna — Anda mengontrol pengguna atau grup mana yang dapat mengakses setiap Ruang Agen. Ini memungkinkan Anda menyelaraskan izin akses dengan struktur organisasi Anda, memastikan tim hanya berinteraksi dengan Ruang Agen yang ditunjuk.
- Isolasi data — Data investigasi, riwayat insiden, dan rekomendasi disimpan secara terpisah dalam setiap Ruang Agen. Informasi dari satu Ruang Agen tidak terlihat atau dapat diakses dari Ruang Agen lain.
- Isolasi data obrolan - Riwayat percakapan obrolan juga terisolasi dalam setiap Ruang Agen. Percakapan dan kueri di satu Ruang Agen tidak terlihat atau dapat diakses dari Ruang Agen lain.

Aplikasi Web Ruang Agen

Setiap Ruang Agen memiliki aplikasi web khusus yang dapat diakses di luar Konsol AWS Manajemen. Lihat [the section called “Apa itu Aplikasi Web DevOps Agen?”](#) untuk mempelajari lebih lanjut tentang aplikasi web.

Kapan menggunakan beberapa Ruang Agen

Pertimbangkan untuk membuat beberapa Ruang Agen untuk mendukung kebutuhan organisasi yang berbeda:

- Pemisahan tim — Buat Ruang Agen khusus untuk tim aplikasi atau unit bisnis yang berbeda untuk mempertahankan batas kepemilikan yang jelas di Ruang Agen.
- Isolasi lingkungan — Pisahkan lingkungan produksi dan non-produksi ke dalam Ruang Agen yang berbeda untuk mencegah akses lintas lingkungan yang tidak disengaja.
- Batas layanan — Sejajarkan Ruang Agen dengan layanan atau batasan aplikasi tertentu untuk menjaga agar investigasi tetap fokus dan relevan.
- Persyaratan kepatuhan — Konfigurasi Ruang Agen terpisah dengan kontrol akses atau pengaturan residensi data yang berbeda untuk memenuhi persyaratan peraturan.

Note

Saat membuat beberapa Ruang Agen, Anda dapat menggunakan AWS akun khusus sebagai akun utama untuk Ruang Agen dan menghubungkan akun aplikasi yang berbeda sebagai akun sekunder. Pendekatan ini memungkinkan Anda untuk mempertahankan kontrol akses granular sambil memastikan bahwa setiap Ruang Agen hanya dapat mengakses sumber

daya khusus untuk cakupan yang dimaksudkan, bahkan ketika menggunakan pembuatan peran otomatis.

Apa itu DevOps Topologi Agen?

AWS DevOps Agen secara otomatis menemukan dan memvisualisasikan sumber daya dan hubungan dalam aplikasi Anda dan menggunakan topologi yang dihasilkan untuk memahami infrastruktur Anda selama investigasi insiden dan saat membuat rekomendasi pencegahan.

Bagaimana grafik topologi dibuat

AWS DevOps Agen membangun grafik topologi melalui beberapa proses otomatis:

- Penemuan sumber daya — Agen secara otomatis memindai AWS akun Anda untuk mengidentifikasi sumber daya seperti instans komputasi, layanan penyimpanan, komponen jaringan, dan database yang merupakan bagian dari aplikasi Anda.
- Deteksi hubungan — Agen menganalisis data konfigurasi, CloudFormation tumpukan, dan tag sumber daya untuk menentukan bagaimana sumber daya berhubungan satu sama lain.
- Pemetaan kode dan penyebaran — Ketika terhubung ke CI/CD jaringan pipa, agen menghubungkan sumber daya infrastruktur kembali ke proses penyebaran mereka dan mengubah kode aplikasi dan infrastruktur.
- Pemetaan perilaku observabilitas — Data dari sistem observabilitas seperti Amazon CloudWatch Application Signals dan Dynatrace digunakan untuk mengidentifikasi perilaku yang diamati yang menunjukkan hubungan antar sumber daya.

Kemampuan kunci

Pemetaan sumber daya menyediakan beberapa kemampuan yang meningkatkan investigasi dan pencegahan insiden:

- Visualisasi interaktif — Jelajahi topologi aplikasi Anda melalui grafik interaktif di Aplikasi Web Operator. Anda dapat memperbesar dan menavigasi topologi untuk memahami hubungan kompleks antar sumber daya. Anda juga dapat menggunakan Obrolan untuk menanyakan informasi topologi menggunakan bahasa alami, seperti 'Tunjukkan semua fungsi Lambda yang terhubung ke tabel DynamoDB ini' atau 'Sumber daya apa yang terpengaruh oleh alarm ini? '.

- Investigasi kontekstual — Selama investigasi insiden, AWS DevOps Agen dibantu oleh topologi sumber daya untuk mengidentifikasi komponen yang terpengaruh, memahami radius ledakan, dan melacak jalur dampak melalui sistem Anda.
- Analisis akar penyebab — Pemahaman rinci tentang hubungan sumber daya membantu menentukan di mana masalah berasal, bahkan dalam sistem terdistribusi yang kompleks dengan banyak saling ketergantungan.
- Penilaian dampak — Saat menganalisis insiden, agen dapat lebih menentukan layanan hilir mana yang mungkin terpengaruh dengan mengidentifikasi rantai ketergantungan dalam topologi.
- Rekomendasi pencegahan — Agen menggunakan wawasan topologi untuk membuat rekomendasi yang ditargetkan untuk peningkatan ketahanan, menyarankan perubahan yang akan memiliki dampak paling signifikan pada stabilitas sistem.

Tampilan topologi

Visualisasi topologi di halaman Topologi di Aplikasi Web Operator menawarkan beberapa tingkat detail:

- Belajar — Tampilan default, yang dihasilkan dari skill Agent Space Understanding. Menampilkan ringkasan terstruktur infrastruktur Anda yang diatur oleh layanan logis dan jalur permintaan.
- Sistem — Menunjukkan batas akun dan wilayah tingkat tinggi.
- Container - Menampilkan tumpukan penerapan seperti CloudFormation tumpukan yang berisi sumber daya terkait.
- Komponen - Menunjukkan komponen individu dalam wadah dan hubungannya.
- Semua Sumber Daya - Menampilkan tampilan lengkap dengan semua sumber daya yang ditemukan dan hubungannya.

Penemuan sumber daya

Sumber daya ditemukan melalui dua metode:

- CloudFormation tumpukan — Agen mencantumkan semua CloudFormation tumpukan dan sumber dayanya di AWS akun utama dan akun sekunder yang terhubung. Ini didukung untuk infrastructure-as-code perangkat apa pun yang digunakan CloudFormation untuk penerapan, termasuk AWS Cloud Development Kit (AWS CDK).

- Resource Explorer — Untuk sumber daya yang tidak digunakan CloudFormation, sumber daya yang ditandai ditemukan dari AWS Resource Explorer. AWS Akun target harus mengaktifkan Resource Explorer. Ini berguna untuk mengidentifikasi batasan aplikasi untuk sumber daya yang digunakan melalui Konsol AWS Manajemen, AWS layanan APIs, atau infrastructure-as-code kerangka kerja lainnya.

Ruang lingkup investigasi di luar topologi

Sementara topologi aplikasi menyediakan konteks penting selama penyelidikan, AWS DevOps Agen tidak terbatas untuk menyelidiki hanya sumber daya yang ditunjukkan dalam topologi. Agen dapat menggunakan sumber data tambahan, seperti AWS layanan APIs atau alat observabilitas yang terhubung, untuk menyelidiki sumber daya yang tidak ada dalam topologi aplikasi.

Untuk membatasi sumber daya yang dapat diakses agen, batasi kebijakan peran yang diberikan kepada agen untuk mengakses sumber daya lintas akun. Untuk informasi selengkapnya, lihat [the section called “Membatasi Akses Agen dalam AWS Akun”](#).

Topologi dan keterampilan Pemahaman Ruang Agen

Grafik topologi dimasukkan ke dalam keterampilan belajar Agent Space Understanding, yang mengkodekan ringkasan terstruktur infrastruktur Anda untuk digunakan selama penyelidikan. Ketika penemuan topologi selesai untuk ruang agen baru, sistem secara otomatis menghasilkan keterampilan Pemahaman Ruang Agen. Untuk informasi lebih lanjut tentang keterampilan yang dipelajari, lihat [the section called “Keterampilan yang Dipelajari”](#).

DevOps Keterampilan Agen

AWS DevOps Keterampilan Agen adalah set instruksi modular yang memperluas kemampuan agen dengan pengetahuan domain khusus dan metodologi investigasi yang disesuaikan dengan infrastruktur dan alur kerja operasional Anda.

Apa itu Keterampilan

Keterampilan adalah direktori mandiri yang berisi instruksi Markdown yang memberikan kemampuan khusus kepada Agen. AWS DevOps Agen mendukung subset [spesifikasi Keterampilan Agen](#) — standar terbuka untuk instruksi dan sumber daya agen pengemasan — hanya mendukung dokumen yang tidak dapat dieksekusi: Instruksi penurunan harga, PDF, gambar, dan file data.

Setiap keterampilan membutuhkan SKILL.md file yang berisi instruksi yang ingin Anda berikan untuk AWS DevOps Agen Anda. Selain SKILL.md file yang diperlukan, keterampilan dapat mencakup:

- Alur kerja investigasi untuk skenario atau jenis infrastruktur tertentu.
- Bahan referensi termasuk pola arsitektur dan prosedur operasional.
- Penargetan tipe agen — Keterampilan dapat ditargetkan ke jenis agen tertentu (Generik,, Triase Insiden On-demand, RCA Insiden, Mitigasi Insiden, Evaluasi) untuk mengurangi konsumsi konteks dan meningkatkan fokus agen.

Mengapa menggunakan Keterampilan

Keterampilan mengubah AWS DevOps Agen dari asisten tujuan umum menjadi spesialis untuk infrastruktur dan alur kerja operasional Anda. Tidak seperti instruksi satu kali yang disediakan dalam pesan obrolan, Keterampilan adalah kemampuan yang dapat digunakan kembali yang dimuat secara otomatis bila relevan dengan tugas yang dilakukan oleh AWS DevOps Agen.

Manfaat utama:

- Spesialisasi agen Anda — Agen Penjahit AWS DevOps dengan prosedur investigasi, praktik terbaik, dan pengetahuan organisasi khusus untuk infrastruktur dan pola operasional Anda.
- Kurangi pengulangan — Buat alur kerja investigasi sekali dan AWS DevOps Agen menggunakannya secara otomatis di semua investigasi yang relevan, sehingga tidak perlu memberikan panduan yang sama berulang kali.
- Kemampuan menulis — Gabungkan beberapa Keterampilan untuk membangun alur kerja investigasi ujung ke ujung. AWS DevOps Agen membaca beberapa keterampilan selama eksekusi, seperti keterampilan untuk mengambil penerapan dari CI/CD pipeline kustom Anda dan keterampilan untuk mencari repositori kode Anda.
- Amplify custom tools — Buat skill yang memandu AWS DevOps Agen dalam menggunakan alat server MCP kustom Anda secara efektif. Keterampilan dapat mendokumentasikan kapan harus menggunakan alat tertentu, parameter apa yang akan digunakan untuk skenario yang berbeda, dan bagaimana menafsirkan hasil untuk mencapai alur kerja khusus untuk infrastruktur Anda.

Bagaimana Keterampilan bekerja

Ketika AWS DevOps Agen menemukan tugas yang relevan, ia memuat keterampilan yang sesuai dan mengikuti instruksi untuk memandu penyelidikannya. Misalnya, keterampilan “Investigasi

Kinerja Basis Data” mungkin mencakup prosedur langkah demi langkah untuk menganalisis masalah pelambatan RDS, memungkinkan agen untuk secara sistematis memeriksa status alarm, menganalisis metrik koneksi, dan mengidentifikasi kueri lambat.

Struktur keterampilan

Keterampilan diatur sebagai direktori yang berisi:

```
my-skill/  
### SKILL.md           # Main skill instructions  
### references/        # Optional: additional reference documentation  
### assets/           # Optional: images, diagrams, data files
```

SKILL.md

SKILL.md ini adalah satu-satunya file wajib. Ini berisi instruksi inti yang ditulis dalam format Markdown. File ini harus:

- Jelaskan kapan dan bagaimana menggunakan keterampilan.
- Berikan prosedur investigasi langkah demi langkah.
- Sertakan pohon keputusan untuk skenario yang berbeda.
- Dokumentasikan output yang diharapkan dan kriteria keberhasilan.

Frontmatter

Frontmatter adalah blok metadata di bagian atas SKILL.md file, tertutup di antara pembatas. --- Ini berisi name dan description bidang yang digunakan AWS DevOps Agen untuk menentukan kapan harus mengaktifkan Keterampilan selama penyelidikan atau tugas.

```
---  
name: rds-performance-investigation  
description: Investigation procedures for RDS performance issues including  
  connection exhaustion, slow queries, replication lag, and storage capacity.  
  Use this skill when investigating database latency, connection errors, or  
  read/write performance degradation.  
---
```

nama — Pengidentifikasi unik untuk Skill. Gunakan huruf kecil, angka, dan tanda hubung saja (maksimum 64 karakter). Tidak boleh memulai atau mengakhiri dengan tanda hubung.

Deskripsi — Penjelasan rinci tentang kapan dan mengapa AWS DevOps Agen harus menggunakan Skill ini. AWS DevOps Agen mengevaluasi bidang ini untuk memutuskan apakah Keterampilan relevan dengan tugas saat ini. Deskripsi yang tidak jelas atau hilang dapat menyebabkan agen melewatkan Keterampilan sepenuhnya, bahkan jika instruksinya ditulis dengan baik.

Penting — Tulis deskripsi dari perspektif agen. Sertakan skenario, layanan, jenis kesalahan, atau gejala tertentu yang seharusnya memicu Keterampilan. Misalnya, “Gunakan keterampilan ini saat menyelidiki latensi database, kesalahan koneksi, atau batas waktu kueri untuk instans Amazon RDS” lebih efektif daripada “keterampilan RDS”.

Saat Anda membuat Skill di UI, sistem akan menghasilkan frontmatter secara otomatis dari nama dan deskripsi yang Anda berikan. Keterampilan yang diunggah sebagai file zip harus menyertakan frontmatter dalam file. SKILL.md

Contoh: Keterampilan lengkap

Contoh berikut menunjukkan keterampilan yang lengkap dan terbentuk dengan baik untuk menyelidiki masalah kinerja RDS. Ini menunjukkan struktur direktori, SKILL.md frontmatter, prosedur investigasi yang dapat ditindaklanjuti, dan file referensi tambahan.

Struktur direktori:

```
rds-performance-investigation/  
### SKILL.md  
### references/  
#   ### rds-metrics-reference.md  
### assets/  
    ### rds-investigation-flowchart.png
```

SKILL.md:

```
---  
name: rds-performance-investigation  
description: Investigation procedures for RDS performance issues including  
  connection exhaustion, slow queries, replication lag, and storage capacity.  
  Use this skill when investigating database latency, connection errors, or  
  read/write performance degradation.  
---  
  
# RDS Performance Investigation
```

Use this skill when customers report database latency, connection errors, query timeouts, or read/write performance degradation.

Step 1: Check alarm status

Query CloudWatch for active alarms on the affected RDS instance. Look for:

- `DatabaseConnections` exceeding 80% of max_connections
- `ReadLatency` or `WriteLatency` above 20ms
- `FreeStorageSpace` below 20% of total storage
- `ReplicaLag` above 30 seconds (read replicas only)

Step 2: Analyze connection metrics

Retrieve `DatabaseConnections` over the past hour. If connections are near the max_connections limit, check for connection pool misconfiguration or long-running idle connections.

Step 3: Identify slow queries

Use Performance Insights (`pi:GetResourceMetrics`) to retrieve the top SQL statements by average active sessions. Focus on queries with high `db.load` contribution or frequent I/O waits.

Step 4: Summarize findings

Provide a summary with:

1. Current performance status (healthy / degraded / critical)
2. Root cause hypothesis with supporting metrics
3. Recommended remediation steps ranked by priority

references/rds-metrik-referensi.md:

RDS CloudWatch Metrics Reference

Metric	Normal Range	Investigation Threshold
DatabaseConnections	< 70% max_connections	> 80% max_connections
ReadLatency	< 5ms	> 20ms

```
| WriteLatency | < 5ms | > 20ms |  
| FreeStorageSpace | > 30% total storage | < 20% total storage |  
| ReplicaLag | < 5 seconds | > 30 seconds |  
| CPUUtilization | < 70% | > 85% |
```

Contoh: Keterampilan penyaringan insiden

Keterampilan yang ditargetkan untuk jenis agen Triage Incident dapat menentukan kriteria untuk melewati insiden secara otomatis. Gunakan ini untuk menyaring insiden yang tidak memerlukan penyelidikan. Ketika insiden baru cocok dengan kriteria lewati, AWS DevOps Agen menandainya sebagai Dilewati. Sistem memberikan alasan yang menjelaskan mengapa itu disaring.

Contoh berikut menunjukkan keterampilan yang melewati insiden prioritas rendah selama pemeliharaan terjadwal:

SKILL.md:

```
---  
name: skip-scheduled-maintenance  
description: Skip low-priority incidents during a scheduled maintenance window.  
  Use this skill to automatically filter MEDIUM and LOW severity alarms that  
  fire during planned maintenance, avoiding unnecessary investigations for  
  expected disruptions.  
---  
  
# Skip Scheduled Maintenance  
  
Skip all incidents that meet BOTH of the following criteria:  
  
1. The incident arrived between **2025-03-15 02:00 UTC** and **2025-03-15 06:00 UTC**  
2. Severity is MEDIUM or LOW  
  
Do NOT skip HIGH or CRITICAL severity incidents, even during the maintenance window.
```

Saat Anda membuat skill ini, pilih Incident Triage sebagai tipe agen. Ini memastikan keterampilan hanya dievaluasi selama fase triase.

Menciptakan Keterampilan

Sebelum membuat keterampilan, Anda harus memiliki Ruang Agen. Untuk informasi selengkapnya, lihat [the section called “Membuat Ruang Agen”](#).

Anda dapat membuat keterampilan dalam dua cara tergantung pada preferensi alur kerja dan kompleksitas keterampilan Anda:

Membuat keterampilan di UI

Keterampilan yang dibuat di Aplikasi Web Operator AWS DevOps Agen berisi nama, deskripsi, dan instruksi dalam satu SKILL.md file.

Untuk membuat keterampilan di UI:

- Arahkan ke halaman Keterampilan di Aplikasi Web Operator Ruang Agen Anda.
- Klik “Tambahkan keterampilan”.
- Pilih “Buat keterampilan” dari modal.
- Isi formulir keterampilan:
 - Nama - Huruf kecil, angka, dan tanda hubung saja (maksimum 64 karakter). Tidak boleh memulai atau mengakhiri dengan tanda hubung. Contoh: `rds-throttling-investigation`
 - Deskripsi — Penjelasan singkat tentang kapan harus menggunakan keterampilan ini (disarankan minimal 100 karakter, maksimum 1.024 karakter). Ini membantu agen menentukan kapan harus mengaktifkan keterampilan.
 - Status - Setel ke Aktif (default) atau Tidak Aktif. Keterampilan tidak aktif tidak digunakan oleh agen.
 - Jenis Agen — Pilih satu atau beberapa jenis agen yang dapat menggunakan keterampilan ini. Generik dipilih secara default dan membuat keterampilan tersedia untuk semua jenis agen. Untuk menargetkan agen tertentu, batalkan pilihan Generik dan pilih dari: On-demand, Triase Insiden, RCA Insiden, Mitigasi Insiden, atau Evaluasi.
 - Instruksi — Step-by-step prosedur dalam format Markdown. Jadilah spesifik dan dapat ditindaklanjuti.
- Klik “Buat” untuk menyimpan keterampilan.

Sistem secara otomatis menghasilkan SKILL.md file dengan struktur frontmatter yang tepat.

Untuk mengedit keterampilan yang dibuat di UI:

- Arahkan ke keterampilan dalam daftar Keterampilan dan klik keterampilan untuk membukanya.
- Klik Edit.
- Ubah nama, deskripsi, atau instruksi.

- Klik Simpan untuk memperbarui keterampilan.

Mengunggah keterampilan

Keterampilan yang diunggah sebagai file zip berisi SKILL.md file ditambah sumber daya tambahan seperti bahan referensi atau aset.

Struktur keterampilan:

```
my-skill.zip
### SKILL.md           # Required: main skill instructions
### references/       # Optional: reference documentation
#   ### architecture.md
#   ### troubleshooting.md
### assets/           # Optional: images, diagrams, data files
    ### topology.png
    ### metrics.csv
```

SKILL.md persyaratan frontmatter:

Keterampilan yang diunggah sebagai file zip harus menyertakan frontmatter di SKILL.md with name dan field. description AWS DevOps Agen menggunakan bidang ini untuk menentukan kapan harus mengaktifkan Skill. Untuk detail tentang menulis frontmatter yang efektif, lihat bagian Frontmatter sebelumnya dalam topik ini.

```
---
name: rds-performance-analysis
description: Comprehensive RDS performance investigation procedures
  for connection exhaustion, slow queries, and storage capacity issues.
  Use when investigating database latency or read/write degradation.
---

# RDS Performance Analysis

[Your skill instructions here...]
```

Untuk membuat keterampilan melalui unggahan zip:

- Buat direktori dengan file keahlian Anda mengikuti struktur di atas.

- Pastikan SKILL.md termasuk frontmatter yang tepat (nama dan deskripsi).
- Kompres direktori menjadi file.zip.
- Arahkan ke halaman Keterampilan di Aplikasi Web Operator Ruang Agen Anda.
- Klik “Tambahkan keterampilan”.
- Pilih “Upload skill” dari modal.
- Seret dan lepas file.zip Anda atau klik untuk menelusuri (hanya file ZIP, maksimum 6 MB).
- Pilih satu atau beberapa jenis agen yang dapat menggunakan keterampilan ini (Generik dipilih secara default dan berlaku untuk semua jenis agen; batal pilihan untuk target On-demand, Triase Insiden, RCA Insiden, Mitigasi Insiden, atau Evaluasi secara khusus).
- Tinjau persyaratan file zip dan hasil validasi.
- Klik “Unggah” untuk menambahkan keterampilan ke Ruang Agen Anda.

Pembatasan penting untuk keterampilan yang diunggah sebagai file zip:

- Skrip saat ini tidak didukung - Keterampilan yang berisi skrip di `scripts/` direktori akan ditolak selama pengunggahan. Eksekusi skrip akan diaktifkan dalam rilis masa depan setelah agen memiliki akses ke lingkungan pengkodean yang aman.
- Batas ukuran - Total ukuran file zip tidak boleh melebihi 6 MB (termasuk semua file).
- SKILL.md required - File zip harus berisi SKILL.md file dengan frontmatter yang valid.

Praktik terbaik untuk keterampilan penamaan:

Gunakan nama yang jelas dan deskriptif seperti “rds-throttling-investigation” daripada nama generik. Nama keterampilan yang baik mencerminkan skenario atau layanan spesifik yang dialaminya, sehingga lebih mudah untuk mengidentifikasi keterampilan yang tepat secara sekilas.

Keterampilan Mengelola

AWS DevOps Agen menyediakan kemampuan manajemen keterampilan yang komprehensif melalui Aplikasi Web Operator:

Keterampilan daftar — Lihat semua Keterampilan di Ruang Agen Anda. Halaman Keterampilan menampilkan nama keterampilan, status Aktif atau Tidak Aktif, tanggal pembuatan, tanggal pembaruan terakhir, dan tindakan yang tersedia.

Keterampilan melihat — Klik pada keterampilan apa pun untuk melihat tampilan detailnya. Keterampilan yang dibuat di UI menampilkan konten yang dapat diedit di mana Anda dapat mengubah nama, deskripsi, atau instruksi langsung di UI dan klik “Simpan” untuk memperbarui. Keterampilan yang diunggah sebagai file zip menampilkan pohon file yang ditampilkan SKILL.md dan direktori tambahan seperti referensi/dan aset/. Klik file di pohon untuk melihat isinya dalam mode hanya-baca.

Memilih agen untuk keterampilan — Konfigurasi jenis agen mana yang dapat menggunakan setiap keterampilan saat membuat atau mengeditnya. Di menu tarik-turun Jenis Agen, pilih satu atau beberapa jenis agen menggunakan kotak centang: Generik (default — berlaku untuk semua jenis agen), (kueri percakapan), Triase Insiden On-demand (penilaian insiden awal), RCA Insiden (analisis akar penyebab), Mitigasi Insiden (respons insiden otomatis), atau Evaluasi (rekomendasi proaktif). Generik dipilih secara default dan membuat keterampilan tersedia untuk semua jenis agen. Keterampilan yang ditargetkan untuk agen tertentu mengurangi konsumsi konteks dan meningkatkan fokus agen.

Mengaktifkan dan menonaktifkan keterampilan — Nonaktifkan keterampilan untuk sementara tanpa menghapusnya menggunakan sakelar. Active/Inactive Buka tampilan detail keterampilan dan alihkan sakelar ke “Tidak Aktif” untuk mencegah agen memuatnya untuk penyelidikan baru sambil mempertahankan semua konten dan konfigurasi. In-progress Investigasi terus menggunakan keterampilan. Beralih kembali ke “Aktif” untuk membuat keterampilan segera tersedia lagi.

Memperbarui keterampilan - Memodifikasi keterampilan yang ada berdasarkan bagaimana mereka diciptakan. Untuk keterampilan yang dibuat di UI, klik “Edit” di tampilan detail keterampilan, ubah nama, deskripsi, atau instruksi, dan klik “Simpan” untuk memperbarui. Untuk keterampilan yang diunggah sebagai file zip, ubah file secara lokal, buat file zip baru, dan unggah versi baru.

Menghapus keterampilan — Hapus keterampilan secara permanen dari Ruang Agen Anda. Buka tampilan daftar keterampilan, klik menu opsi lainnya () dan pilih “Hapus,” tinjau peringatan tentang penghapusan permanen, ketik nama keterampilan untuk mengonfirmasi, dan klik “Hapus Keterampilan.” Penghapusan tidak dapat dibatalkan. In-progress Investigasi mungkin terpengaruh jika mereka mencoba memuat keterampilan yang dihapus. Untuk keterampilan yang diunggah sebagai file zip, unduh file zip sebelum dihapus sebagai cadangan. Pertimbangkan untuk menonaktifkan keterampilan alih-alih menghapusnya jika Anda mungkin membutuhkannya lagi.

Migrasi dari Runbook

Runbook yang ada secara otomatis dimigrasikan ke Keterampilan tanpa memerlukan tindakan pelanggan. Saat Ruang Agen Anda beralih ke model Keterampilan, semua Runbook diubah menjadi Keterampilan dan muncul di UI Keterampilan Anda. Setelah migrasi, Anda dapat:

- Tinjau Keterampilan yang dimigrasi — Periksa apakah migrasi otomatis mengonversi Runbook Anda dengan benar.
- Perbarui sesuai kebutuhan — Edit Keterampilan langsung di UI untuk menyempurnakan instruksi, memperbarui deskripsi, atau mengonfigurasi penargetan jenis agen.
- Perluas dengan referensi — Untuk Keterampilan yang akan mendapat manfaat dari bahan referensi tambahan atau diagram arsitektur, buat ulang sebagai keterampilan unggah zip dengan direktori referensi/atau aset/.
- Buat Keterampilan baru — Tambahkan Keterampilan baru untuk alur kerja investigasi yang sebelumnya tidak tercakup oleh Runbook.

Hubungi AWS Support jika Anda mengalami masalah dengan Keterampilan yang dimigrasi secara otomatis atau memerlukan bantuan terkait pembaruan pasca-migrasi.

Keterampilan yang Dipelajari

Apa Keterampilan yang Dipelajari?

Keterampilan yang dipelajari adalah file pengetahuan terstruktur yang dihasilkan DevOps Agen dari data Ruang Agen Anda. Setiap keterampilan yang dipelajari mengkodekan jenis pengetahuan tertentu yang digunakan AWS DevOps Agen saat melakukan tugas. Saat peluncuran, dua keterampilan yang dipelajari tersedia: Pemahaman Ruang Agen dan Praktik Terbaik Penggunaan Alat.

Pemahaman Ruang Agen

Keterampilan Agent Space Understanding (`understanding-agent-space`) menganalisis akun cloud Anda yang terhubung, repositori kode, dan integrasi telemetri untuk membangun peta sumber daya dan hubungan di Ruang Agen.

Keterampilan menghasilkan SKILL .md file utama dan satu set file referensi. File utama berisi ikhtisar sistem bahasa sederhana dengan konsep domain utama, lingkungan penerapan (pasangan

AWS akun dan wilayah, langganan dan wilayah Azure, dan sebagainya), diagram arsitektur tingkat kontainer yang menunjukkan bagaimana layanan logis terhubung, jalur permintaan yang merupakan pusat aplikasi Anda dengan komponen yang mereka lintasi, dan pemetaan repositori kode ke wadah.

Setiap wadah logis menerima file referensi khusus yang menjelaskan komponen internalnya (komputasi, data, pesan, jaringan, dan lainnya) dengan jenis sumber daya dan pengidentifikasi fisik seperti ARNs, nama tabel, dan antrian. URLs File referensi juga menangkap cakupan observabilitas, termasuk alarm, dasbor, dan monitor yang ditautkan ke setiap komponen. Ini juga memetakan setiap komponen ke repositori kode, paket, dan infrastructure-as-code definisi terkait, menyediakan rantai ketertelusuran lengkap dari kode sumber ke sumber daya yang digunakan.

Setiap jalur permintaan kritis menerima file referensi khusus yang menjelaskan aliran end-to-end permintaan lengkap pada perincian komponen, dari titik masuk melalui setiap layanan perantara, penyimpanan data, dan ketergantungan eksternal. File tersebut mencakup diagram alir berurutan yang menunjukkan urutan operasi dan mekanisme interaksi antar komponen, bersama dengan tanggung jawab masing-masing peserta. Ini juga mengkatalogkan sinyal pengamatan yang relevan dengan jalur: pola grup log untuk setiap lompatan, metrik kunci (latensi, tingkat kesalahan, pelambatan, kuota token) dengan nama dan dimensi alarm mereka, dan rentang jejak terdistribusi yang dapat dikorelasikan di seluruh layanan dan akun.

Alat Gunakan Praktik Terbaik

Keterampilan Tool Use Best Practices menganalisis penggunaan alat investigasi sebelumnya untuk mengekstrak pola penggunaan yang efektif, mode kegagalan umum, dan panduan parameter. Ini membantu DevOps Agen menghindari jebakan yang diketahui dan menjalankan investigasi dengan lebih sedikit langkah yang terbuang sia-sia. Keterampilan menghasilkan file utama dan satu set file referensi per alat. File utama berfungsi sebagai indeks perutean yang mencantumkan setiap alat dengan skenario investigasi yang didukungnya dan menautkan ke file referensi yang sesuai.

Setiap file referensi per-alat dapat mencakup hingga tiga bagian:

- **Praktik Terbaik** — Teknik berbasis investigasi yang diekstrak dari penggunaan alat yang berhasil, seperti templat kueri CloudWatch Log Insights, ruang nama dan dimensi metrik khusus lingkungan, dan filter sumber peristiwa. CloudTrail Setiap entri diatur di sekitar skenario investigasi dan mencakup nilai parameter konkret dan contoh yang diamati dalam penyelidikan sebelumnya.
- **Kesalahan Umum** — Mode kegagalan berulang dan perbaikannya. Setiap entri menjelaskan kondisi kesalahan tertentu, seperti menanyakan akun yang tidak dapat diakses atau membuat kueri agregasi yang salah bentuk, dan memberikan tindakan korektif sehingga agen dapat menghindari atau memulihkan dari kesalahan tanpa membuang langkah investigasi.

- **Manajemen Output** — Panduan untuk panggilan alat yang cenderung mengembalikan respons besar. Setiap entri menjelaskan perubahan parameter atau strategi pemrosesan yang mengurangi ukuran output sambil mempertahankan nilai diagnostik.

Ketika akses infrastruktur langsung tersedia, keterampilan memvalidasi pola terhadap lingkungan Anda sebelum memasukkannya. Pola yang dikonfirmasi dinyatakan dengan percaya diri, pola yang belum dikonfirmasi menggunakan bahasa yang hati-hati, dan pola yang tidak terbukti dikecualikan. Ini membuat keterampilan tetap selaras dengan keadaan infrastruktur Anda saat ini.

Mengelola Keterampilan yang Dipelajari

Pembaruan — DevOps Agen secara otomatis menghasilkan dan memperbarui keterampilan yang dipelajari berdasarkan aktivitas di Ruang Agen Anda. Berikut ini menjelaskan kapan setiap keterampilan diperbarui.

DevOps Agen menghasilkan keterampilan Praktik Terbaik Penggunaan Alat yang diperbarui setiap 30 investigasi.

Keterampilan Pemahaman Ruang Agen dihasilkan oleh agen pembelajaran, yang berjalan setiap kali Anda menambahkan, memperbarui, atau menghapus kemampuan atau integrasi Ruang Agen.

Untuk meregenerasi keterampilan yang dipelajari secara manual, pilih tombol Regenerasi pada halaman Topologi di aplikasi operator, atau mengobrol dengan agen dan minta untuk memperbarui keterampilan yang dipelajari.

Penonaktifan - Keterampilan yang dipelajari aktif secara default. Saat aktif, DevOps Agen memuatnya di awal setiap tugas DevOps Agen. Untuk menghentikan keterampilan yang dipelajari agar tidak diterapkan, nonaktifkan keterampilan dari penampil keterampilan di aplikasi operator. Menonaktifkan keterampilan tidak menghapusnya. Keterampilan dipertahankan dan dapat diaktifkan kembali kapan saja. Ketika keterampilan dinonaktifkan, DevOps Agen beroperasi tanpa sepengetahuan keterampilan itu.

Tampilan topologi — Halaman Topologi di aplikasi web Agent Space Anda menggunakan Keterampilan Memahami Ruang Agen untuk menampilkan lingkungan Ruang Agen Anda secara visual sebagai wadah dan komponen logis. Klik wadah apa pun untuk melihat komponennya, pengidentifikasi sumber daya, dan telemetri.

Instruksi agen

Gunakan instruksi agen untuk memberikan panduan yang selalu diterapkan AWS DevOps Agen untuk setiap sesi. Sesi adalah percakapan tunggal atau investigasi dengan agen. Pada halaman Agen di Aplikasi Web Operator Ruang Agen, Anda dapat mengatur instruksi Global yang berlaku untuk semua agen, atau menetapkan instruksi untuk agen terkelola tertentu seperti triase Obrolan atau Insiden. Instruksi ini disimpan sebagai AGENTS.md file. Tidak seperti [the section called “DevOps Keterampilan Agen”](#), yang memuat sesuai permintaan ketika agen mencocokkan deskripsi keterampilan dengan tugas saat ini, instruksi agen selalu ada sejak awal setiap sesi, terlepas dari apa yang sedang dikerjakan agen.

Apa itu instruksi agen

Instruksi agen memberikan panduan tanpa syarat dan selalu aktif untuk agen Anda. Pada awal setiap sesi, layanan agen mengambil instruksi yang dikonfigurasi untuk Ruang Agen Anda dan menyuntikkan konten mereka langsung ke prompt sistem agen. Agen tidak memutuskan apakah akan memuatnya, mereka selalu hadir.

Setiap sesi agen menerima instruksi dari instruksi global dan instruksi khusus agen yang relevan, misalnya, Obrolan.

Instruksi agen disimpan sebagai AGENTS.md file dan berbeda dari beberapa [the section called “DevOps Keterampilan Agen”](#) cara penting:

Aspek	Keterampilan	Instruksi agen (AGENTS.md)
Nama dan deskripsi	Diperlukan	Tidak berlaku
Format konten	Bundel penurunan harga atau ZIP	Hanya penurunan harga
File sumber daya	Didukung	Tidak Support
Injeksi konteks	Sesuai permintaan (agen memutuskan melalui pencocokan deskripsi keterampilan)	Selalu (tanpa syarat, setiap sesi)

Aspek	Keterampilan	Instruksi agen (AGENTS.md)
Keunikan	Beberapa per Ruang Agen	Satu per agen (satu untuk instruksi global, satu per agen terkelola)

Instruksi agen tidak memiliki bidang nama atau deskripsi. AGENTS.md File yang mendasarinya hanya berisi penurunan harga tanpa frontmatter, tidak ada dukungan bundel ZIP, dan tidak ada file sumber daya.

Mengapa menggunakan instruksi agen

Instruksi agen memberi Anda cara yang andal untuk memastikan panduan tertentu selalu dalam konteks, tanpa bergantung pada keputusan pemuatan keterampilan agen.

Manfaat utama:

- **Prediktabilitas:** Instruksi selalu ada, terlepas dari tugas apa yang sedang dikerjakan agen. Tidak diperlukan pencocokan deskripsi, dan agen tidak dapat melewati konten.
- **Cakupan terjamin:** Tidak seperti Keterampilan, yang mungkin atau mungkin tidak dimuat agen tergantung pada relevansi tugas, instruksi agen selalu disuntikkan pada awal setiap sesi.
- **Kebijakan tetap:** Gunakan instruksi agen untuk kebijakan operasional tetap, pedoman keamanan, standar pengkodean, atau panduan apa pun yang harus berlaku untuk setiap sesi tanpa kecuali.
- **Cakupan yang ditargetkan:** Anda dapat menerapkan instruksi ke semua jenis agen sekaligus menggunakan instruksi global, atau membatasi instruksi ke jenis agen tertentu ketika panduan hanya relevan dengan pekerjaan agen tersebut.

Bagaimana instruksi agen bekerja

Ketika sesi dimulai, layanan agen mengambil instruksi yang dikonfigurasi untuk Ruang Agen Anda dan menyuntikkan kontennya ke dalam prompt sistem agen sebelum sesi dimulai. Ini terjadi secara otomatis untuk setiap sesi. Agen tidak mengevaluasi apakah akan memuatnya; itu selalu menyuntikkan konten.

Setiap sesi baru memuat instruksi baru saat startup. Jika Anda memperbarui instruksi, perubahan akan segera berlaku untuk sesi yang dimulai setelah Anda menyimpan. Sesi yang sudah berlangsung terus menggunakan konten yang dimuat saat dimulai.

Pelengkupan menentukan instruksi mana yang diterima sesi. Instruksi global berlaku untuk semua jenis agen di Ruang Agen Anda, sehingga setiap sesi menerimanya. Agent-specific instruksi hanya berlaku untuk sesi jenis agen tertentu. Sesi menerima instruksi dari instruksi global dan instruksi khusus agen yang relevan.

Pelengkupan jenis agen

Scoping mengontrol sesi agen mana yang menerima serangkaian instruksi tertentu. Ada dua opsi pelengkupan:

- Instruksi global: Berlaku untuk semua jenis agen di Ruang Agen Anda. Setiap sesi agen menerima konten ini.
- Agent-specific: Hanya berlaku untuk sesi dari jenis agen yang dipilih.

Agen terkelola yang tersedia untuk instruksi khusus agen adalah:

- Obrolan - Ad-hoc pertanyaan dan permintaan selama sesi obrolan.
- Triase insiden - Penyaringan alarm, klasifikasi keparahan, dan pelengkupan awal.
- Insiden RCA - Analisis akar penyebab dengan pengumpulan dan validasi bukti.
- Mitigasi insiden - Short-term remediasi dan rekomendasi perbaikan jangka panjang.
- Evaluasi - Penilaian kinerja agen dan pemeriksaan kepatuhan kebijakan.

Panduan ukuran konten

Setiap kali Anda memulai percakapan atau investigasi, agen membaca seluruh instruksi Anda sebelum melakukan hal lain. Agen memiliki jumlah memori kerja yang tetap per sesi, dan instruksi Anda menggunakan beberapa di antaranya. Semakin besar file, semakin sedikit ruang yang tersisa untuk pertanyaan Anda, investigasi, log yang dibaca agen, dan alasannya sendiri. Instruksi yang lebih pendek dan terfokus memberi agen lebih banyak kapasitas untuk menyelesaikan masalah Anda.

- Batas keras: 25 KB
- Ukuran yang disarankan: 120 baris (direkomendasikan untuk sebagian besar konfigurasi)

Fokuskan instruksi Anda pada panduan yang harus ada di setiap sesi. Untuk prosedur investigasi khusus yang hanya berlaku untuk tugas-tugas tertentu, pertimbangkan untuk menggunakan [the section called “DevOps Keterampilan Agen”](#) sebagai gantinya.

Contoh

Contoh berikut menunjukkan instruksi agen yang terbentuk dengan baik dengan panduan investigasi, standar pemformatan respons, dan persyaratan keamanan yang berlaku untuk setiap sesi agen.

```
# Agent Instructions

## Investigation approach
- Always check CloudWatch alarms and recent deployments before proposing a root cause.

## Response format
- Lead with a one-sentence summary of findings before listing details.
- Include the AWS region and resource identifier for any resource you reference.
- Use bullet points for lists of findings or recommendations.

## Security
- Never log, display, or suggest storing credentials or secrets in plaintext.
- When recommending IAM changes, follow least-privilege principles.
```

Mengatur instruksi agen

Sebelum mengatur instruksi agen, Anda harus memiliki Ruang Agen. Untuk informasi selengkapnya, lihat [the section called “Membuat Ruang Agen”](#).

Setiap agen memiliki tepat satu set instruksi. Ketika Anda menyimpan konten baru, itu menimpa konten yang ada untuk agen itu.

Untuk mengatur instruksi global (berlaku untuk semua agen):

1. Arahkan ke halaman Agen di Aplikasi Web Operator Ruang Agen Anda.
2. Pilih Lihat di samping Instruksi global.
3. Masukkan instruksi penurunan harga Anda di editor.
4. Pilih Simpan.

Untuk mengatur instruksi untuk agen tertentu:

1. Arahkan ke halaman Agen di Aplikasi Web Operator Ruang Agen Anda.
2. Di bawah Agen terkelola, pilih Lihat di samping agen yang ingin Anda konfigurasi: Obrolan, Triase insiden, RCA Insiden, Mitigasi insiden, atau Evaluasi.
3. Masukkan instruksi penurunan harga Anda di editor.
4. Pilih Simpan.

Mengelola instruksi agen

AWS DevOps Agen menyediakan kemampuan manajemen untuk instruksi agen melalui Aplikasi Web Operator.

Melihat petunjuk: Arahkan ke halaman Agen dan pilih Lihat di samping Instruksi global atau agen terkelola tertentu. Editor menunjukkan konten saat ini. Gunakan tab Pratinjau untuk melihat penurunan harga yang dirender atau tab Kode untuk melihat penurunan harga mentah.

Instruksi pengeditan: Buka agen seperti dijelaskan di atas, ubah konten di editor, dan pilih Simpan.

Mengunggah instruksi dari file: Buka agen, lalu pilih tombol Unggah di editor untuk mengunggah file penurunan harga dari komputer Anda.

Petunjuk pengunduhan: Buka agen, lalu pilih tombol Unduh di editor untuk mengunduh konten saat ini sebagai file.

Menghapus instruksi: Buka agen, pilih tombol Hapus di editor, dan konfirmasi penghapusan. Tindakan ini tidak dapat dibatalkan. Pertimbangkan untuk mengunduh konten terlebih dahulu jika Anda mungkin membutuhkannya lagi.

Wilayah yang Didukung

Topik ini menjelaskan AWS Wilayah tempat Anda dapat menggunakan AWS DevOps Agen. Untuk informasi selengkapnya tentang AWS Wilayah, lihat [Menentukan AWS Wilayah mana yang dapat digunakan akun Anda](#) dalam Panduan Referensi Manajemen AWS Akun.

Pemantauan sumber daya lintas wilayah

AWS DevOps Agen dapat memantau dan menyelidiki sumber daya di AWS akun yang terletak di AWS Wilayah mana pun, terlepas dari Wilayah yang didukung tempat Anda membuat Ruang Agen.

Saat Anda mengaitkan AWS akun dengan Ruang Agen, agen menemukan dan memetakan sumber daya di semua Wilayah dalam akun tersebut. Ini berarti Anda tidak memerlukan Ruang Agen di setiap Wilayah tempat beban kerja Anda berjalan.

Pilih Wilayah yang didukung berdasarkan residensi data pilihan Anda, kedekatan dengan tim operasi Anda, atau persyaratan organisasi.

Wilayah yang Didukung

AWS DevOps Agen tersedia di AWS Wilayah berikut.

Nama wilayah	Kode Wilayah	Tautan Konsol
US East (Northern Virginia)	us-east-1	Buka konsol
AS Barat (Oregon)	us-west-2	Buka konsol
Asia Pasifik (Sydney)	ap-southeast-2	Buka konsol
Asia Pasifik (Tokyo)	ap-northeast-1	Buka konsol
Eropa (Frankfurt)	eu-central-1	Buka konsol
Eropa (Irlandia)	eu-west-1	Buka konsol

Titik akhir layanan

Nama wilayah	Kode Wilayah	Titik akhir	Protokol
US East (N. Virginia)	us-east-1	aidevops.us-east-1 .amazonaws.com	HTTPS
US West (Oregon)	us-west-2	aidevops.us-west-2 .amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	aidevops.ap-southe ast-2.amazonaws.co m	HTTPS

Nama wilayah	Kode Wilayah	Titik akhir	Protokol
Asia Pacific (Tokyo)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europe (Frankfurt)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europe (Ireland)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Pertimbangan-pertimbangan

- Pemilihan Wilayah Luar Angkasa Agen — Ruang Agen dan datanya (investigasi,

topologi, rekomendasi) disimpan di Wilayah tempat Anda membuatnya. Pilih Wilayah yang memenuhi persyaratan residensi data Anda.

- Pemantauan Lintas Wilayah - Sumber daya dalam AWS akun yang terkait dengan Agen

Ruang dipantau terlepas dari Wilayah mana sumber daya tersebut digunakan. Anda tidak perlu membuat Ruang Agen terpisah di setiap Wilayah tempat beban kerja Anda berjalan.

- Integrasi pihak ketiga — Koneksi ke CI/CD penyedia (GitHub, GitLab),

alat observabilitas (Dynatrace, Datadog, New Relic, Splunk), dan server MCP dikonfigurasi per Ruang Agen dan tidak bergantung pada Wilayah.

Memulai dengan AWS DevOps Agen

Dalam panduan memulai ini, Anda akan membuat Ruang Agen dasar, mengonfigurasi izin minimal, dan melakukan penyelidikan pertama yang didukung AI.

Topik:

- [the section called “Membuat Ruang Agen”](#)
- [the section called “AWS DevOps Panduan orientasi Agen CLI”](#)
- [the section called “Menciptakan lingkungan pengujian”](#)
- [the section called “Memulai AWS DevOps Agen menggunakan AWS CDK”](#)
- [the section called “Memulai dengan AWS DevOps Agen menggunakan AWS CloudFormation”](#)
- [the section called “Memulai dengan AWS DevOps Agen menggunakan Terraform”](#)

Membuat Ruang Agen

Ruang Agen mendefinisikan alat dan infrastruktur yang dapat diakses AWS DevOps Agen. Panduan ini memandu Anda melalui pembuatan Ruang Agen, mengonfigurasi akses akun utama, dan mengaktifkan Aplikasi Web DevOps Agen. Lihat “Apa itu Ruang Agen” untuk mempelajari lebih lanjut tentang konsep Ruang Agen.

Membuat Ruang Agen

Akses konsol AWS DevOps Agen

1. Masuk ke Konsol AWS Manajemen
2. Arahkan ke konsol AWS DevOps Agen

Nama Agen Space

1. Klik Buat Ruang Agen

Di bagian detail Ruang Agen, berikan:

1. Di bidang Nama, masukkan nama untuk Ruang Agen Anda

2. (Opsional) Di bidang Deskripsi, tambahkan detail tentang tujuan Ruang Agen
3. (Opsional) Dari dropdown bahasa respons Agen, pilih bahasa yang digunakan agen saat menghasilkan tanggapan, temuan, dan hasil investigasi. Pilihannya meliputi: Bahasa Indonesia, Mandarin (Simplified/PRC), Chinese (Traditional/Taiwan), Inggris (Inggris), Prancis (Prancis), Jerman (Jerman), Italia (Italia), Jepang (Jepang), Korea (Korea), Portugis (Brasil), Spanyol (Amerika Latin), Turki (Turki), Arab (Arab Saudi), Thailand (Thailand), dan Vietnam (Vietnam). Jika tidak ada bahasa yang dipilih, agen merespons dalam bahasa input. Pengaturan ini juga digunakan untuk menentukan bahasa untuk kasus AWS Support yang dibuat melalui fitur [Ask for human support](#).

Mengkonfigurasi akses akun utama

Di bagian Beri akses AWS sumber daya Ruang Agen ini, Anda akan menyiapkan peran IAM untuk memberikan akses Ruang Agen ke AWS akun utama. Akun utama adalah AWS akun tempat Anda membuat Ruang Agen. AWS DevOps Agen memerlukan peran IAM untuk menemukan dan mengakses AWS sumber daya dalam akun ini selama penyelidikan.

Pilih metode konfigurasi peran. Pilih salah satu opsi berikut:

Opsi 1: Buat peran AWS DevOps Agen baru secara otomatis (disarankan)

Opsi ini secara otomatis membuat peran dengan izin yang sesuai bagi AWS DevOps Agen untuk menyelidiki sumber daya di akun Anda.

Note

Anda harus memiliki izin IAM untuk membuat peran baru untuk menggunakan opsi ini.

1. Pilih Auto-create peran AWS DevOps Agen baru
2. (Opsional) Perbarui nama peran Ruang Agen yang akan dibuat

Opsi 2: Tetapkan peran yang ada

Gunakan opsi ini ketika administrator lain sebelumnya telah membuat peran khusus untuk AWS DevOps Agen.

1. Pilih Tetapkan peran yang ada

2. Dari menu tarik-turun, pilih peran yang ada yang memiliki izin yang sesuai

Opsi 3: Buat peran AWS DevOps Agen baru menggunakan templat kebijakan

Gunakan opsi ini ketika Anda perlu membatasi layanan dan sumber daya yang dapat diakses agen di akun utama.

1. Pilih Buat peran AWS DevOps Agen baru menggunakan templat kebijakan
2. Ikuti petunjuk untuk membuat kebijakan kepercayaan peran baru dan kebijakan inline.

Mengaktifkan Aplikasi Web Ruang Agen

Aplikasi Web adalah tempat personel berinteraksi dengan AWS DevOps Agen untuk investigasi insiden dan meninjau rekomendasi. Lihat [Arsitektur Konsol AWS DevOps Agen \[tautan\]](#) untuk mempelajari lebih lanjut. Saat diaktifkan, pengguna dapat mengakses Aplikasi Web Ruang Agen melalui tautan autentikasi IAM dari Konsol AWS Manajemen.

Pilih salah satu opsi berikut:

Opsi 1: Buat peran AWS DevOps Agen baru secara otomatis (disarankan)

Opsi ini secara otomatis membuat peran dengan izin yang sesuai untuk mengakses Aplikasi Web DevOps Agen.

Note

Anda harus memiliki izin IAM untuk membuat peran baru untuk menggunakan opsi ini.

1. Pilih Auto-create peran AWS DevOps Agen baru
2. Tinjau izin yang akan diberikan untuk peran tersebut

Opsi 2: Tetapkan peran yang ada

Gunakan opsi ini ketika administrator lain sebelumnya telah membuat peran operator.

1. Pilih Tetapkan peran yang ada
2. Dari menu tarik-turun, pilih peran yang ada yang memiliki izin yang sesuai

Opsi 3: Buat peran AWS DevOps Agen baru menggunakan templat kebijakan

Gunakan opsi ini saat Anda perlu menyesuaikan izin untuk akses aplikasi web.

1. Pilih Buat peran AWS DevOps Agen baru menggunakan templat kebijakan
2. Ikuti petunjuk untuk membuat kebijakan kepercayaan peran baru dan kebijakan inline.

Menambahkan tag (opsional)

Anda dapat menambahkan AWS tag ke Ruang Agen Anda selama pembuatan. Tag adalah pasangan nilai kunci yang membantu Anda mengatur dan mengidentifikasi sumber daya Anda. Anda dapat menambahkan hingga 50 tag per Ruang Agen. Untuk menambahkan tag, perluas bagian Tag pada halaman Create Agent Space dan klik Add new tag.

Penciptaan ruang agen lengkap

Setelah semua bagian diisi, klik Buat

Memverifikasi pengaturan Ruang Agen Anda

Setelah dikonfigurasi, tombol akses Operator akan muncul di halaman detail Ruang Agen. Mengkliknya akan membuka Aplikasi Web di tab baru dan berhasil mengautentikasi.

Langkah selanjutnya

Setelah menyiapkan Ruang Agen Anda, pertimbangkan langkah-langkah berikut:

- Tambahkan akun sekunder jika aplikasi Anda menjangkau beberapa AWS akun
- Konfigurasi integrasi pihak ketiga seperti alat observabilitas atau sistem tiket
- Menyiapkan otentikasi Pusat AWS Identitas untuk lingkungan produksi
- Jelajahi pemetaan sumber daya aplikasi Anda untuk membantu AWS DevOps Agen memahami infrastruktur Anda

AWS DevOps Panduan orientasi Agen CLI

Ikhtisar

Dengan AWS DevOps Agen, Anda dapat memantau dan mengelola AWS infrastruktur Anda. Panduan ini memandu Anda melalui pengaturan AWS DevOps Agen dengan menggunakan AWS

Command Line Interface (AWS CLI). Anda membuat peran IAM, menyiapkan ruang agen, dan mengaitkan AWS akun Anda. Anda juga mengaktifkan aplikasi operator dan secara opsional menghubungkan integrasi pihak ketiga. Panduan ini membutuhkan waktu sekitar 20 menit untuk menyelesaikannya.

AWS DevOps Agen tersedia di enam AWS Wilayah: AS Timur (Virginia N.), AS Barat (Oregon), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), dan Eropa (Irlandia). Untuk informasi selengkapnya tentang Wilayah yang didukung, lihat [the section called “Wilayah yang Didukung”](#).

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki yang berikut:

- AWS CLI versi 2 diinstal dan dikonfigurasi
- Otentikasi ke akun AWS pemantauan Anda
- Izin untuk membuat peran AWS Identity and Access Management (IAM) dan melampirkan kebijakan
- AWS Akun untuk digunakan sebagai akun pemantauan
- Keakraban dengan sintaks AWS CLI dan JSON

Sepanjang panduan ini, ganti nilai placeholder berikut dengan milik Anda sendiri:

- `<MONITORING_ACCOUNT_ID>`— ID AWS akun 12 digit Anda untuk akun pemantauan (utama)
- `<EXTERNAL_ACCOUNT_ID>`— 12 digit ID AWS akun dari akun sekunder untuk dipantau (digunakan pada langkah 4)
- `<REGION>`— Kode AWS Wilayah untuk ruang agen Anda (misalnya, `us-east-1` atau `eu-central-1`)
- `<AGENT_SPACE_ID>`— Pengidentifikasi ruang agen yang dikembalikan oleh perintah `create-agent-space`

Pengaturan peran IAM

1. Buat peran ruang DevOps Agen

Buat kebijakan kepercayaan IAM dengan menjalankan perintah berikut:

```
cat > devops-agentspace-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>"
        },
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/*"
        }
      }
    }
  ]
}
EOF
```

Buat peran IAM:

```
aws iam create-role \
  --region <REGION> \
  --role-name DevOpsAgentRole-AgentSpace \
  --assume-role-policy-document file:///devops-agentspace-trust-policy.json
```

Simpan peran ARN dengan menjalankan perintah berikut:

```
aws iam get-role --role-name DevOpsAgentRole-AgentSpace --query 'Role.Arn' --output
text
```

Lampirkan kebijakan AWS terkelola:

```
aws iam attach-role-policy \
  --role-name DevOpsAgentRole-AgentSpace \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
```

Membuat dan melampirkan kebijakan inline untuk memungkinkan pembuatan peran terkait layanan Resource Explorer:

```
cat > devops-agentspace-additional-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateServiceLinkedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/aws-service-role/resource-explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
EOF

aws iam put-role-policy \
  --role-name DevOpsAgentRole-AgentSpace \
  --policy-name AllowCreateServiceLinkedRoles \
  --policy-document file:///devops-agentspace-additional-policy.json
```

2. Buat peran IAM aplikasi operator

Buat kebijakan kepercayaan IAM dengan menjalankan perintah berikut:

```
cat > devops-operator-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

```

    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>"
      },
      "ArnLike": {
        "aws:SourceArn":
"arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/*"
      }
    }
  }
]
}
EOF

```

Buat peran IAM:

```

aws iam create-role \
  --role-name DevOpsAgentRole-WebappAdmin \
  --assume-role-policy-document file:///devops-operator-trust-policy.json \
  --region <REGION>

```

Simpan peran ARN dengan menjalankan perintah berikut:

```

aws iam get-role --role-name DevOpsAgentRole-WebappAdmin --query 'Role.Arn' --output
text

```

Lampirkan kebijakan aplikasi operator AWS terkelola:

```

aws iam attach-role-policy \
  --role-name DevOpsAgentRole-WebappAdmin \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsOperatorAppAccessPolicy

```

Kebijakan terkelola ini memberikan izin aplikasi operator untuk mengakses fitur ruang agen. Fitur-fitur ini termasuk investigasi, rekomendasi, manajemen pengetahuan, obrolan, dan integrasi AWS Support. Kebijakan mencakup akses ke ruang agen tertentu dengan menggunakan `aws:PrincipalTag/AgentSpaceId` kondisi tersebut. Untuk informasi selengkapnya tentang daftar lengkap tindakan, lihat [the section called “DevOps Izin Agen IAM”](#).

Langkah-langkah orientasi

1. Buat ruang agen

Jalankan perintah berikut untuk membuat ruang agen:

```
aws devops-agent create-agent-space \  
  --name "MyAgentSpace" \  
  --description "AgentSpace for monitoring my application" \  
  --region <REGION>
```

Secara opsional, tentukan `--kms-key-arn` untuk menggunakan kunci AWS KMS yang dikelola pelanggan untuk enkripsi. Anda juga dapat menggunakan `--tags` untuk menambahkan tag sumber daya dan `--locale` mengatur bahasa untuk tanggapan agen.

Simpan `agentSpaceId` dari respons (terletak di `agentSpace.agentSpaceId`).

Untuk membuat daftar spasi agen Anda nanti, jalankan perintah berikut:

```
aws devops-agent list-agent-spaces \  
  --region <REGION>
```

2. Kaitkan AWS akun Anda

Kaitkan AWS akun Anda untuk mengaktifkan penemuan topologi. Atur `accountType` ke salah satu nilai berikut:

- `monitor`— Akun utama tempat ruang agen ada. Akun ini menampung agen dan digunakan untuk penemuan topologi.
- `source`— Akun tambahan yang dipantau agen. Gunakan jenis ini saat Anda mengaitkan akun eksternal di langkah 4.

```
aws devops-agent associate-service \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --service-id aws \  
  --configuration '{  
    "aws": {  
      "assumableRoleArn": "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/DevOpsAgentRole-  
AgentSpace",  
      "accountId": "<MONITORING_ACCOUNT_ID>",
```

```
    "accountType": "monitor"
  }
}' \
--region <REGION>
```

3. Aktifkan aplikasi operator

Alur otentikasi dapat menggunakan IAM, IAM Identity Center (IDC), atau penyedia identitas eksternal (iDP). Jalankan perintah berikut untuk mengaktifkan aplikasi operator untuk ruang agen Anda:

```
aws devops-agent enable-operator-app \
  --agent-space-id <AGENT_SPACE_ID> \
  --auth-flow iam \
  --operator-app-role-arn "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/DevOpsAgentRole-WebappAdmin" \
  --region <REGION>
```

Untuk autentikasi IAM Identity Center, gunakan `--auth-flow idc` dan sediakan `--idc-instance-arn`. Untuk penyedia identitas eksternal, gunakan `--auth-flow idp` dan sediakan `--issuer-url`, `--idp-client-id`, dan `--idp-client-secret`. Untuk informasi selengkapnya, lihat [the section called “Menyiapkan Autentikasi Pusat Identitas IAM”](#) dan [the section called “Menyiapkan Otentikasi Penyedia Identitas Eksternal \(IDP\)”](#).

Catatan: Jika sebelumnya Anda membuat peran aplikasi operator untuk ruang agen lain di akun Anda, Anda dapat menggunakan kembali peran tersebut ARN.

4. (Opsional) Kaitkan akun sumber tambahan

Untuk memantau akun tambahan dengan AWS DevOps Agen, buat peran lintas akun IAM.

Buat peran lintas akun di akun eksternal

Beralih ke akun eksternal dan buat kebijakan kepercayaan. `MONITORING_ACCOUNT_ID` ini adalah akun utama yang menampung ruang agen yang Anda atur di langkah 2. Konfigurasi ini memungkinkan layanan AWS DevOps Agen untuk mengambil peran dalam akun sumber sekunder atas nama akun pemantauan.

Jalankan perintah berikut untuk membuat kebijakan kepercayaan:

```
cat > devops-cross-account-trust-policy.json << 'EOF'
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "aidevops.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>",
        "sts:ExternalId":
"arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/<AGENT_SPACE_ID>"
      }
    }
  }
]
}
EOF

```

Buat peran IAM lintas akun:

```

aws iam create-role \
  --role-name DevOpsAgentCrossAccountRole \
  --assume-role-policy-document file://devops-cross-account-trust-policy.json

```

Simpan peran ARN dengan menjalankan perintah berikut:

```

aws iam get-role --role-name DevOpsAgentCrossAccountRole --query 'Role.Arn' --output
text

```

Lampirkan kebijakan AWS terkelola:

```

aws iam attach-role-policy \
  --role-name DevOpsAgentCrossAccountRole \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy

```

Lampirkan kebijakan inline untuk mengizinkan pembuatan peran terkait layanan Resource Explorer di akun eksternal:

```

cat > devops-cross-account-additional-policy.json << 'EOF'

```

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateServiceLinkedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<EXTERNAL_ACCOUNT_ID>:role/aws-service-role/resource-explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
EOF

aws iam put-role-policy \
  --role-name DevOpsAgentCrossAccountRole \
  --policy-name AllowCreateServiceLinkedRoles \
  --policy-document file:///devops-cross-account-additional-policy.json

```

Kaitkan akun eksternal

Beralih kembali ke akun pemantauan Anda, lalu jalankan perintah berikut untuk mengaitkan akun eksternal:

```

aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id aws \
  --configuration '{
    "sourceAws": {
      "accountId": "<EXTERNAL_ACCOUNT_ID>",
      "accountType": "source",
      "assumableRoleArn": "arn:aws:iam::<EXTERNAL_ACCOUNT_ID>:role/DevOpsAgentCrossAccountRole"
    }
  }' \
  --region <REGION>

```

5. (Opsional) Rekanan GitHub

Catatan: Anda harus terlebih dahulu mendaftar GitHub melalui konsol AWS DevOps Agen dengan menggunakan OAuth alur sebelum Anda dapat mengaitkannya melalui CLI.

Untuk petunjuk tentang mendaftar GitHub melalui konsol, lihat [the section called “Menghubungkan ke CI/CD jaringan pipa”](#).

Daftar layanan terdaftar:

```
aws devops-agent list-services \  
  --region <REGION>
```

Simpan <SERVICE_ID> untuk ServiceType: `github`

Setelah Anda mendaftar GitHub di konsol, asosiasikan GitHub repositori dengan menjalankan perintah berikut:

```
aws devops-agent associate-service \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --service-id <SERVICE_ID> \  
  --configuration '{  
    "github": {  
      "repoName": "<GITHUB_REPO_NAME>",  
      "repoId": "<GITHUB_REPO_ID>",  
      "owner": "<GITHUB_OWNER>",  
      "ownerType": "organization"  
    }  
  }' \  
  --region <REGION>
```

6. (Opsional) Daftar dan asosiasikan ServiceNow

Pertama, daftarkan ServiceNow layanan dengan OAuth kredensial:

```
aws devops-agent register-service \  
  --service servicenow \  
  --service-details '{  
    "servicenow": {  
      "instanceUrl": "<SERVICENOW_INSTANCE_URL>",
```

```

    "authorizationConfig": {
      "oAuthClientCredentials": {
        "clientName": "<SERVICENOW_CLIENT_NAME>",
        "clientId": "<SERVICENOW_CLIENT_ID>",
        "clientSecret": "<SERVICENOW_CLIENT_SECRET>"
      }
    }
  }
}' \
--region <REGION>

```

Simpan yang dikembalikan<SERVICE_ID>, lalu kaitkan ServiceNow:

```

aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "servicenow": {
      "instanceUrl": "<SERVICENOW_INSTANCE_URL>"
    }
  }' \
  --region <REGION>

```

7. (Opsional) Daftar dan asosiasikan Dynatrace

Pertama, daftarkan layanan Dynatrace dengan OAuth kredensi:

```

aws devops-agent register-service \
  --service dynatrace \
  --service-details '{
    "dynatrace": {
      "accountUrn": "<DYNATRACE_ACCOUNT_URN>",
      "authorizationConfig": {
        "oAuthClientCredentials": {
          "clientName": "<DYNATRACE_CLIENT_NAME>",
          "clientId": "<DYNATRACE_CLIENT_ID>",
          "clientSecret": "<DYNATRACE_CLIENT_SECRET>"
        }
      }
    }
  }' \
  --region <REGION>

```

Simpan yang dikembalikan<SERVICE_ID>, lalu kaitkan Dynatrace. Sumber daya bersifat opsional. Lingkungan menentukan lingkungan Dynatrace mana yang akan diasosiasikan.

```
aws devops-agent associate-service \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --service-id <SERVICE_ID> \  
  --configuration '{  
    "dynatrace": {  
      "envId": "<DYNATRACE_ENVIRONMENT_ID>",  
      "resources": [  
        "<DYNATRACE_RESOURCE_1>",  
        "<DYNATRACE_RESOURCE_2>"  
      ]  
    }  
  }'  
  --region <REGION>
```

Responsnya mencakup informasi webhook untuk integrasi. Anda dapat menggunakan webhook ini untuk memicu penyelidikan dari Dynatrace. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan Dynatrace”](#).

8. (Opsional) Daftar dan asosiasikan Splunk

Pertama, daftarkan layanan Splunk dengan BearerToken kredensial.

Titik akhir menggunakan format berikut: `https://<XXX>.api.scs.splunk.com/<XXX>/mcp/v1/`

```
aws devops-agent register-service \  
  --service mcpserversplunk \  
  --service-details '{  
    "mcpserversplunk": {  
      "name": "<SPLUNK_NAME>",  
      "endpoint": "<SPLUNK_ENDPOINT>",  
      "authorizationConfig": {  
        "bearerToken": {  
          "tokenName": "<SPLUNK_TOKEN_NAME>",  
          "tokenValue": "<SPLUNK_TOKEN_VALUE>"  
        }  
      }  
    }  
  }'  
  \
```

```
--region <REGION>
```

Simpan yang dikembalikan<SERVICE_ID>, lalu kaitkan Splunk:

```
aws devops-agent associate-service \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --service-id <SERVICE_ID> \  
  --configuration '{  
    "mcpserverSplunk": {  
      "name": "<SPLUNK_NAME>",  
      "endpoint": "<SPLUNK_ENDPOINT>"  
    }  
  }' \  
  --region <REGION>
```

Responsnya mencakup informasi webhook untuk integrasi. Anda dapat menggunakan webhook ini untuk memicu penyelidikan dari Splunk. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan Splunk”](#).

9. (Opsional) Daftar dan asosiasikan New Relic

Pertama, daftarkan layanan New Relic dengan kredensial kunci API.

Wilayah: Salah satu US atau EU.

Bidang opsional: applicationIds, entityGuids, alertPolicyIds

```
aws devops-agent register-service \  
  --service mcpservernewrelic \  
  --service-details '{  
    "mcpservernewrelic": {  
      "authorizationConfig": {  
        "apiKey": {  
          "apiKey": "<YOUR_NEW_RELIC_API_KEY>",  
          "accountId": "<YOUR_ACCOUNT_ID>",  
          "region": "US",  
          "applicationIds": ["<APP_ID_1>", "<APP_ID_2>"],  
          "entityGuids": ["<ENTITY_GUID_1>"],  
          "alertPolicyIds": ["<POLICY_ID_1>"]  
        }  
      }  
    }  
  }'
```

```
}' \  
--region <REGION>
```

Simpan yang dikembalikan<SERVICE_ID>, lalu kaitkan New Relic:

```
aws devops-agent associate-service \  
--agent-space-id <AGENT_SPACE_ID> \  
--service-id <SERVICE_ID> \  
--configuration '{  
  "mcpservernewrelic": {  
    "accountId": "<YOUR_ACCOUNT_ID>",  
    "endpoint": "https://mcp.newrelic.com/mcp/"  
  }  
}' \  
--region <REGION>
```

Responsnya mencakup informasi webhook untuk integrasi. Anda dapat menggunakan webhook ini untuk memicu penyelidikan dari New Relic. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan Relik Baru”](#).

10. (Opsional) Daftar dan asosiasikan Datadog

Anda harus terlebih dahulu mendaftarkan Datadog melalui konsol AWS DevOps Agen dengan menggunakan OAuth alur sebelum Anda dapat mengaitkannya melalui CLI. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan DataDog”](#).

Daftar layanan terdaftar:

```
aws devops-agent list-services \  
--region <REGION>
```

Simpan <SERVICE_ID> untuk ServiceType:. mcpserverdatadog

Kemudian kaitkan Datadog:

```
aws devops-agent associate-service \  
--agent-space-id <AGENT_SPACE_ID> \  
--service-id <SERVICE_ID> \  
--configuration '{  
  "mcpserverdatadog": {  
    "name": "Datadog-MCP-Server",  
    "endpoint": "<DATADOG_MCP_ENDPOINT>"  
  }  
}'
```

```
}  
}' \  
--region <REGION>
```

Responsnya mencakup informasi webhook untuk integrasi. Anda dapat menggunakan webhook ini untuk memicu penyelidikan dari Datadog. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan DataDog”](#).

11. (Opsional) Hapus ruang agen

Menghapus ruang agen akan menghapus semua asosiasi, konfigurasi, dan data investigasi untuk ruang agen tersebut. Tindakan ini tidak dapat dibatalkan.

Untuk menghapus ruang agen, jalankan perintah berikut:

```
aws devops-agent delete-agent-space \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --region <REGION>
```

Verifikasi

Untuk memverifikasi penyiapan Anda, jalankan perintah berikut:

```
# List your agent spaces  
aws devops-agent list-agent-spaces \  
  --region <REGION>  
  
# Get details of a specific agent space  
aws devops-agent get-agent-space \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --region <REGION>  
  
# List associations for an agent space  
aws devops-agent list-associations \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --region <REGION>
```

Langkah selanjutnya

- Untuk menghubungkan integrasi tambahan, lihat [Mengkonfigurasi kemampuan untuk AWS DevOps Agen](#).

- Untuk mempelajari keterampilan dan kemampuan agen, lihat [the section called “DevOps Keterampilan Agen”](#).
- Untuk memahami aplikasi web operator, lihat [the section called “Apa itu Aplikasi Web DevOps Agen?”](#).

Catatan

- Ganti `<AGENT_SPACE_ID><MONITORING_ACCOUNT_ID>,,<EXTERNAL_ACCOUNT_ID>,<REGION>,,` dan sebagainya dengan nilai aktual Anda.
- Untuk mengetahui daftar Wilayah yang didukung, lihat [the section called “Wilayah yang Didukung”](#).

Menciptakan lingkungan pengujian

Panduan ini menyediakan pengujian langsung untuk memvalidasi fungsionalitas respons insiden AWS DevOps Agen menggunakan arsitektur sampel. Gunakan suplemen ini jika Anda ingin menguji DevOps Agen sebelum menghubungkan sistem produksi Anda.

Prasyarat

- AWS akun dengan akses administratif
- AWS DevOps Ruang Agen dibuat dengan dan dikonfigurasi menggunakan alur peran Auto create DevOps Agent
- Untuk pengujian EC2: VPC yang ada dengan setidaknya satu subnet di wilayah tempat Anda akan menerapkan.

Ikhtisar biaya dan keamanan

Perlindungan biaya

- Tes EC2: GRATIS (Tingkat AWS Gratis) atau ~ \$0,02 selama 2 jam
- Tes Lambda: GRATIS (tingkat gratis 1M requests/month)
- CloudWatch: GRATIS (10 alarm, termasuk metrik dasar)
- Perkiraan biaya total yang diharapkan: \$0,00 - \$0,05 untuk pengujian lengkap

Fitur keamanan dalam tes ini

- Auto-termination: shutdown Built-in otomatis
- Tingkat Gratis memenuhi syarat: Menggunakan jenis instans terkecil
- Cakupan terbatas: Sumber daya uji minimal dan terisolasi
- Pembersihan mudah: Langkah-langkah konsol sederhana untuk menghapus semuanya
- Tidak ada dampak produksi: Lingkungan pengujian yang benar-benar terpisah

Siapkan AWS akun untuk pengujian

Important

Sumber daya infrastruktur perlu digunakan di AWS akun tempat Anda membuat akun cloud utama DevOps Agen Space. Wilayah spesifik tidak masalah.

1. Masuk ke AWS Konsol: <https://console.aws.amazon.com>
2. Pastikan Anda bekerja di AWS akun yang sama di mana Ruang DevOps Agen Anda berada
3. Anda dapat menggunakan wilayah mana pun untuk sumber daya pengujian Anda

Note

Pemetaan 1:1 antara akun utama DevOps Agen Anda dan sumber daya lingkungan pengujian yang Anda buat menyederhanakan penyiapan pengujian. Anda dapat dengan mudah memperluas Ruang DevOps Agen Anda untuk memasukkan akun sekunder dan mengaktifkan investigasi lintas akun.

Pilih tes Anda

Anda dapat menjalankan tes secara independen atau keduanya bersama-sama:

Opsi uji A: Uji kapasitas CPU EC2

Tujuan: Validasi kemampuan AWS DevOps Agen untuk mendeteksi dan menyelidiki masalah kinerja EC2

Perkiraan waktu: 5 menit setup +10 menit eksekusi otomatis

Kesulitan: Sepenuhnya otomatis (tidak perlu langkah manual)

Opsi uji B: Tes tingkat kesalahan Lambda

Tujuan: Validasi kemampuan AWS DevOps Agen untuk mendeteksi dan menyelidiki kesalahan fungsi Lambda

Perkiraan waktu: pengaturan 10 menit+2 menit untuk memicu

Kesulitan: Sangat mudah

Opsi uji A: Uji kapasitas CPU EC2

Langkah 1: Menyebarkan CloudFormation tumpukan untuk tes EC2

Kami akan menggunakannya CloudFormation untuk membuat sumber daya pengujian kami, yang memungkinkan AWS DevOps Agen melacak dan menyelidikinya dengan benar.

1. Arahkan ke CloudFormation:

- a. Di AWS Console, cari "CloudFormation" dan klik CloudFormation
- b. Klik Buat tumpukan> Dengan sumber daya baru (standar)

2. Unggah templat:

- a. Buat file lokal baru yang disebut `AWS-DevOpsAgent-ec2-test.yaml`
- b. Salin dan tempel CloudFormation template ini ke dalam file:

```
i.
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: ID of an existing VPC where the test instance will be launched.
  SubnetId:
    Type: AWS::EC2::Subnet::Id
    Description: ID of an existing subnet within the selected VPC. Choose a
    subnet that routes to an internet gateway if you plan to connect via SSH.
  MyIP:
    Type: String
    Description: Your current IP address for SSH access (find at https://
    whatismyipaddress.com)
    Default: '0.0.0.0/0'
```

```
Resources:
# Security Group for SSH access
TestSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS DevOps Agent beta testing security group
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref MyIP
        Description: SSH access from your IP
    Tags:
      - Key: Name
        Value: AWS-DevOpsAgent-Test-SG
      - Key: Purpose
        Value: AWS-DevOpsAgent-Testing
# Key Pair for SSH access
TestKeyPair:
  Type: AWS::EC2::KeyPair
  Properties:
    KeyName: AWS-DevOpsAgent-test-key
    KeyType: rsa
  Tags:
    - Key: Name
      Value: AWS-DevOpsAgent-Test-Key
    - Key: Purpose
      Value: AWS-DevOpsAgent-Testing
# IAM Role for Session Manager access
SSMInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ec2.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
  Tags:
    - Key: Name
```

```

        Value: AWS-DevOpsAgent-Test-SSMRole
    - Key: Purpose
      Value: AWS-DevOpsAgent-Testing
# Instance profile wrapping the SSM role
SSMInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref SSMInstanceRole
# EC2 Instance for CPU testing
TestInstance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: t3.micro
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
    KeyName: !Ref TestKeyPair
    SubnetId: !Ref SubnetId
    SecurityGroupIds:
      - !GetAtt TestSecurityGroup.GroupId
    IamInstanceProfile: !Ref SSMInstanceProfile
    InstanceInitiatedShutdownBehavior: terminate
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash
        yum update -y
        yum install -y htop

        # Create the CPU stress test script
        cat > /home/ec2-user/cpu-stress-test.sh << 'EOF'
        #!/bin/bash
        echo "Starting AWS DevOpsAgent CPU Stress Test"
        echo "Time: $(date)"
        echo "Instance: $(curl -s http://169.254.169.254/latest/meta-data/
instance-id)"
        echo ""

        # Get number of CPU cores
        CORES=$(nproc)
        echo "CPU Cores: $CORES"
        echo ""

        echo "Starting stress test (5 minutes)..."
        echo "This will generate >70% CPU usage to trigger CloudWatch alarm"

```

```
echo ""

# Create CPU load using yes command
echo "Starting CPU load processes..."
for i in $(seq 1 $CORES); do
    (yes > /dev/null) &
    CPU_PID=$!
    echo "Started CPU load process $i (PID: $CPU_PID)"
    echo $CPU_PID >> /tmp/cpu_test_pids
done

# Auto-cleanup after 5 minutes
(sleep 300 && echo "Stopping CPU load processes..." && kill $(cat /
tmp/cpu_test_pids 2>/dev/null) 2>/dev/null && rm -f /tmp/cpu_test_pids) &

echo ""
echo "CPU load processes started for 5 minutes"
echo "Check CloudWatch for alarm trigger in 3-5 minutes"
EOF

chmod +x /home/ec2-user/cpu-stress-test.sh
chown ec2-user:ec2-user /home/ec2-user/cpu-stress-test.sh

# Create auto-shutdown script (safety mechanism)
cat > /home/ec2-user/auto-shutdown.sh << 'SHUTDOWN_EOF'
#!/bin/bash
echo "Auto-shutdown scheduled for 2 hours from now: $(date)"
sleep 7200
echo "Auto-shutdown executing at: $(date)"
sudo shutdown -h now
SHUTDOWN_EOF

chmod +x /home/ec2-user/auto-shutdown.sh
nohup /home/ec2-user/auto-shutdown.sh > /home/ec2-user/auto-
shutdown.log 2>&1 &

echo "AWS DevOpsAgent test setup completed at $(date)" > /home/ec2-
user/setup-complete.txt
Tags:
- Key: Name
  Value: AWS-DevOpsAgent-Test-Instance
- Key: Purpose
  Value: AWS-DevOpsAgent-Testing
# CloudWatch Alarm for CPU utilization
```

```
CPUALarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmName: AWS-DevOpsAgent-EC2-CPU-Test
    AlarmDescription: AWS-DevOpsAgent beta test - EC2 CPU utilization alarm
    MetricName: CPUUtilization
    Namespace: AWS/EC2
    Statistic: Average
    Period: 60
    EvaluationPeriods: 1
    Threshold: 70
    ComparisonOperator: GreaterThanThreshold
    Dimensions:
      - Name: InstanceId
        Value: !Ref TestInstance
    TreatMissingData: notBreaching
Outputs:
  InstanceId:
    Description: EC2 Instance ID for testing
    Value: !Ref TestInstance

  SecurityGroupId:
    Description: Security Group ID
    Value: !GetAtt TestSecurityGroup.GroupId

  AlarmName:
    Description: CloudWatch Alarm Name
    Value: !Ref CPUALarm

  SSHCommand:
    Description: SSH command to connect to instance
    Value: !Sub 'ssh -i "AWS-DevOpsAgent-test-key.pem" ec2-user@
${TestInstance.PublicDnsName}'
```

- c. Di CloudFormation konsol, pilih Unggah file templat
 - d. Klik Pilih berkas
 - e. Pilih `AWS-DevOpsAgent-ec2-test.yaml` file
 - f. Klik Berikutnya
3. Konfigurasi tumpukan:
- a. Nama tumpukan: `AWS-DevOpsAgent-EC2-Test`
 - b. Parameter:

- i. VpcId: Pilih VPC yang ada dari dropdown.
 - ii. SubnetId: Pilih subnet dalam VPC yang Anda pilih. Untuk akses SSH, subnet harus merutekan ke gateway internet, dan instance harus memiliki alamat IPv4 publik yang terkait. Jika tidak, SSHCommand output akan kosong dan koneksi SSH tidak akan berhasil.
 - iii. MyIP: Biarkan sebagai default 0.0.0.0/0 (Anda dapat mengamankan ini nanti jika diperlukan)
- c. Klik Berikutnya
4. Konfigurasi opsi tumpukan:
- a. Tinggalkan default, klik Berikutnya
5. Tinjau dan buat:
- a. Periksa Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM
 - b. Klik Kirim
6. Tunggu penyelesaian:
- a. Pembuatan tumpukan membutuhkan waktu 3-5 menit
 - b. Status akan berubah dari CREATE_IN_PROGRESS CREATE_COMPLETE
 - c. Penting: Instans EC2 Anda sekarang menjadi bagian dari CloudFormation tumpukan yang AWS DevOpsAgent dapat melacak!

Opsional: Akses SSH aman (hanya jika Anda berencana untuk terhubung ke instans)

Lewati langkah ini jika Anda hanya ingin menjalankan pengujian otomatis

1. Temukan grup keamanan:
 - a. Di AWS Konsol, buka CloudFormation dan pilih AWS-DevOpsAgent-EC2-Test tumpukan
 - b. Buka tab Output dan salin nilai SecurityGroupId (dimulai dengan sg-)
 - c. Buka EC2 → Grup Keamanan dan tempel ID ke bilah pencarian untuk membuka grup keamanan
2. Perbarui aturan SSH:
 - a. Pilih grup keamanan → tab Aturan masuk → Edit aturan masuk
 - b. Temukan aturan SSH (port 22)
 - c. Ubah sumber dari 0.0.0.0/0 ke IP Anda: [YOUR_IP]/32
 - d. Dapatkan IP Anda dari <https://whatismyipaddress.com>

Langkah 2: Tunggu eksekusi tes otomatis

1. Eksekusi uji otomatis:

- Tes stress CPU akan secara otomatis dimulai 5 menit setelah peluncuran instance
- Tidak diperlukan intervensi manual - tunggu saja, tes berjalan sepenuhnya di latar belakang

2. Pantau tes:

- Instans melakukan booting dan menyiapkan tes secara otomatis
- Skrip akan berjalan selama 5 menit dan menghasilkan > 70% penggunaan CPU
- CloudWatch alarm harus dipicu dalam 8-10 menit total (5 menit penundaan+3-5 menit untuk alarm)

3. Opsional: Jalankan ulang manual (untuk pengujian tambahan):

- Connect ke instans Anda: konsol EC2 → → Connect **AWS-DevOpsAgent-Test-Instance** → Session Manager
- Jalankan stress test lagi: `./cpu-stress-test.sh`
- Sempurna untuk AWS DevOpsAgent respons pengujian beberapa kali

Opsi uji B: Tes tingkat kesalahan Lambda

Langkah 1: Menyebarkan CloudFormation tumpukan untuk tes Lambda

1. Arahkan ke CloudFormation:

- a. Di AWS konsol, pergi ke CloudFormation
- b. Klik Buat tumpukan → Dengan sumber daya baru (standar)

2. Unggah templat:

- a. Buat file lokal baru yang disebut `AWS-DevOpsAgent-lambda-test.yaml`
- b. Salin dan tempel CloudFormation template ini ke dalam file:

```
i.
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOpsAgent Lambda Error Test Stack'
Resources:
  # IAM Role for Lambda function
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
```

```
AssumeRolePolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
      Principal:
        Service: lambda.amazonaws.com
      Action: sts:AssumeRole
ManagedPolicyArns:
  - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
Tags:
  - Key: Name
    Value: AWS-DevOpsAgent-Lambda-Test-Role
  - Key: Purpose
    Value: AWS-DevOpsAgent-Testing
# Lambda function that generates errors
TestLambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    FunctionName: AWS-DevOpsAgent-test-lambda
    Runtime: python3.12
    Handler: index.lambda_handler
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile: |
        import json
        import random
        import time
        from datetime import datetime
        def lambda_handler(event, context):
            print(f"AWS DevOpsAgent Test Lambda - {datetime.now()}")
            print(f"Event: {json.dumps(event)}")

            # Intentionally generate errors for testing
            error_scenarios = [
                "Simulated database connection timeout",
                "Test API rate limit exceeded",
                "Intentional validation error for AWS DevOpsAgent testing"
            ]

            # Always throw an error for testing purposes
            error_message = random.choice(error_scenarios)
            print(f"Generating test error: {error_message}")

            # This will create a Lambda error that CloudWatch will detect
```

```
        raise Exception(f"AWS DevOpsAgent Test Error: {error_message}")
    Description: AWS DevOpsAgent beta test function - intentionally generates
errors
    Timeout: 30
    Tags:
      - Key: Name
        Value: AWS-DevOpsAgent-Test-Lambda
      - Key: Purpose
        Value: AWS-DevOpsAgent-Testing
# CloudWatch Alarm for Lambda errors
LambdaErrorAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmName: AWS-DevOpsAgent-Lambda-Error-Test
    AlarmDescription: AWS-DevOpsAgent beta test - Lambda error rate alarm
    MetricName: Errors
    Namespace: AWS/Lambda
    Statistic: Sum
    Period: 60
    EvaluationPeriods: 1
    Threshold: 0
    ComparisonOperator: GreaterThanThreshold
    Dimensions:
      - Name: FunctionName
        Value: !Ref TestLambdaFunction
    TreatMissingData: notBreaching
Outputs:
  LambdaFunctionName:
    Description: Lambda Function Name for testing
    Value: !Ref TestLambdaFunction

  LambdaFunctionArn:
    Description: Lambda Function ARN
    Value: !GetAtt TestLambdaFunction.Arn

  AlarmName:
    Description: CloudWatch Alarm Name
    Value: !Ref LambdaErrorAlarm

  TestCommand:
    Description: AWS CLI command to test the function
    Value: !Sub 'aws lambda invoke --function-name ${TestLambdaFunction} --
payload "{\"test\": \"AWS DevOpsAgent validation\"}" response.json'
```

- c. Di CloudFormation konsol, pilih Unggah file templat
 - d. Klik Pilih berkas
 - e. Pilih `AWS-DevOpsAgent-lambda-test.yaml` file
 - f. Klik Berikutnya
3. Konfigurasi tumpukan:
 - a. Nama tumpukan: `AWS-DevOpsAgent-Lambda-Test`
 - b. Klik Berikutnya
 4. Konfigurasi opsi tumpukan:
 - a. Tinggalkan default, klik Berikutnya
 5. Tinjau dan buat:
 - a. Periksa Saya mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM
 - b. Klik Kirim
 6. Tunggu penyelesaian:
 - a. Pembuatan tumpukan membutuhkan waktu 2-3 menit
 - b. Status akan berubah menjadi `CREATE_COMPLETE`

Langkah 2: Memicu kesalahan Lambda

1. Arahkan ke konsol Lambda:
 - a. Pergi ke konsol AWS Lambda
 - b. Temukan fungsi Anda `AWS-DevOpsAgent-test-lambda`
2. Uji fungsinya:
 - a. Klik tab Uji
 - b. Klik Buat acara baru
 - c. Nama acara: `AWS-DevOpsAgent-test-event`
 - d. Gunakan payload JSON ini:

i.

```
{
  "test": "AWS DevOpsAgent validation",
  "timestamp": "2024-01-01T00:00:00Z"
}
```

3. Menghasilkan kesalahan:

- a. Klik tombol Uji 3 kali (tunggu 10 detik di antara masing-masing)
- b. Setiap tes akan menghasilkan kesalahan yang disengaja
- c. CloudWatch alarm harus dipicu dalam 2-3 menit
- d. AWS DevOpsAgentseharusnya sekarang dapat mendeteksi alarm dengan Investigasi di aplikasi Operator yang akan Anda atur selanjutnya.

Validasi AWS DevOps Deteksi agen

Langkah 1: CloudWatch Alarm periksa kewarasan (opsional)

Langkah ini untuk memastikan bahwa tes di atas sekarang dalam keadaan alarm.

Untuk Tes EC2:

- Di CloudWatch konsol, buka Alarm
- Tunggu 3-5 menit setelah memulai stress test
- Alarm Anda harus ditampilkan Dalam keadaan alarm
- Jika masih "OK": Tunggu 2-3 menit lagi (CloudWatch metrik dapat ditunda)

Untuk Tes Lambda:

- Periksa `AWS-DevOpsAgent-Lambda-Error-Test` alarm
- Harus ditampilkan Dalam alarm dalam 2-3 menit setelah menjalankan tes

Langkah 2: Mulai AWS DevOps Investigasi Agen

1. Buka AWS DevOps Agen Anda AgentSpace
2. Klik Akses admin. Ini akan membuka aplikasi web DevOps Agent Space di jendela baru
3. Klik tombol Mulai Investigasi di sisi kanan layar
4. Lengkapi formulir berikut:
 - a. Detail investigasi: Jelaskan investigasi yang ingin Anda jalankan. Sertakan detail apa pun yang Anda bisa tentang tujuan investigasi, area untuk dijelajahi, atau informasi yang relevan.

- b. Titik awal investigasi: Jelaskan informasi yang ingin Anda mulai penyelidikan. Anda dapat menyebutkan alarm, metrik, cuplikan log, atau apa pun untuk memberi DevOps Agen titik awal untuk bekerja. Dalam hal ini, berikan ringkasan alarm yang baru saja Anda buat.
 - c. Tanggal dan waktu kejadian (ISO 8601 lebih disukai): YYYY-MM-DDTHH:MMZ
 - d. Beri nama investigasi Anda: contoh: `0ncall_investigation_1:2025-10-27`
 - e. AWS ID akun untuk insiden tersebut
 - f. Wilayah tempat kejadian itu terjadi
 - g. Prioritas - AWS DevOpsAgent memungkinkan dua investigasi bersamaan. Prioritas memungkinkan Anda untuk menentukan urutan pelaksanaan investigasi Anda.
5. Klik Selidiki untuk meluncurkan penyelidikan.
 6. Klik Investigasi Anda yang tercantum di dasbor. Anda akan dibawa ke layar Detail Investigasi di mana Anda dapat melihat langkah-langkah terperinci yang diambil DevOps Agen.

Hasil yang Diharapkan

Hasil tes EC2:

- Mendeteksi alarm CPU EC2
- Mengidentifikasi akar penyebab: “beban kerja pengujian stres CPU”
- Menampilkan garis waktu: Uji stres → Lonjakan CPU → Alarm
- Memberikan rekomendasi untuk pemantauan dan penskalaan

Hasil tes Lambda:

- Mendeteksi lonjakan tingkat kesalahan Lambda
- Mengidentifikasi akar penyebab: “Pengecualian uji yang disengaja”
- Menampilkan garis waktu: Pemanggilan fungsi → Kesalahan → Alarm
- Memberikan rekomendasi untuk penanganan dan pemantauan kesalahan

Instruksi pembersihan

Tes pembersihan A (tes EC2)

Pembersihan otomatis

- Instance akan otomatis berakhir setelah 2 jam (dibangun ke dalam CloudFormation template)

Pembersihan manual (segera)

1. Hapus CloudFormation Stack:

- a. Pergi ke CloudFormation konsol
- b. Pilih AWS-DevOpsAgent-EC2-Test tumpukan
- c. Klik Hapus
- d. Konfirmasikan penghapusan
- e. Ini akan secara otomatis menghapus semua sumber daya: instans EC2, grup keamanan, key pair, dan alarm CloudWatch

Tes pembersihan B (tes Lambda)

1. Hapus CloudFormation Stack:

- a. Pergi ke CloudFormation konsol
- b. Pilih AWS-DevOpsAgent-Lambda-Test tumpukan
- c. Klik Hapus
- d. Konfirmasikan penghapusan
- e. Ini akan secara otomatis menghapus semua sumber daya: Fungsi Lambda, peran IAM, dan alarm CloudWatch

Pemecahan masalah

Masalah umum

“Tidak dapat terhubung ke instans EC2”

- **Periksa Grup Keamanan:** Pastikan SSH (port 22) terbuka untuk IP Anda

- Periksa Izin Kunci: Jalankan `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Verifikasi IP Publik: Instance harus memiliki IP publik yang ditetapkan
- Tunggu Instance: Pastikan instance dalam status “Running”

“Alarm tidak memicu”

- Tunggu Metrik: CloudWatch metrik dapat memakan waktu 2-5 menit untuk muncul
- Periksa CPU Load: SSH ke instance dan jalankan `top` untuk memverifikasi `CPU > 70%`
- Verifikasi Uji Stres: Jalankan `ps aux | grep yes` untuk melihat apakah proses pemuatan sedang berjalan
- Tunggu Diperpanjang: Terkadang membutuhkan waktu hingga 7-8 menit untuk pemacu alarm pertama

Uji validasi

Pengujian AWS DevOp Agen Anda berhasil ketika:

Validasi teknis

- Akurasi Investigasi: Hasil tes EC2 harus menunjukkan dengan benar bahwa alarm dipicu karena beban CPU. Hasil tes Lambda harus menunjukkan bahwa ini adalah kegagalan yang disengaja.
- Akurasi Garis Waktu: Urutan peristiwa yang ditampilkan dengan benar
- Kualitas Rekomendasi: Saran yang dapat ditindaklanjuti disediakan

Memulai AWS DevOps Agen menggunakan AWS CDK

Ikhtisar

Panduan ini menunjukkan cara menggunakan AWS Cloud Development Kit (AWS CDK) untuk membuat dan menyebarkan sumber daya AWS DevOps Agen. Aplikasi AWS CDK mengotomatiskan pembuatan ruang agen, peran AWS Identity and Access Management (IAM) and Access Management (IAM), aplikasi operator, dan asosiasi akun. AWS AWS CloudFormation

Pendekatan AWS CDK mengotomatiskan langkah-langkah manual yang dijelaskan dalam [panduan orientasi CLI](#) dengan mendefinisikan semua sumber daya yang diperlukan sebagai infrastruktur sebagai kode.

AWS DevOps Agen tersedia di 6 AWS Wilayah berikut: AS Timur (Virginia N.), AS Barat (Oregon), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), dan Eropa (Irlandia). Untuk informasi selengkapnya tentang Wilayah yang didukung, lihat [the section called “Wilayah yang Didukung”](#).

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki yang berikut:

- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi dengan kredensi yang sesuai
- Node.js versi 18 atau yang lebih baru
- AWS Antarmuka baris perintah CDK (CLI) diinstal secara global. Untuk menginstal AWS CDK CLI, jalankan perintah berikut:

```
npm install -g aws-cdk
```

- Satu AWS akun untuk akun pemantauan (primer)
- (Opsional) AWS Akun kedua jika Anda ingin mengatur pemantauan lintas akun

Apa yang dicakup oleh panduan ini

Panduan ini dibagi menjadi dua bagian:

- Bagian 1 — Menyebarkan ruang agen dengan aplikasi operator dan AWS asosiasi di akun pemantauan Anda. Setelah Anda menyelesaikan bagian ini, agen dapat memantau masalah di akun itu.
- Bagian 2 (Opsional) - Tambahkan AWS asosiasi sumber untuk akun layanan dan gunakan peran IAM lintas akun ke akun itu. Konfigurasi ini memungkinkan ruang agen untuk memantau sumber daya di seluruh akun.

Sumber daya dibuat

Bagian 1: DevOpsAgentStack (akun pemantauan)

- Peran IAM (DevOpsAgentRole-AgentSpace) — Diasumsikan oleh layanan DevOps Agen untuk memantau akun. Termasuk kebijakan AIDevOpsAgentAccessPolicy terkelola dan kebijakan inline yang memungkinkan pembuatan peran terkait layanan Resource Explorer.

- Peran IAM (DevOpsAgentRole-WebappAdmin) — Peran aplikasi operator dengan kebijakan AIDevOpsOperatorAppAccessPolicy terkelola untuk operasi agen.
- Ruang agen (MyCDKAgentSpace) — Ruang agen pusat, dibuat dengan menggunakan AWS::DevOpsAgent::AgentSpace CloudFormation sumber daya. Termasuk konfigurasi aplikasi operator.
- Asosiasi (AWS monitor) — Menautkan akun pemantauan ke ruang agen dengan menggunakan AWS::DevOpsAgent::Association CloudFormation sumber daya.
- Asosiasi (AWS sumber) — (Opsional) Menautkan akun layanan ke ruang agen untuk pemantauan lintas akun.

Bagian 2: ServiceStack (akun layanan, opsional)

- Peran IAM (DevOpsAgentRole-SecondaryAccount) - Peran lintas akun dengan nama tetap. Dipercaya oleh ruang agen di akun pemantauan. Termasuk kebijakan AIDevOpsAgentAccessPolicy terkelola dan kebijakan inline yang memungkinkan pembuatan peran terkait layanan Resource Explorer.
- Fungsi Lambda (echo-service) - Layanan contoh sederhana yang menggemakan peristiwa masukan kembali.

Pengaturan

Langkah 1: Kloning repositori sampel

Jalankan perintah berikut untuk mengkloning repositori dan ubah ke direktori proyek:

```
git clone https://github.com/aws-samples/sample-aws-devops-agent-cdk.git
cd sample-aws-devops-agent-cdk
```

Langkah 2: Instal dependensi

Jalankan perintah berikut untuk menginstal dependensi proyek:

```
npm install
```

Bagian 1: Menyebarkan ruang agen

Di bagian ini, Anda membuat ruang agen, peran IAM, aplikasi operator, dan AWS asosiasi di akun pemantauan Anda.

Langkah 1: Konfigurasi ID akun pemantauan

Buka `lib/constants.ts` dan atur ID akun pemantauan Anda:

Contoh berikut menunjukkan konstanta untuk memperbarui:

```
export const MONITORING_ACCOUNT_ID = "<YOUR_MONITORING_ACCOUNT_ID>";
```

Langkah 2: Bootstrap lingkungan AWS CDK

Jika Anda belum melakukan bootstrap AWS CDK di akun pemantauan Anda, jalankan perintah berikut:

```
cdk bootstrap aws://<MONITORING_ACCOUNT_ID>/<REGION> --profile monitoring
```

Langkah 3: Bangun dan terapkan

Jalankan perintah berikut untuk membangun TypeScript kode dan menyebarkan tumpukan:

```
npm run build
cdk deploy DevOpsAgentStack --profile monitoring
```

Langkah 4: Rekam output tumpukan

Setelah penerapan selesai, AWS CDK mencetak output tumpukan. Catat nilai-nilai ini untuk digunakan nanti.

Contoh berikut menunjukkan output yang diharapkan:

```
Outputs:
DevOpsAgentStack.AgentSpaceArn = arn:aws:aidevops:<REGION>:123456789012:agentspace/
abc123
DevOpsAgentStack.AgentSpaceRoleArn = arn:aws:iam::123456789012:role/DevOpsAgentRole-
AgentSpace
```

```
DevOpsAgentStack.OperatorRoleArn = arn:aws:iam::123456789012:role/DevOpsAgentRole-WebappAdmin
DevOpsAgentStack.AssociationId = assoc-xyz
```

Jika Anda berencana untuk menyelesaikan Bagian 2, simpan `AgentSpaceArn` nilainya. Anda memerlukannya untuk mengkonfigurasi tumpukan akun layanan.

Langkah 5: Verifikasi penyebaran

Untuk memverifikasi bahwa ruang agen berhasil dibuat, jalankan perintah AWS CLI berikut:

```
aws devopsagent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

Pada titik ini, ruang agen Anda digunakan dengan aplikasi operator diaktifkan dan akun pemantauan Anda terkait. Agen dapat memantau masalah di akun ini.

Bagian 2 (Opsional): Tambahkan pemantauan lintas akun

Di bagian ini, Anda memperpanjang pengaturan sehingga ruang agen Anda dapat memantau sumber daya di AWS akun kedua (akun layanan). Ini melibatkan dua tindakan:

1. Menambahkan AWS asosiasi sumber di `DevOpsAgentStack` yang menunjuk ke akun layanan.
2. Menyebarkan `ServiceStack` ke akun layanan dengan peran IAM yang mempercayai ruang agen.

Important

Anda harus menyelesaikan Bagian 1 sebelum melanjutkan. `ServiceStack` Membutuhkan `AgentSpaceArn` dari output `DevOpsAgentStack` penyebaran.

Langkah 1: Konfigurasi ID akun layanan

Buka `lib/constants.ts` dan atur ID akun layanan Anda:

Contoh berikut menunjukkan konstanta untuk memperbarui:

```
export const SERVICE_ACCOUNT_ID = "<YOUR_SERVICE_ACCOUNT_ID>";
```

DevOpsAgentStack Membuat AWS asosiasi sumber dengan menggunakan ID akun ini. Jika Anda menerapkan DevOpsAgentStack sebelum menyetel nilai ini, gunakan ulang untuk membuat asosiasi:

Jalankan perintah berikut untuk menerapkan ulang:

```
npm run build
cdk deploy DevOpsAgentStack --profile monitoring
```

Langkah 2: Atur ruang agen ARN

Salin AgentSpaceArn nilai dari DevOpsAgentStack output (Bagian 1, Langkah 4) dan atur di `lib/constants.ts`:

Contoh berikut menunjukkan konstanta untuk memperbarui:

```
export const AGENT_SPACE_ARN =
  "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/<SPACE_ID>";
```

ServiceStack Menggunakan nilai ini untuk cakupan kebijakan kepercayaan pada peran akun sekunder. Hanya ServiceStack disintesis ketika nilai ini ditetapkan.

Langkah 3: Bootstrap akun layanan

Jika Anda belum melakukan bootstrap AWS CDK di akun layanan Anda, jalankan perintah berikut:

```
cdk bootstrap aws://<SERVICE_ACCOUNT_ID>/<REGION> --profile service
```

Langkah 4: Menyebarkan ServiceStack

Jalankan perintah berikut untuk membangun dan menerapkan ServiceStack dengan menggunakan kredensial untuk akun layanan:

```
npm run build
cdk deploy ServiceStack --profile service
```

Ini menciptakan sumber daya berikut di akun layanan:

- Peran IAM (DevOpsAgentRole-SecondaryAccount) yang mempercayai ruang agen di akun pemantauan
- Sebuah echo Lambda function `echo-service ()` sebagai contoh layanan

Langkah 5: Verifikasi penyebaran

Untuk mengonfirmasi bahwa fungsi Lambda berhasil diterapkan, jalankan perintah berikut untuk menguji layanan echo:

```
aws lambda invoke \  
  --function-name echo-service \  
  --payload '{"test": "hello world"}' \  
  --profile service \  
  response.json  
cat response.json
```

Pemecahan masalah

Bagian ini menjelaskan masalah umum dan cara mengatasinya.

CloudFormation jenis sumber daya tidak ditemukan

- Verifikasi bahwa Anda menerapkan di file. [the section called “Wilayah yang Didukung”](#)
- Konfirmasikan bahwa AWS CLI Anda dikonfigurasi dengan izin yang sesuai.

Pembuatan peran IAM gagal

- Verifikasi bahwa peran penerapan Anda memiliki izin untuk membuat peran IAM.
- Periksa apakah ketentuan kebijakan kepercayaan sesuai dengan ID akun Anda.

Penerapan lintas akun gagal dengan “Tidak dapat mengambil peran dalam akun target”

- Setiap tumpukan harus digunakan dengan kredensial untuk akun target. Gunakan `--profile` bendera untuk menentukan profil AWS CLI yang benar.
- Verifikasi bahwa AWS CDK telah di-bootstrap di akun target.

Penundaan propagasi IAM

- Perubahan peran IAM dapat memakan waktu beberapa menit untuk disebarkan. Jika pembuatan ruang agen gagal segera setelah pembuatan peran, tunggu beberapa menit dan gunakan kembali.

Pembersihan

Untuk menghapus semua sumber daya, hancurkan tumpukan dalam urutan terbalik.

Jalankan perintah berikut untuk menghancurkan tumpukan:

```
# If you deployed the ServiceStack, destroy it first
cdk destroy ServiceStack --profile service
# Then destroy the DevOpsAgentStack
cdk destroy DevOpsAgentStack --profile monitoring
```

Peringatan: Tindakan ini secara permanen menghapus ruang agen Anda dan semua data terkait. Tindakan ini tidak dapat dibatalkan. Pastikan Anda telah mencadangkan informasi penting apa pun sebelum melanjutkan.

Pertimbangan keamanan

- Aplikasi AWS CDK membuat peran IAM dengan kebijakan kepercayaan yang hanya memungkinkan kepala `aidevops.amazonaws.com` layanan untuk mengambilnya.
- Kebijakan kepercayaan mencakup ketentuan yang membatasi akses ke AWS akun spesifik Anda dan ARN ruang agen.
- Semua kebijakan mengikuti prinsip hak istimewa paling sedikit. Tinjau dan sesuaikan kebijakan IAM berdasarkan persyaratan keamanan organisasi Anda.
- Peran lintas akun (`DevOpsAgentRole-SecondaryAccount`) menggunakan nama tetap dan dicakup ke ARN ruang agen tertentu.

Langkah selanjutnya

Setelah Anda menerapkan AWS DevOps Agen Anda dengan menggunakan AWS CDK:

1. Pelajari tentang berbagai kemampuan DevOps Agen dalam [Panduan Pengguna AWS DevOps Agen](#).
2. Pertimbangkan untuk mengintegrasikan penyebaran AWS CDK ke dalam CI/CD saluran pipa Anda untuk manajemen infrastruktur otomatis.

Sumber daya tambahan

- [AWS DevOps Panduan Pengguna Agen](#)
- [Contoh repositori CDK](#) di situs web GitHub
- [Panduan orientasi CLI](#)

Memulai dengan AWS DevOps Agen menggunakan AWS CloudFormation

Ikhtisar

Panduan ini menunjukkan cara menggunakan AWS CloudFormation templat untuk membuat dan menyebarkan sumber daya AWS DevOps Agen. Template mengotomatiskan pembuatan ruang agen, peran AWS Identity and Access Management (IAM), aplikasi operator, dan asosiasi AWS akun sebagai infrastruktur sebagai kode.

CloudFormation Pendekatan ini mengotomatiskan langkah-langkah manual yang dijelaskan dalam panduan [orientasi CLI](#) dengan mendefinisikan semua sumber daya yang diperlukan dalam templat YAMAL deklaratif.

AWS DevOps Agen tersedia di 6 AWS Wilayah berikut: AS Timur (Virginia N.), AS Barat (Oregon), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), dan Eropa (Irlandia). Untuk informasi selengkapnya tentang Wilayah yang didukung, lihat [the section called “Wilayah yang Didukung”](#).

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki yang berikut:

- AWS Command Line Interface (AWS CLI) diinstal dan dikonfigurasi dengan kredensi yang sesuai
- Izin untuk membuat peran dan tumpukan IAM CloudFormation
- Satu AWS akun untuk akun pemantauan (primer)
- (Opsional) AWS Akun kedua jika Anda ingin mengatur pemantauan lintas akun

Apa yang dicakup oleh panduan ini

Panduan ini dibagi menjadi dua bagian:

- Bagian 1 — Menyebarkan ruang agen dengan aplikasi operator dan AWS asosiasi di akun pemantauan Anda. Setelah Anda menyelesaikan bagian ini, agen dapat memantau masalah di akun itu.
- Bagian 2 (Opsional) - Menyebarkan peran IAM lintas akun ke akun sekunder dan tambahkan asosiasi sumber. AWS Konfigurasi ini memungkinkan ruang agen untuk memantau sumber daya di seluruh akun.

Bagian 1: Menyebarkan ruang agen

Di bagian ini, Anda membuat CloudFormation templat yang menyediakan ruang agen, peran IAM, aplikasi operator, dan AWS asosiasi di akun pemantauan Anda.

Langkah 1: Buat CloudFormation template

Simpan template berikut sebagai `devops-agent-stack.yaml`:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS DevOps Agent - Agent Space with IAM roles, operator app, and AWS
  association

Parameters:
  AgentSpaceName:
    Type: String
    Default: MyCloudFormationAgentSpace
    Description: Name for the agent space
  AgentSpaceDescription:
    Type: String
    Default: Agent space deployed with CloudFormation
    Description: Description for the agent space

Resources:
  # IAM role assumed by the DevOps Agent service to monitor the account
  DevOpsAgentSpaceRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: DevOpsAgentRole-AgentSpace
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
```

```

    Service: aidevops.amazonaws.com
    Action: sts:AssumeRole
    Condition:
      StringEquals:
        aws:SourceAccount: !Ref AWS::AccountId
      ArnLike:
        aws:SourceArn: !Sub arn:aws:aidevops:${AWS::Region}:
${AWS::AccountId}:agentspace/*
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
    Policies:
      - PolicyName: AllowCreateServiceLinkedRoles
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Sid: AllowCreateServiceLinkedRoles
          Effect: Allow
          Action:
            - iam:CreateServiceLinkedRole
          Resource:
            - !Sub arn:aws:iam:${AWS::AccountId}:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer

# IAM role for the operator app interface
DevOpsOperatorRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: DevOpsAgentRole-WebappAdmin
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: aidevops.amazonaws.com
          Action:
            - sts:AssumeRole
            - sts:TagSession
        Condition:
          StringEquals:
            aws:SourceAccount: !Ref AWS::AccountId
          ArnLike:
            aws:SourceArn: !Sub arn:aws:aidevops:${AWS::Region}:
${AWS::AccountId}:agentspace/*
    ManagedPolicyArns:

```

```
- arn:aws:iam::aws:policy/AIDevOpsOperatorAppAccessPolicy

# The agent space resource
AgentSpace:
  Type: AWS::DevOpsAgent::AgentSpace
  DependsOn:
    - DevOpsAgentSpaceRole
    - DevOpsOperatorRole
  Properties:
    Name: !Ref AgentSpaceName
    Description: !Ref AgentSpaceDescription
    OperatorApp:
      Iam:
        OperatorAppRoleArn: !GetAtt DevOpsOperatorRole.Arn

# Association linking the monitoring account to the agent space
MonitorAssociation:
  Type: AWS::DevOpsAgent::Association
  Properties:
    AgentSpaceId: !GetAtt AgentSpace.AgentSpaceId
    ServiceId: aws
    Configuration:
      Aws:
        AssumableRoleArn: !GetAtt DevOpsAgentSpaceRole.Arn
        AccountId: !Ref AWS::AccountId
        AccountType: monitor

Outputs:
  AgentSpaceId:
    Description: The agent space ID
    Value: !GetAtt AgentSpace.AgentSpaceId
  AgentSpaceArn:
    Description: The agent space ARN
    Value: !GetAtt AgentSpace.Arn
  AgentSpaceRoleArn:
    Description: The agent space IAM role ARN
    Value: !GetAtt DevOpsAgentSpaceRole.Arn
  OperatorRoleArn:
    Description: The operator app IAM role ARN
    Value: !GetAtt DevOpsOperatorRole.Arn
```

Langkah 2: Menyebarakan tumpukan

Jalankan perintah berikut untuk menyebarakan tumpukan. Ganti <REGION> dengan [the section called “Wilayah yang Didukung”](#) (misalnya,us-east-1).

```
aws cloudformation deploy \  
  --template-file devops-agent-stack.yaml \  
  --stack-name DevOpsAgentStack \  
  --capabilities CAPABILITY_NAMED_IAM \  
  --region <REGION>
```

Langkah 3: Rekam output tumpukan

Setelah penerapan selesai, jalankan perintah berikut untuk mengambil output stack. Catat nilai-nilai ini untuk digunakan nanti.

```
aws cloudformation describe-stacks \  
  --stack-name DevOpsAgentStack \  
  --query 'Stacks[0].Outputs' \  
  --region <REGION>
```

Contoh berikut menunjukkan output yang diharapkan:

```
[  
  {  
    "OutputKey": "AgentSpaceId",  
    "OutputValue": "abc123def456"  
  },  
  {  
    "OutputKey": "AgentSpaceArn",  
    "OutputValue": "arn:aws:aidevops:<REGION>:<ACCOUNT_ID>:agentspace/abc123def456"  
  },  
  {  
    "OutputKey": "AgentSpaceRoleArn",  
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-AgentSpace"  
  },  
  {  
    "OutputKey": "OperatorRoleArn",  
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-WebappAdmin"  
  }  
]
```

Jika Anda berencana untuk menyelesaikan Bagian 2, simpan `AgentSpaceArn` nilainya. Anda memerlukannya untuk mengonfigurasi peran lintas akun.

Langkah 4: Verifikasi penyebaran

Untuk memverifikasi bahwa ruang agen berhasil dibuat, jalankan perintah AWS CLI berikut:

```
aws devops-agent get-agent-space \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --region <REGION>
```

Pada titik ini, ruang agen Anda digunakan dengan aplikasi operator diaktifkan dan akun pemantauan Anda terkait. Agen dapat memantau masalah di akun ini.

Bagian 2 (Opsional): Tambahkan pemantauan lintas akun

Di bagian ini, Anda memperpanjang pengaturan sehingga ruang agen Anda dapat memantau sumber daya di AWS akun kedua (akun layanan). Ini melibatkan dua tindakan:

1. Menyebarkan peran IAM di akun layanan yang mempercayai ruang agen.
2. Menambahkan AWS asosiasi sumber di akun pemantauan yang menunjuk ke akun layanan.

Catatan: Anda harus menyelesaikan Bagian 1 sebelum melanjutkan. Templat akun layanan memerlukan output tumpukan `AgentSpaceArn` dari Bagian 1.

Langkah 1: Buat template akun layanan

Simpan template berikut sebagai `devops-agent-service-account.yaml`. Template ini membuat peran IAM lintas akun di akun sekunder.

```
AWSTemplateFormatVersion: '2010-09-09'  
Description: AWS DevOps Agent - Cross-account IAM role for secondary account monitoring  
  
Parameters:  
  MonitoringAccountId:  
    Type: String  
    Description: The 12-digit AWS account ID of the monitoring account  
  AgentSpaceArn:  
    Type: String  
    Description: The ARN of the agent space from the monitoring account
```

```

Resources:
  # Cross-account IAM role trusted by the agent space
  DevOpsSecondaryAccountRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: DevOpsAgentRole-SecondaryAccount
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: aidevops.amazonaws.com
            Action: sts:AssumeRole
            Condition:
              StringEquals:
                aws:SourceAccount: !Ref MonitoringAccountId
              ArnLike:
                aws:SourceArn: !Ref AgentSpaceArn
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
    Policies:
      - PolicyName: AllowCreateServiceLinkedRoles
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Sid: AllowCreateServiceLinkedRoles
              Effect: Allow
              Action:
                - iam:CreateServiceLinkedRole
              Resource:
                - !Sub arn:aws:iam::${AWS::AccountId}:role/aws-service-role/resource-explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer

Outputs:
  SecondaryAccountRoleArn:
    Description: The cross-account IAM role ARN
    Value: !GetAtt DevOpsSecondaryAccountRole.Arn

```

Langkah 2: Menyebarkan tumpukan akun layanan

Menggunakan kredensi untuk akun layanan, jalankan perintah berikut:

```
aws cloudformation deploy \
```

```
--template-file devops-agent-service-account.yaml \  
--stack-name DevOpsAgentServiceAccountStack \  
--capabilities CAPABILITY_NAMED_IAM \  
--parameter-overrides \  
  MonitoringAccountId=<MONITORING_ACCOUNT_ID> \  
  AgentSpaceArn=<AGENT_SPACE_ARN> \  
--region <REGION>
```

Langkah 3: Tambahkan AWS asosiasi sumber

Beralih kembali ke akun pemantauan dan buat AWS asosiasi sumber. Anda dapat melakukan ini dengan membuat tumpukan terpisah atau dengan memperbarui template asli. Contoh berikut menggunakan template mandiri.

Simpan template berikut sebagaidevops-agent-source-association.yaml:

```
AWSTemplateFormatVersion: '2010-09-09'  
Description: AWS DevOps Agent - Source AWS association for cross-account monitoring  
  
Parameters:  
  AgentSpaceId:  
    Type: String  
    Description: The agent space ID from the monitoring account stack  
  ServiceAccountId:  
    Type: String  
    Description: The 12-digit AWS account ID of the service account  
  ServiceAccountRoleArn:  
    Type: String  
    Description: The ARN of the DevOpsAgentRole-SecondaryAccount role in the service  
account  
  
Resources:  
  SourceAssociation:  
    Type: AWS::DevOpsAgent::Association  
    Properties:  
      AgentSpaceId: !Ref AgentSpaceId  
      ServiceId: aws  
      Configuration:  
        SourceAws:  
          AccountId: !Ref ServiceAccountId  
          AccountType: source  
          AssumableRoleArn: !Ref ServiceAccountRoleArn
```

Outputs:

```
SourceAssociationId:
  Description: The source association ID
  Value: !Ref SourceAssociation
```

Terapkan tumpukan asosiasi menggunakan kredensi akun pemantauan:

```
aws cloudformation deploy \
  --template-file devops-agent-source-association.yaml \
  --stack-name DevOpsAgentSourceAssociationStack \
  --parameter-overrides \
    AgentSpaceId=<AGENT_SPACE_ID> \
    ServiceAccountId=<SERVICE_ACCOUNT_ID> \
    ServiceAccountRoleArn=arn:aws:iam::<SERVICE_ACCOUNT_ID>:role/DevOpsAgentRole-
SecondaryAccount \
  --region <REGION>
```

Verifikasi

Verifikasi pengaturan Anda dengan menjalankan perintah AWS CLI berikut:

```
# List your agent spaces
aws devops-agent list-agent-spaces \
  --region <REGION>

# Get details of a specific agent space
aws devops-agent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>

# List associations for an agent space
aws devops-agent list-associations \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

Pemecahan masalah

Bagian ini menjelaskan masalah umum dan cara mengatasinya.

CloudFormation jenis sumber daya tidak ditemukan

- Verifikasi bahwa Anda menerapkan di file. [the section called “Wilayah yang Didukung”](#)

- Konfirmasikan bahwa AWS CLI Anda dikonfigurasi dengan izin yang sesuai.

Pembuatan peran IAM gagal

- Verifikasi bahwa kredensial penerapan Anda memiliki izin untuk membuat peran IAM dengan nama kustom (). `CAPABILITY_NAMED_IAM`
- Periksa apakah ketentuan kebijakan kepercayaan sesuai dengan ID akun Anda.

Penerapan lintas akun gagal

- Setiap tumpukan harus digunakan dengan kredensial untuk akun target. Gunakan `--profile` bendera untuk menentukan profil AWS CLI yang benar.
- Verifikasi bahwa `AgentSpaceArn` parameter cocok dengan ARN yang tepat dari output tumpukan Bagian 1.

Penundaan propagasi IAM

- Perubahan peran IAM dapat memakan waktu beberapa menit untuk disebar. Jika pembuatan ruang agen gagal segera setelah pembuatan peran, tunggu beberapa menit dan gunakan kembali.

Pembersihan

Untuk menghapus semua sumber daya, hapus tumpukan dalam urutan terbalik.

Peringatan: Tindakan ini secara permanen menghapus ruang agen Anda dan semua data terkait. Tindakan ini tidak dapat dibatalkan. Pastikan Anda telah mencadangkan informasi penting apa pun sebelum melanjutkan.

Jalankan perintah berikut untuk menghapus tumpukan:

```
# If you deployed the source association stack, delete it first
aws cloudformation delete-stack \
  --stack-name DevOpsAgentSourceAssociationStack \
  --region <REGION>

aws cloudformation wait stack-delete-complete \
  --stack-name DevOpsAgentSourceAssociationStack \
  --region <REGION>
```

```
# If you deployed the service account stack, delete it next (using service account
credentials)
aws cloudformation delete-stack \
  --stack-name DevOpsAgentServiceAccountStack \
  --region <REGION>

aws cloudformation wait stack-delete-complete \
  --stack-name DevOpsAgentServiceAccountStack \
  --region <REGION>

# Delete the main stack last
aws cloudformation delete-stack \
  --stack-name DevOpsAgentStack \
  --region <REGION>
```

Langkah selanjutnya

Setelah Anda menerapkan AWS DevOps Agen Anda dengan menggunakan AWS CloudFormation:

- Untuk menghubungkan integrasi tambahan, lihat [Mengkonfigurasi kemampuan untuk AWS DevOps Agen](#).
- Untuk mempelajari keterampilan dan kemampuan agen, lihat [the section called “DevOps Keterampilan Agen”](#).
- Untuk memahami aplikasi web operator, lihat [the section called “Apa itu Aplikasi Web DevOps Agen?”](#).

Memulai dengan AWS DevOps Agen menggunakan Terraform

Ikhtisar

Panduan ini menunjukkan cara menggunakan Terraform untuk membuat dan menerapkan AWS DevOps sumber daya Agen. Konfigurasi Terraform mengotomatiskan pembuatan ruang agen, peran IAM, aplikasi operator, dan asosiasi akun. AWS

Pendekatan Terraform mengotomatiskan langkah-langkah manual yang dijelaskan dalam [panduan orientasi CLI](#) dengan mendefinisikan semua sumber daya yang diperlukan sebagai infrastruktur sebagai kode.

AWS DevOps Agen tersedia di 6 AWS Wilayah berikut: AS Timur (Virginia N.), AS Barat (Oregon), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), dan Eropa (Irlandia). Untuk informasi selengkapnya tentang Wilayah yang didukung, lihat [the section called “Wilayah yang Didukung”](#).

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki yang berikut:

- Terraform \geq 1.0 diinstal
- AWS CLI diinstal dan dikonfigurasi dengan kredensial yang sesuai
- Satu AWS akun untuk akun pemantauan (primer)
- (Opsional) AWS Akun kedua jika Anda ingin mengatur pemantauan lintas akun

Apa yang dicakup oleh panduan ini

Panduan ini dibagi menjadi dua bagian:

- Bagian 1 — Menyebarkan ruang agen dengan aplikasi operator dan AWS asosiasi di akun pemantauan Anda. Setelah menyelesaikan bagian ini, agen dapat memantau masalah di akun itu.
- Bagian 2 (Opsional) - Tambahkan AWS asosiasi sumber untuk akun layanan dan gunakan peran IAM lintas akun ditambah Lambda gema ke akun itu. Ini memungkinkan ruang agen untuk memantau sumber daya di seluruh akun.

Sumber daya dibuat

Bagian 1: Memantau akun

- Peran IAM (DevOpsAgentRole-AgentSpace-*) — Diasumsikan oleh layanan DevOps Agen untuk memantau akun. Termasuk kebijakan AIDevOpsAgentAccessPolicy terkelola dan kebijakan inline yang memungkinkan pembuatan peran terkait layanan Resource Explorer.
- Peran IAM (DevOpsAgentRole-WebappAdmin-*) — Peran aplikasi operator dengan kebijakan AIDevOpsOperatorAppAccessPolicy terkelola untuk operasi agen.
- Ruang agen (nama yang dapat dikonfigurasi) - Ruang agen pusat, dibuat menggunakan `awssc_devopsagent_agent_space` sumber daya. Termasuk konfigurasi aplikasi operator.
- Asosiasi (AWS monitor) — Menautkan akun pemantauan ke ruang agen menggunakan `awssc_devopsagent_association` sumber daya.

- Asosiasi (AWS sumber) — (Opsional) Menautkan akun layanan ke ruang agen untuk pemantauan lintas akun.

Bagian 2: Akun layanan (opsional)

- Peran IAM (DevOpsAgentRole-SecondaryAccount-TF) - Peran lintas akun dengan nama tetap. Dipercaya oleh ruang agen di akun pemantauan. Termasuk kebijakan AIDevOpsAgentAccessPolicy terkelola dan kebijakan inline yang memungkinkan pembuatan peran terkait layanan Resource Explorer.
- Fungsi Lambda (echo-service-tf) - Layanan contoh sederhana yang menggemakan peristiwa masukan kembali.

Pengaturan

Langkah 1: Kloning repositori sampel

```
git clone https://github.com/aws-samples/sample-aws-devops-agent-terraform.git
cd sample-aws-devops-agent-terraform
```

Langkah 2: Konfigurasi variabel

Salin file variabel contoh dan sesuaikan untuk lingkungan Anda:

```
cp terraform.tfvars.example terraform.tfvars
```

Edit `terraform.tfvars` dengan nama dan deskripsi ruang agen Anda:

```
agent_space_name          = "MyCompanyAgentSpace"
agent_space_description    = "DevOps Agent Space for monitoring production workloads"
```

Bagian 1: Menyebarkan ruang agen

Di bagian ini, Anda membuat ruang agen, peran IAM, aplikasi operator, dan AWS asosiasi di akun pemantauan Anda.

Langkah 1: Terapkan dengan otomatisasi (disarankan)

Gunakan skrip penerapan yang disediakan untuk penyiapan yang disederhanakan:

```
./deploy.sh
```

Script ini secara otomatis:

- Memeriksa prasyarat (Terraform, CLI, kredensial) AWS
- Membuat `terraform.tfvars` dari contoh jika diperlukan
- Menginisialisasi, memvalidasi, merencanakan, dan menerapkan Terraform

Atau, jika Anda lebih suka kontrol manual:

```
terraform init
terraform plan
terraform apply
```

Ketik `yes` saat diminta untuk mengonfirmasi penerapan.

Langkah 2: Rekam output

Setelah penerapan selesai, Terraform mencetak output. Catat nilai-nilai ini untuk digunakan nanti:

```
Outputs:
agent_space_id           = "abc123"
agent_space_arn          =
  "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/abc123"
agent_space_name         = "MyCompanyAgentSpace"
devops_agentspace_role_arn = "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/
DevOpsAgentRole-AgentSpace-a1b2c3d4"
devops_operator_role_arn = "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/
DevOpsAgentRole-WebappAdmin-a1b2c3d4"
primary_account_id       = "<MONITORING_ACCOUNT_ID>"
primary_account_association_id = "assoc-xyz"
```

Jika Anda berencana untuk menyelesaikan Bagian 2, simpan `agent_space_arn` nilainya. Anda akan membutuhkannya untuk mengkonfigurasi sumber daya akun layanan.

Langkah 3: Verifikasi penyebaran

Jalankan skrip verifikasi pasca-penerapan:

```
./post-deploy.sh
```

Atau gunakan AWS CLI untuk memverifikasi bahwa ruang agen berhasil dibuat:

```
aws devops-agent get-agent-space \  
  --agent-space-id <AGENT_SPACE_ID> \  
  --region <REGION>
```

Pada titik ini, ruang agen Anda digunakan dengan aplikasi operator diaktifkan dan akun pemantauan Anda terkait. Agen dapat memantau masalah di akun ini.

Bagian 2 (Opsional): Tambahkan pemantauan lintas akun

Di bagian ini, Anda memperpanjang pengaturan sehingga ruang agen dapat memantau sumber daya di AWS akun kedua (akun layanan). Ini melibatkan dua tindakan:

1. Menambahkan AWS asosiasi sumber yang menunjuk ke akun layanan.
2. Menerapkan peran IAM lintas akun dan fungsi Lambda gema ke dalam akun layanan.

Important

Anda harus menyelesaikan Bagian 1 sebelum melanjutkan. Sumber daya akun layanan memerlukan output penyebaran `agent_space_arn` dari Bagian 1.

Langkah 1: Konfigurasi ID akun layanan

Diterraform.tfvars, setel ID akun layanan Anda:

```
service_account_id = "<YOUR_SERVICE_ACCOUNT_ID>"
```

Langkah 2: Atur ruang agen ARN

Salin `agent_space_arn` nilai dari output Bagian 1 (Langkah 2) dan atur diterraform.tfvars:

```
agent_space_arn = "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/  
<SPACE_ID>"
```

Sumber daya akun layanan menggunakan nilai ini untuk mencakup kebijakan kepercayaan pada peran akun sekunder. Sumber daya ini hanya dibuat ketika nilai ini ditetapkan.

Langkah 3: Konfigurasi penyedia `aws.service`

Di `main.tf`, konfigurasi alias `aws.service` penyedia dengan kredensial untuk akun layanan. Anda dapat menggunakan profil bernama atau peran asumsi:

Menggunakan profil:

```
provider "aws" {
  alias    = "service"
  region  = var.aws_region
  profile  = "your-service-account-profile"
}
```

Atau menggunakan peran asumsi:

```
provider "aws" {
  alias    = "service"
  region  = var.aws_region
  assume_role {
    role_arn = "arn:aws:iam::<SERVICE_ACCOUNT_ID>:role/OrganizationAccountAccessRole"
  }
}
```

Langkah 4: Menyebarkan

Terapkan konfigurasi yang diperbarui:

```
terraform apply
```

Ini menciptakan sumber daya berikut di akun layanan:

- Peran IAM (`DevOpsAgentRole-SecondaryAccount-TF`) yang mempercayai ruang agen di akun pemantauan
- Sebuah echo Lambda function `echo-service-tf` () sebagai contoh layanan

Ini juga menciptakan AWS asosiasi sumber di akun pemantauan yang menautkan akun layanan.

Langkah 5: Verifikasi penyebaran

Uji layanan gema untuk mengonfirmasi bahwa fungsi Lambda berhasil diterapkan:

```
aws lambda invoke \  
  --function-name echo-service-tf \  
  --payload '{"test": "hello world"}' \  
  --profile <your-service-account-profile> \  
  --region <REGION> \  
  response.json  
cat response.json
```

Pemecahan masalah

Penundaan propagasi IAM

- Konfigurasi mencakup 30 detik `time_sleep` antara pembuatan peran IAM dan pembuatan Agent Space. Layanan DevOps Agen memvalidasi kebijakan kepercayaan peran operator selama pembuatan Ruang Agen, dan ini bisa gagal jika IAM belum sepenuhnya disebar. Jika Anda masih melihat kesalahan kebijakan kepercayaan, tunggu sebentar dan jalankan `terraform apply` lagi — peran IAM sudah ada dan aplikasi akan mengambil tempat yang ditinggalkannya.

Kesalahan izin

- Verifikasi bahwa AWS kredensial Anda memiliki izin IAM yang diperlukan untuk membuat peran dan kebijakan.
- Periksa apakah ketentuan kebijakan kepercayaan sesuai dengan ID akun Anda.

Penerapan lintas akun gagal

- `aws.serviceProvider` harus dikonfigurasi dengan kredensi untuk akun layanan. Gunakan profil bernama atau blok peran asumsi.
- Verifikasi bahwa `agent_space_arn` nilainya cocok dengan ARN dari output Bagian 1.

Jenis sumber daya Terraform tidak ditemukan

- Verifikasi bahwa Anda memiliki versi `awscc` penyedia `~> 1.0` atau yang lebih baru. `awscc_devopsagent_association` Sumber daya `awscc_devopsagent_agent_space` dan sumber daya membutuhkan penyedia AWS Cloud Control.

Pembersihan

Untuk menghapus semua sumber daya, hancurkan dalam urutan terbalik jika Anda menggunakan Bagian 2:

```
./cleanup.sh
```

Atau secara manual:

```
terraform destroy
```

Peringatan: Ini menghapus ruang agen Anda dan semua data terkait secara permanen. Pastikan Anda telah mencadangkan informasi penting apa pun sebelum melanjutkan.

Pertimbangan keamanan

- Konfigurasi Terraform membuat peran IAM dengan kebijakan kepercayaan yang hanya memungkinkan kepala `aidevops.amazonaws.com` layanan untuk mengambilnya.
- Kebijakan kepercayaan mencakup ketentuan yang membatasi akses ke AWS akun spesifik Anda dan ARN ruang agen.
- Semua kebijakan mengikuti prinsip hak istimewa paling sedikit. Tinjau dan sesuaikan kebijakan IAM berdasarkan persyaratan keamanan organisasi Anda.
- Peran lintas akun (`DevOpsAgentRole-SecondaryAccount-TF`) menggunakan nama tetap dan dicakup ke ARN ruang agen tertentu.

Langkah selanjutnya

Setelah Anda menerapkan AWS DevOps Agen Anda menggunakan Terraform:

1. Pelajari tentang berbagai kemampuan DevOps Agen dalam [Panduan Pengguna AWS DevOps Agen](#).
2. Pertimbangkan untuk mengintegrasikan penerapan Terraform ke dalam CI/CD saluran pipa Anda untuk manajemen infrastruktur otomatis.

Sumber daya tambahan

- [AWS DevOps Panduan Pengguna Agen](#)
- [Contoh repositori Terraform](#)
- [Panduan orientasi CLI](#)

Bekerja dengan DevOps Agen

Bekerja dengan DevOps Agen

AWS DevOps Agen bekerja bersama tim operasi Anda di seluruh siklus hidup insiden penuh — mulai dari deteksi hingga investigasi, pemulihan, dan pencegahan. Topik berikut menjelaskan cara menggunakan DevOps Agen untuk mengelola setiap fase siklus hidup ini.

Respon insiden otonom

Ketika insiden terdeteksi — baik melalui integrasi bawaan dengan sistem tiket Anda, webhook dari alat pemantauan Anda, atau pemicu manual — DevOps Agen secara otomatis memulai penyelidikan. Agen menganalisis metrik, log, jejak, perubahan kode, dan riwayat penerapan untuk menentukan akar penyebab dan mengusulkan rencana mitigasi. Jika Anda memerlukan bantuan tambahan, Anda dapat melakukan eskalasi langsung ke AWS Support dari aplikasi web DevOps Agent Space, yang secara otomatis membagikan konteks investigasi dengan teknisi dukungan sehingga Anda tidak perlu mengulangi apa yang sudah ditemukan agen. Untuk informasi selengkapnya, lihat [the section called “Respon insiden otonom”](#).

Tugas sesuai permintaan DevOps

Kapan saja selama siklus hidup insiden, Anda dapat berinteraksi dengan DevOps Agen melalui antarmuka obrolan percakapan. Ajukan pertanyaan tentang AWS sumber daya Anda, kesehatan sistem, status alarm, dan riwayat penyebaran menggunakan bahasa alami. Obrolan sadar konteks — saat Anda melihat penyelidikan tertentu, Anda dapat mengarahkan agen untuk mengeksplorasi hipotesis tertentu, fokus pada log tertentu, atau memperbarui analisis akar penyebabnya. Anda juga dapat menanyakan konfigurasi sumber daya, tren kesalahan, dan wawasan investigasi di seluruh lingkungan tanpa menavigasi antar konsol. Untuk informasi selengkapnya, lihat [the section called “DevOps Tugas Sesuai Permintaan”](#).

Pencegahan insiden proaktif

Setelah menyelesaikan insiden, DevOps Agen menganalisis pola di seluruh riwayat investigasi Anda untuk menghasilkan rekomendasi yang mencegah insiden di masa depan dan mengurangi waktu rata-rata untuk mendeteksi. Rekomendasi mencakup empat area: postur observabilitas, kesenjangan pengujian, perubahan kode, dan arsitektur infrastruktur. Agen menjalankan evaluasi setiap minggu

dan memperbarui rekomendasi saat insiden baru terjadi. Anda dapat menerima, menolak, atau melacak rekomendasi, dan agen belajar dari umpan balik Anda untuk menyempurnakan saran masa depan. Untuk informasi selengkapnya, lihat [the section called “Pencegahan insiden proaktif”](#).

Interface dengan Agen DevOps

AWS DevOps Agen mendukung beberapa metode akses termasuk konsol aplikasi web, integrasi MCP untuk IDEs, Agent Client Protocol (ACP), webhook untuk otomatisasi berbasis peristiwa, dan akses API langsung. Lihat informasi yang lebih lengkap di [the section called “Interface dengan Agen DevOps”](#).

Respon insiden otonom

Memulai Investigasi

Investigasi respons insiden dapat dimulai dengan salah satu dari tiga cara.

- Built-in integrasi - Anda dapat menghubungkan Ruang DevOps Agen ke sistem tiket seperti ServiceNow menggunakan integrasi bawaan. Setelah terhubung, investigasi respons insiden DevOps Agen akan secara otomatis dipicu dari tiket dukungan, dan DevOps Agen Anda akan memberikan pembaruan temuan utamanya, analisis akar penyebab, dan rencana mitigasi ke dalam tiket asal.
- Webhooks - Anda dapat menggunakan webhook untuk mengirim acara ke Agen. AWS DevOps Misalnya Anda dapat menggunakan webhook untuk memicu investigasi respons insiden dari tiket PagerDuty atau alarm Grafana.
- Secara manual - Anda dapat memulai investigasi respons insiden secara manual dari tab Respons Insiden di aplikasi web DevOps Agent Space mana pun. Anda dapat memasukkan teks formulir gratis yang menjelaskan insiden yang Anda ingin DevOps Agen Anda selidiki, dan itu akan membuat rencana investigasi, mengumpulkan temuan, menentukan akar penyebab, dan menawarkan untuk menghasilkan rencana mitigasi. Anda juga dapat memilih dari beberapa titik awal yang telah dikonfigurasi sebelumnya untuk memulai Investigasi Anda dengan cepat: Alarm terbaru untuk menyelidiki alarm terpicu terbaru Anda dan menganalisis metrik dan log yang mendasarinya untuk menentukan akar penyebabnya, Penggunaan CPU yang tinggi untuk menyelidiki metrik pemanfaatan CPU yang tinggi di seluruh sumber daya komputasi Anda dan mengidentifikasi proses atau layanan mana yang menghabiskan sumber daya yang berlebihan, atau lonjakan tingkat kesalahan untuk menyelidiki peningkatan terbaru dalam tingkat kesalahan aplikasi dengan menganalisis metrik, log aplikasi, dan mengidentifikasi sumber kegagalan.

Incident Response Dashboard

Start an investigation

Describe the investigation you'd like to run. Include any details you can about the investigation goals, areas, to explore, or relevant information.

Latest alarm

High CPU usage

Error rate spike

Start Investigation

Setelah Anda mengklik “Mulai Investigasi”, Anda akan diminta untuk memberikan beberapa detail tambahan untuk membantu agen memfokuskan pekerjaannya. Dialog investigasi mencakup bidang-bidang berikut:

- Detail investigasi — Pre-filled dengan deskripsi Anda. Anda dapat mengedit ini untuk menyempurnakan ruang lingkup investigasi.
- Titik awal investigasi — Secara opsional menggambarkan alarm tertentu, metrik, cuplikan log, atau titik awal lainnya untuk agen.
- Tanggal dan waktu kejadian — Auto-filled dengan waktu saat ini dalam format UTC. Sesuaikan jika kejadian itu terjadi sebelumnya.
- Beri nama investigasi Anda — Auto-generated dengan stempel waktu. Anda dapat menyesuaikan ini (maksimum 400 karakter).
- Prioritas - Pilih prioritas investigasi dari dropdown (Medium adalah default).

Tinjau dan sesuaikan bidang-bidang ini sesuai kebutuhan, lalu klik “Mulai menyelidiki...” untuk memulai. Anda kemudian akan dibawa ke halaman detail investigasi di mana Anda dapat melihat DevOps Agen Anda beraksi!

Triase insiden

Fase triase adalah tahap pertama dari sistem respons insiden AWS DevOps Agen. Ketika peristiwa eksternal memicu, seperti alarm dari Datadog, tiket insiden dari, atau masalah dari Dynatrace ServiceNow, AWS DevOps Agen secara otomatis memprosesnya dalam hitungan detik untuk menentukan apakah itu harus diselidiki secara independen atau terkait dengan penyelidikan yang ada.

Fungsi utama dari tahap triase adalah korelasi insiden - mengidentifikasi insiden terkait dan mengkonsolidasikannya ke dalam penyelidikan tunggal untuk menghindari duplikat pekerjaan dan pemborosan sumber daya. Ketika insiden baru tiba, AWS DevOps Agen menganalisisnya bersama penyelidikan aktif dalam jendela lihat ke belakang (biasanya 20 menit). Menggunakan AI-powered analisis, ia memeriksa faktor-faktor seperti kesamaan komponen, wilayah geografis, dan pola waktu untuk menentukan hubungan antara insiden.

AWS DevOps Agen membuat salah satu dari tiga keputusan:

- **Terkait** — Mengkorelasikan insiden tersebut dengan penyelidikan yang ada dan mengirimkan pesan pengarah ke penyelidikan itu dengan konteks tentang insiden baru tersebut.
- **Dilewati** — Insiden ini cocok dengan kriteria lewati yang ditentukan dalam keterampilan dan secara otomatis diberhentikan tanpa penyelidikan. Untuk informasi selengkapnya, lihat [the section called “DevOps Keterampilan Agen”](#).
- **Proceed** — Menjadwalkan penyelidikan independen baru untuk insiden tersebut.

Melihat keputusan triase

Ketika insiden terkait, penyelidikan utama menerima pesan kemudi yang berisi rincian insiden terkait dan alasan korelasi. Di aplikasi web AWS DevOps Agent Space, Anda akan melihat status LINKED bersama dengan alasan korelasi yang menjelaskan mengapa insiden ditautkan. Investigasi utama menampilkan daftar semua insiden terkait, memungkinkan Anda untuk melihat cakupan penuh masalah terkait yang sedang diselidiki bersama. Sistem tiket eksternal Anda (ServiceNow, PagerDuty, dll.) dan saluran komunikasi (Slack) akan menerima pemberitahuan bahwa insiden itu terkait dengan alasan korelasi.

Ketika insiden dilewati, aplikasi web AWS DevOps Agent Space menampilkan status SKIPPED bersama dengan alasan menjelaskan mengapa insiden itu disaring. Sistem tiket eksternal dan saluran komunikasi Anda juga menerima pemberitahuan bahwa insiden itu dilewati bersama dengan alasan lewati.

Memperbaiki keputusan triase

Jika AWS DevOps Agen salah menautkan insiden, Anda dapat memutuskan tautannya secara manual melalui aplikasi web AWS DevOps Agent Space. Ini menjadwalkan ulang insiden yang tidak terkait sebagai penyelidikan independen. Anda juga dapat memberikan aturan korelasi khusus dengan membuat Keterampilan AWS DevOps Agen yang berisi logika korelasi Anda dan mengaitkannya dengan tahap triase.

Jika AWS DevOps Agen salah melewati insiden, Anda dapat melepaskannya secara manual melalui aplikasi web AWS DevOps Agent Space. Ini menjadwalkan ulang insiden untuk diselidiki. Untuk menyesuaikan insiden mana yang dilewati, modifikasi atau nonaktifkan keterampilan yang menentukan kriteria lewati.

Minta dukungan manusia

AWS DevOps Agen dapat terhubung langsung dengan AWS Support untuk merampingkan proses respons insiden Anda. Bila Anda membutuhkan bantuan tambahan dari AWS Support, dari aplikasi web DevOps Agent Space Anda dapat membuat kasus dukungan yang secara otomatis berbagi konteks investigasi dengan teknisi AWS Support, sehingga mengurangi waktu yang dibutuhkan untuk menjelaskan masalah Anda.

Cara kerjanya

Saat menyelidiki suatu insiden, AWS DevOps Agen membuat log komprehensif analisisnya, termasuk:

- Temuan investigasi akar penyebab
- Metrik, log, dan jejak dianalisis
- Perubahan kode dan riwayat penerapan ditinjau
- Tindakan remediasi direkomendasikan
- Garis waktu peristiwa dan perilaku sistem

Anda dapat meningkatkan penyelidikan Anda ke AWS Support langsung dari aplikasi web AWS DevOps Agent Space. Ketika Anda melakukannya, AWS DevOps Agen secara otomatis meneruskan log penyelidikannya ke AWS Support, memberikan konteks lengkap kepada teknisi dukungan tentang penyelidikan Anda tanpa mengharuskan Anda mengumpulkan dan menjelaskan detailnya secara manual.

Mengobrol dengan AWS Support

Setelah membuat kasus dukungan, Anda dapat berkomunikasi dengan AWS Support di jendela obrolan terpisah di dalam aplikasi web AWS DevOps Agent Space Anda. Hal ini memungkinkan Anda untuk:

- Diskusikan masalah Anda dengan teknisi AWS Support di samping garis waktu investigasi AWS DevOps Agen Anda
- Lihat analisis otomatis AWS DevOps Agen dan panduan ahli AWS Dukungan di antarmuka yang sama
- Bagikan informasi atau klarifikasi tambahan dengan mulus sesuai kebutuhan

Pengalaman obrolan membuat penyelidikan AWS DevOps Agen dan percakapan AWS Support Anda mudah diakses, memungkinkan kolaborasi dan resolusi yang lebih cepat.

Bahasa kasus Support

Saat Anda membuat kasus dukungan melalui AWS DevOps Agen, kasing secara otomatis dibuat dalam bahasa yang dikonfigurasi dalam setelan bahasa respons Agen Ruang Agen Anda. Ini memastikan bahwa kasus dukungan Anda diarahkan ke insinyur dukungan yang berbicara bahasa pilihan Anda.

Misalnya, jika bahasa Agen Ruang Anda diatur ke bahasa Jepang, kasus dukungan Anda akan diarahkan ke insinyur Japanese-speaking dukungan. Jika tidak ada bahasa yang dikonfigurasi, atau jika bahasa yang dikonfigurasi tidak didukung oleh AWS Support untuk kategori kasus yang dipilih, kasus default ke bahasa Inggris.

AWS Support saat ini mendukung bahasa berikut untuk perutean kasus: Mandarin, Inggris, Prancis, Jepang, Korea, Portugis, dan Spanyol. Untuk mengubah bahasa yang digunakan untuk kasus dukungan, perbarui setelan bahasa respons Agen dalam konfigurasi Ruang Agen Anda. Untuk informasi selengkapnya, lihat [the section called “Membuat Ruang Agen”](#).

Persyaratan rencana Support

Kemampuan Anda untuk membuat dan berinteraksi dengan kasus dukungan melalui AWS DevOps Agen bergantung pada paket AWS Support Anda. Silakan lihat [panduan pengguna Support Plans](#) untuk mempelajari lebih lanjut tentang hak Anda.

Catatan Pelanggan Dukungan Dasar tidak dapat membuat kasus dukungan teknis dan oleh karena itu tidak dapat meningkatkan investigasi AWS DevOps Agen untuk Support AWS Developer Support, pelanggan dapat membuat kasus melalui AWS DevOps Agen, tetapi harus mengunjungi [Pusat AWS Dukungan](#) untuk berkorespondensi dengan teknisi Support, karena Dukungan Pengembang tidak menyertakan dukungan berbasis obrolan. Semua paket lain dapat menggunakan pengalaman obrolan terintegrasi dalam Agen. AWS DevOps Untuk detail selengkapnya tentang hak paket dukungan, termasuk waktu respons dan tingkat keparahan kasus yang tersedia, lihat Panduan Pengguna [AWS Support Plans](#).

Informasi apa yang dibagikan AWS Support

Saat Anda membuat kasus dukungan dari aplikasi web AWS DevOps Agent Space, informasi berikut akan dibagikan secara otomatis dengan AWS Support:

- Garis waktu investigasi: Catatan kronologis analisis AWS DevOps Agen
- Informasi sumber daya: AWS Sumber daya yang terpengaruh
- Data observabilitas: Metrik, log, dan jejak yang relevan dari alat pemantauan terintegrasi Anda
- Perubahan terbaru: Penerapan kode, perubahan infrastruktur, dan pembaruan konfigurasi
- Upaya Remediasi: AWS DevOps Agen Tindakan direkomendasikan
- Penilaian dampak: Ruang lingkup dan tingkat keparahan insiden

Semua data yang dibagikan dengan AWS Support mengikuti konfigurasi residensi dan keamanan AWS data yang ada. AWS DevOps Agen hanya membagikan informasi yang terkait dengan penyelidikan spesifik Anda dan menghormati kebijakan tata kelola data organisasi Anda.

Memulai

Untuk menggunakan integrasi AWS Support AWS DevOps Agen:

1. Pastikan Anda memiliki AWS Support Plan yang aktif.
2. Verifikasi izin IAM AWS DevOps Agen Anda termasuk pembuatan kasus dukungan (dukungan:CreateCase, dukungan:DescribeCases).
3. Saat AWS DevOps Agen sedang menyelidiki masalah dan Anda memerlukan bantuan AWS Dukungan, pilih Minta dukungan manusia dari aplikasi web DevOps Agent Space Anda.
4. Tinjau ringkasan investigasi yang akan dibagikan dengan AWS Support.
5. Pilih tingkat keparahan kasus yang sesuai berdasarkan hak paket dukungan Anda.

6. Kirim kasus - AWS DevOps Agen secara otomatis menyertakan log investigasi Anda.

Jendela obrolan terbuka secara otomatis, memungkinkan Anda untuk segera mulai berkolaborasi dengan AWS Support.

Pencegahan insiden proaktif

AWS DevOps Agen menganalisis pola di seluruh investigasi insiden Anda untuk memberikan rekomendasi yang ditargetkan yang terus meningkatkan postur operasional Anda dan mencegah insiden di masa depan. Akses pencegahan insiden proaktif melalui halaman Perbaikan di Aplikasi Web Operator.

Cara kerja pencegahan insiden proaktif

AWS DevOps Agen mengevaluasi investigasi insiden baru-baru ini untuk mengidentifikasi perbaikan jangka panjang untuk mencegah insiden future dan mempercepat mean time to detection (MTTD). Agen menganalisis beberapa insiden untuk mengidentifikasi rekomendasi yang dapat mencegah seluruh kelas insiden di masa depan, dengan fokus pada rekomendasi yang paling berdampak untuk memastikan mereka dapat ditindaklanjuti.

Secara default, agen secara otomatis menjalankan evaluasi setiap minggu. Anda dapat menjeda jadwal jika Anda lebih suka menjalankan evaluasi hanya sesuai permintaan. Evaluasi manual selalu tersedia, yang berguna ketika penyelidikan baru-baru ini menjamin perputaran cepat pada perbaikan yang direkomendasikan.

Agen mengidentifikasi peningkatan di empat kategori, yang ditunjukkan dalam bagan Kategorisasi Rekomendasi di halaman Perbaikan:

- **Observabilitas** — Rekomendasi untuk meningkatkan pemantauan, peringatan, pencatatan, dan visibilitas sistem untuk mendeteksi masalah lebih cepat dan lebih akurat.
- **Infrastruktur** — Rekomendasi untuk mengoptimalkan konfigurasi sumber daya, penyetelan kapasitas, dan ketahanan arsitektur.
- **Tata Kelola** — Rekomendasi untuk memperkuat proses penyebaran, perbaikan saluran pipa, praktik pengujian, dan kontrol operasional.
- **Pengoptimalan kode** — Rekomendasi untuk meningkatkan kualitas kode aplikasi, penanganan kesalahan, dan ketahanan kode.

Kategorisasi ini membantu Anda memahami di mana peningkatan operasional Anda paling dibutuhkan dan memungkinkan Anda memprioritaskan rekomendasi berdasarkan area fokus tim Anda.

Manfaat

- Mencegah insiden berulang — Mengatasi akar penyebab secara sistematis daripada berulang kali menanggapi jenis masalah yang sama
- Kurangi kerja keras operasional — Bebaskan tim Anda dari pemadam kebakaran berulang untuk fokus pada inovasi dan peningkatan strategis
- Meningkatkan ketahanan sistem — Memperkuat infrastruktur, observabilitas, dan proses penyebaran Anda berdasarkan data insiden nyata
- Belajar dari pola historis — Manfaatkan wawasan dari insiden masa lalu untuk membuat perbaikan yang ditargetkan yang memiliki dampak terbesar

Ringkasan agen

Ringkasan Agen di halaman Perbaikan Aplikasi Web memberikan deskripsi hasil dari evaluasi terakhir insiden baru-baru ini. Ringkasan tersebut menjelaskan jumlah investigasi insiden yang dianalisis, insiden mana yang mirip dengan yang sebelumnya, dan rekomendasi mana yang dibuat atau diperbarui dengan informasi baru.

Ringkasan ini membantu Anda dengan cepat memahami apa yang ditemukan agen selama evaluasi terbaru dan menyoroti rekomendasi paling penting yang dapat memiliki dampak terbesar pada postur operasional Anda.

Mengontrol evaluasi

Anda dapat mengontrol kapan AWS DevOps Agen mengevaluasi insiden dan menghasilkan rekomendasi:

- Menjalankan evaluasi secara manual — Klik tombol Jalankan Sekarang di halaman Perbaikan untuk segera memulai evaluasi. Ini berguna ketika penyelidikan baru-baru ini menjamin perputaran cepat pada perbaikan yang direkomendasikan.
- Menghentikan evaluasi aktif — Klik tombol Hentikan Evaluasi di halaman Perbaikan untuk menghentikan evaluasi yang sedang berlangsung.

Mengelola rekomendasi

AWS DevOps Agen memberikan rekomendasi di halaman Perbaikan tempat Anda dapat meninjau dan mengelolanya:

- **Melihat rincian rekomendasi** — Klik pada rekomendasi untuk membuka halaman rincian rekomendasi, di mana Anda dapat melihat informasi lebih lanjut tentang perbaikan yang disarankan termasuk insiden yang menginformasikan rekomendasi, dampak yang diharapkan, dan langkah selanjutnya. Untuk rekomendasi dengan perubahan kode, Anda juga dapat melihat spesifikasi siap agen yang dapat diserahkan ke agen pengkodean untuk implementasi.
- **Keep** — Klik 'Keep' untuk menyimpan rekomendasi di backlog Anda untuk melacak. Ini memungkinkan Anda untuk memantau perbaikan mana yang Anda rencanakan untuk diterapkan dan melacak kemajuannya.
- **Buang** — Klik 'Buang' untuk menghapus rekomendasi dari backlog Anda. Ketika Anda membuang rekomendasi, Anda dapat memberikan penjelasan bahasa alami mengapa itu tidak memenuhi kebutuhan Anda. Agen belajar dari umpan balik ini dan menggunakannya untuk menginformasikan rekomendasi future, memastikan mereka menjadi lebih selaras dengan prioritas operasional dan persyaratan Anda dari waktu ke waktu.
- **Diimplementasikan** — Klik 'Diimplementasikan' untuk menandai rekomendasi sebagai selesai. Ini membantu Anda melacak perbaikan mana yang telah diterapkan dan memungkinkan agen untuk mengukur efektivitas rekomendasinya dari waktu ke waktu.
- **Penghapusan otomatis** — Rekomendasi yang belum ditandai sebagai Tetap atau Diimplementasikan dapat dihapus setelah sekitar 6 minggu jika tidak ada insiden baru yang dicegah dengan menerapkan rekomendasi. Ini memastikan halaman Perbaikan berfokus pada peningkatan yang paling relevan untuk tantangan operasional Anda.
- **Pembaruan rekomendasi** — Rekomendasi yang ada diperbarui ketika insiden baru ditemukan yang akan dicegah oleh rekomendasi. Pembaruan dapat mengubah prioritas rekomendasi atau menyempurnakan rekomendasi berdasarkan wawasan baru.

Prioritas rekomendasi

AWS DevOps Agen secara otomatis memberi peringkat rekomendasi Anda berdasarkan prioritas untuk membantu Anda fokus pada peningkatan yang paling berdampak terlebih dahulu. Peringkat mempertimbangkan konteks spesifik tim Anda, pola operasional, dan tingkat keparahan masalah yang ditangani setiap rekomendasi.

Bagaimana prioritas bekerja

Setiap siklus evaluasi, agen memberi peringkat rekomendasi aktif Anda (yang dalam keadaan diusulkan atau disimpan) menggunakan kombinasi:

- **AI-powered Peringkat** — Agen menilai kepentingan relatif dari rekomendasi teratas Anda berdasarkan relevansi kategori, tingkat keparahan insiden, dan dampak operasional.
- **Penilaian deterministik** — Untuk backlog yang lebih besar, agen menerapkan skor prioritas berdasarkan frekuensi insiden, pola keparahan, dan kebaruan untuk memastikan pemesanan yang konsisten di luar item peringkat teratas.

Daftar peringkat muncul di halaman Perbaikan dengan posisi peringkat numerik (1 menjadi prioritas tertinggi). Rekomendasi yang telah dibuang atau diterapkan tidak diberi peringkat.

Menyesuaikan prioritas

Anda dapat memengaruhi cara agen memberi peringkat rekomendasi dengan mengomunikasikan prioritas tim Anda melalui antarmuka obrolan:

- **Menyetel preferensi kategori** — Beri tahu agen kategori rekomendasi mana yang paling penting bagi tim Anda (misalnya, “Kami memprioritaskan peningkatan observabilitas daripada perubahan infrastruktur”). Agen menyimpan preferensi ini dan menggunakannya dalam evaluasi peringkat masa depan.
- **Menyediakan konteks** — Bagikan informasi tentang proyek yang akan datang, persyaratan kepatuhan, atau area fokus tim. Agen menggabungkan konteks ini ketika menentukan rekomendasi mana yang harus diprioritaskan.

Untuk memperbarui preferensi Anda, gunakan antarmuka obrolan dan jelaskan prioritas tim Anda dalam bahasa alami. Agen akan mengonfirmasi bahwa ia telah memahami dan akan menerapkan preferensi Anda dalam siklus evaluasi berikutnya.

Stabilitas peringkat

Peringkat rekomendasi dapat berubah di antara siklus evaluasi ketika:

- Rekomendasi baru ditambahkan yang memiliki prioritas lebih tinggi daripada yang sudah ada
- Preferensi yang dinyatakan tim Anda berubah

- Data insiden baru memperkuat atau melemahkan kasus untuk rekomendasi

Rekomendasi yang telah Anda tandai sebagai Keep mempertahankan posisinya di backlog Anda terlepas dari perubahan peringkat, memastikan alur kerja Anda tidak terganggu.

Agent-ready spesifikasi

Untuk rekomendasi yang melibatkan perubahan kode atau konfigurasi, AWS DevOps Agen dapat menghasilkan spesifikasi siap agen. Spesifikasi ini menyediakan dokumen terstruktur yang dapat diserahkan langsung ke agen pengkodean untuk implementasi.

Spesifikasi meliputi:

- Pernyataan masalah — Ringkasan masalah dan akar penyebabnya
- Ringkasan solusi — Deskripsi tingkat tinggi dari pendekatan yang direkomendasikan
- Repositori target - Repositori spesifik tempat perubahan perlu dilakukan
- Perubahan kode — Deskripsi terperinci tentang apa yang perlu diubah dan mengapa, dengan jalur file tertentu dan pertimbangan implementasi
- Persyaratan pengujian - Skenario apa yang perlu diuji
- Rencana implementasi — Pendekatan bertahap untuk mengimplementasikan perubahan

Agent-ready spesifikasi mempercepat implementasi dengan menyediakan agen pengkodean dengan konteks yang mereka butuhkan untuk membuat perubahan siap produksi tanpa memerlukan bolak-balik ekstensif dengan para insinyur.

Menerapkan rekomendasi

Untuk memaksimalkan nilai rekomendasi pencegahan insiden proaktif, pertimbangkan praktik berikut untuk menindaklanjutinya:

- Menggunakan spesifikasi siap agen — Untuk rekomendasi dengan perubahan kode, gunakan spesifikasi yang dihasilkan untuk mempercepat implementasi dengan menyerahkannya ke agen pengkodean atau menggunakannya sebagai panduan terperinci untuk implementasi manual.
- Menambahkan rekomendasi ke backlog tiket Anda — Salin rekomendasi ke sistem tiket tim Anda atau alat manajemen proyek untuk memastikan mereka diprioritaskan bersama pekerjaan teknik lainnya.

- Memprioritaskan rekomendasi berdasarkan dampak — Fokus pertama pada rekomendasi yang membahas jenis insiden yang paling sering atau parah, atau yang mempengaruhi sistem kritis.
- Melacak kemajuan implementasi — Memantau rekomendasi mana yang telah diterapkan dan mengukur efektivitasnya dengan mengamati apakah insiden serupa menurun dari waktu ke waktu.
- Berkoordinasi dengan tim pengembangan — Bagikan rekomendasi dengan tim yang sesuai yang memiliki sistem yang terpengaruh, memastikan mereka memiliki konteks dan sumber daya yang diperlukan untuk menerapkan perbaikan.

DevOps Tugas Sesuai Permintaan

AWS DevOps Agent On Demand Tasks adalah asisten percakapan yang didukung kecerdasan buatan (AI) generatif yang memungkinkan tim operasi untuk menanyakan arsitektur aplikasi mereka, menganalisis kesehatan sistem, dan mengakses wawasan investigasi menggunakan bahasa alami. Anda dapat mengajukan pertanyaan tentang AWS sumber daya, metrik sistem, status alarm, riwayat penerapan, dan pola insiden. Obrolan memberikan jawaban langsung yang didasarkan pada data infrastruktur dan operasi Anda yang sebenarnya, sehingga tidak perlu menavigasi di antara beberapa AWS konsol atau alat pemantauan.

Obrolan terintegrasi di seluruh aplikasi web DevOps Agent Space dan memberikan respons sadar konteks berdasarkan halaman yang Anda lihat. Antarmuka mempertahankan riwayat percakapan, memungkinkan Anda untuk melanjutkan diskusi sebelumnya dan membangun kueri sebelumnya.

Kemampuan tugas

AWS DevOps Agent On Demand Tasks menyediakan kemampuan komprehensif untuk membantu Anda mengelola dan memahami infrastruktur Anda:

Kueri sumber daya — Tanyakan tentang AWS sumber daya di Ruang Agen Anda, termasuk fungsi Lambda, tabel DynamoDB, penerapan EKS, sertifikat, dan konfigurasi infrastruktur. Obrolan dapat memfilter dan menganalisis sumber daya berdasarkan atribut seperti versi runtime, pengaturan kapasitas, atau status penerapan. Misalnya, tanyakan “Berapa banyak Lambda yang menggunakan Python 3.8?” atau “Apakah saya memiliki sertifikat yang akan kedaluwarsa?”

Analisis kesehatan sistem — Kueri metrik kesehatan sistem saat ini dan historis, termasuk status alarm, tingkat kesalahan, pemanfaatan CPU, dan ketersediaan layanan. Obrolan dapat menghasilkan ringkasan kesehatan yang mencakup periode waktu tertentu dan mengidentifikasi tren perilaku sistem. Ajukan pertanyaan seperti “Alarm mana yang menyala dalam 24 jam terakhir?” atau “Ada kesalahan 5xx dalam satu jam terakhir?”

Wawasan investigasi — Akses informasi dari investigasi yang telah selesai dan sedang berlangsung, termasuk analisis akar penyebab, hipotesis yang dieksplorasi, log yang ditinjau, dan pola resolusi. Obrolan dapat mengidentifikasi penyebab insiden umum dan memberikan rekomendasi berdasarkan data historis. Pertanyaan “Apa penyebab paling umum dari insiden bulan lalu?” atau “Berapa waktu resolusi rata-rata untuk menyelesaikan investigasi?”

Investigasi steering — Saat melihat halaman detail investigasi, pandu investigasi dengan mengarahkan agen untuk fokus pada log tertentu, mengeksplorasi hipotesis tertentu, atau memperbarui analisis akar penyebab. Berikan input kemudi seperti “Fokus pada log untuk layanan pembayaran dan perbarui RCA Anda” atau “Jelajahi hipotesis bahwa pelambatan DynamoDB menyebabkan masalah.”

Artefak obrolan — Menghasilkan laporan dan dokumen terstruktur, seperti ringkasan kesehatan operasional, laporan kesalahan, dan analisis insiden. Artefak muncul di panel khusus dan mendukung pengeditan berversi dalam percakapan.

Lampiran file — Lampirkan gambar, dokumen, dan file kode ke pesan Anda sehingga Chat dapat menganalisisnya dalam konteks. Misalnya, lampirkan tangkapan layar dasbor alarm, file konfigurasi YAMB, atau PDF runbook, dan tanyakan Obrolan apa yang harus dilakukan selanjutnya. Lihat [Mengirim lampiran file](#) untuk detailnya.

Penyaringan rekomendasi — Rekomendasi pencegahan insiden kueri dengan kriteria tertentu, seperti rekomendasi yang terkait dengan layanan tertentu atau masalah operasional. Obrolan menjelaskan dampak dan pertimbangan implementasi untuk setiap rekomendasi. Misalnya, “Tunjukkan rekomendasi yang akan mencegah insiden yang melibatkan DynamoDB” atau “Rekomendasi mana yang akan membantu saya mendeteksi masalah latensi permintaan lebih cepat?”

Mengakses Obrolan

Obrolan tersedia sebagai panel persisten di sisi kiri aplikasi web DevOps Agent Space. Bilah sisi kiri mencakup tombol obrolan + Baru, bagian Halaman untuk menavigasi ke Insiden, Peningkatan, dan Topologi, dan bagian Obrolan yang menampilkan percakapan terbaru Anda. Pilih Lihat semua untuk melihat riwayat percakapan lengkap Anda.

Obrolan memberikan respons sadar konteks berdasarkan tempat Anda mengaksesnya:

Topologi — Ajukan pertanyaan umum tentang sumber daya, arsitektur, dan kesehatan operasional Ruang Agen Anda. Obrolan memiliki visibilitas penuh ke semua akun dan layanan yang terhubung.

Dari konteks ini, Anda dapat menanyakan konfigurasi sumber daya, riwayat penerapan, informasi topologi, dan integrasi alat observabilitas.

Respons Insiden — Saat melihat halaman respons insiden, ajukan pertanyaan tentang tren investigasi, waktu penyelesaian, dan pola insiden di seluruh Ruang Agen Anda. Obrolan dapat menganalisis data investigasi historis untuk mengidentifikasi penyebab umum dan peluang peningkatan.

Detail Investigasi — Saat melihat investigasi tertentu, Chat memberikan tanggapan sadar konteks tentang penyelidikan itu. Tanyakan tentang log yang ditinjau, hipotesis dieksplorasi, kesimpulan akar penyebab, dan rencana mitigasi. Anda juga dapat memberikan masukan kemudi untuk memandu fokus investigasi.

Pencegahan — Dari halaman pencegahan, kueri rekomendasi dengan filter, pahami mengapa rekomendasi dibuat, dan jelajahi pendekatan implementasi. Obrolan membantu Anda memprioritaskan dan memahami dampak rekomendasi pencegahan insiden.

Antarmuka obrolan tetap tersedia saat Anda beralih antar halaman, tetapi konteksnya berubah untuk memberikan informasi yang relevan untuk tampilan Anda saat ini. Ketika Anda memulai percakapan baru, itu dimulai tanpa konteks sebelumnya. Saat Anda melanjutkan percakapan yang ada, Chat menyimpan riwayat percakapan lengkap untuk pertanyaan tindak lanjut.

Context-aware tanggapan

Obrolan menyesuaikan tanggapannya berdasarkan halaman yang Anda lihat di aplikasi web DevOps Agent Space. Kesadaran konteks ini memastikan Anda menerima informasi yang relevan tanpa perlu menentukan investigasi atau ruang lingkup sumber daya yang Anda tanyakan.

Saat melihat halaman detail investigasi, Chat secara otomatis memahami bahwa Anda bertanya tentang penyelidikan spesifik tersebut. Pertanyaan seperti “Log apa yang Anda lihat?” atau “Hipotesis mana yang Anda jelajahi?” mengacu pada investigasi yang saat ini ditampilkan. Saat Anda memberikan input kemudi, Chat menerapkannya ke investigasi aktif dan membuat versi akar penyebab baru jika sesuai.

Pada halaman pencegahan, Chat memahami bahwa Anda tertarik dengan rekomendasi pencegahan insiden. Kueri secara otomatis memfilter dan menganalisis rekomendasi dalam konteks Ruang Agen Anda. Sistem mengenali apakah Anda bertanya tentang rekomendasi umum atau rincian rekomendasi spesifik.

Saat mengakses Obrolan dari halaman Topologi, Obrolan menyediakan visibilitas luas di semua sumber daya, metrik, dan data historis di Ruang Agen Anda. Anda dapat bertanya tentang sumber daya, layanan, atau masalah operasional apa pun tanpa menentukan konteks investigasi atau rekomendasi.

Kesadaran konteks ini menghilangkan kebutuhan untuk berulang kali menentukan investigasi, rekomendasi, atau ruang lingkup sumber daya yang Anda referensikan, menciptakan aliran percakapan yang lebih alami.

Mengelola percakapan

Chat menyimpan riwayat percakapan untuk memungkinkan Anda melanjutkan diskusi sebelumnya dan referensi kueri sebelumnya.

Membuat percakapan baru — Klik tombol “Sesi baru” di panel obrolan untuk memulai percakapan baru tanpa konteks sebelumnya. Percakapan baru tidak membawa informasi dari obrolan sebelumnya, memungkinkan Anda untuk mengajukan pertanyaan yang tidak terkait tanpa kebingungan.

Mengakses riwayat percakapan — Klik “Riwayat” untuk melihat semua percakapan sebelumnya dalam Ruang Agen Anda. Percakapan diatur secara kronologis dengan stempel waktu dan teks pratinjau. Riwayat percakapan disimpan selama 90 hari dan bersifat pribadi untuk akun pengguna Anda dalam Ruang Agen.

Percakapan lanjutan — Pilih percakapan apa pun dari riwayat Anda untuk melanjutkan di mana Anda tinggalkan. Obrolan mempertahankan konteks lengkap pesan sebelumnya, memungkinkan Anda untuk mengajukan pertanyaan lanjutan yang merujuk pada bagian percakapan sebelumnya. Saat Anda mengganti halaman saat melihat percakapan, konteks percakapan tetap ada tetapi konteks khusus halaman diperbarui berdasarkan lokasi Anda saat ini.

Perhatikan bahwa riwayat percakapan diisolasi dalam setiap Ruang Agen. Percakapan di satu Ruang Agen tidak terlihat atau dapat diakses dari Ruang Agen lainnya. Isolasi ini memastikan bahwa informasi sensitif tetap terkotak-kotak sesuai dengan batas-batas organisasi Anda.

Menghasilkan artefak

AWS DevOps Agen mendukung artefak obrolan — dokumen terstruktur dan berversi yang dihasilkan oleh agen selama percakapan. Artefak menyediakan panel interaktif khusus di UI obrolan untuk meninjau dan mengedit AI-generated konten, seperti laporan operasional, ringkasan kesalahan, dan penilaian kesehatan.

Anda dapat meminta artefak dari halaman mana pun di aplikasi web DevOps Agent Space. Obrolan menggunakan konteks halaman saat ini untuk mencakup konten artefak.

Bagaimana artefak bekerja

Saat Anda meminta Chat untuk membuat atau memperbarui konten, Chat menghasilkan artefak — biasanya dokumen yang diformat — dan menampilkannya di panel artefak di samping percakapan.

Hasilkan - Kirim permintaan bahasa alami untuk membuat laporan atau dokumen. Misalnya, tanyakan “Buat laporan kesehatan operasional mingguan untuk Ruang Agen saya” atau “Tunjukkan laporan untuk kesalahan 4xx saya dari minggu lalu”.

Ulasan — Artefak muncul di panel khusus di samping percakapan. Anda dapat meninjau konten lengkap sambil terus berinteraksi dengan Obrolan.

Edit — Minta perubahan pada artefak melalui Obrolan. Misalnya, tanyakan “Tambahkan bagian tentang Lambda cold start” atau “Perbarui laporan untuk menyertakan data bulan lalu”. Obrolan membuat artefak versi baru dengan perubahan yang Anda minta.

Mengirim lampiran file

Anda dapat melampirkan file ke pesan obrolan Anda sehingga Obrolan dapat membacanya bersama dengan pertanyaan Anda. Gunakan lampiran untuk membagikan apa yang Anda lihat — tangkapan layar dasbor atau alarm, file konfigurasi, kode sumber, runbook operasional — dan minta agen untuk beralasan secara langsung.

File dicakup ke Ruang Agen Anda — file tersebut tidak terlihat dari Ruang Agen lain, dan akses dibatasi oleh izin IAM yang sama yang menandai sisa Obrolan. File diunggah ke penyimpanan Ruang Agen terkelola segera setelah Anda melampirkannya.

Cara melampirkan file

Anda dapat menambahkan file ke pesan dengan tiga cara:

- Pilih ikon unggah di toolbar input obrolan dan pilih satu atau beberapa file dari perangkat Anda.
- Seret dan lepas satu atau beberapa file ke area input obrolan.
- Tempel gambar langsung dari clipboard Anda, misalnya setelah mengambil tangkapan layar.

Setiap file yang Anda lampirkan muncul sebagai chip di input obrolan dengan indikator kemajuan unggahan. Untuk melihat pratinjau file, pilih chipnya. Untuk menghapus file, pilih X pada chip. Tombol Kirim tetap dinonaktifkan saat file terlampir masih diunggah.

Tipe file yang didukung

Chat menerima tiga kategori file berikut:

- Gambar —png,jpeg,jpg,gif, webp
- Dokumen — pdf, csv, doc, docx, xls, xlsx, html, txt, md
- File teks dan kode — json, yaml, yml, xml, js, ts, py, java, rb, go, rs, sh, bash, log, cfg, ini, toml

File di luar kategori ini ditolak sebelum diunggah.

Batas

Batasan berikut berlaku untuk setiap pesan:

Kuota	Nilai
Ukuran maksimum file	3,75 MB
Lampiran per pesan (campuran jenis apa pun)	20
Dari jumlah tersebut, dokumen biner (PDF, DOC, DOCX, XLS, XLSX)	hingga 5

Selain itu, teks pesan dan konten lampiran Anda harus sesuai dengan jendela konteks per-pesan model. Jika pesan dan lampirannya terlalu besar, Chat menolak pesan dan meminta Anda untuk mengurangi ukuran atau jumlah lampiran sebelum mengirim.

Kasus penggunaan

Cara umum untuk menggunakan lampiran file dengan DevOps Agen:

- Lampirkan tangkapan layar dasbor alarm atau kesalahan dan minta Obrolan untuk menafsirkan apa yang gagal dan ke mana harus mencari selanjutnya.

- Lampirkan kode sumber layanan dan minta Chat untuk meninjau perubahan, menyarankan perbaikan, atau menjelaskan perilakunya.
- Lampirkan file konfigurasi (misalnya, konfigurasi YAMG, JSON, atau TOLL) dan minta Chat untuk memecahkan masalah mengapa penerapan, alarm, atau integrasi berperilaku buruk.
- Lampirkan buku runbook operasional atau laporan pasca-insiden PDF dan minta Chat untuk mengubahnya menjadi keterampilan — agen mengekstrak prosedur dan menyimpannya ke Ruang Agen Anda sehingga penyelidikan di masa depan dapat menerapkannya secara otomatis.

Kueri Sampel

Contoh berikut menunjukkan jenis pertanyaan yang dapat Anda ajukan Obrolan. Contoh-contoh ini diatur berdasarkan kasus penggunaan dan konteks.

Kueri pembuatan artifak

Dari halaman mana pun di aplikasi web DevOps Agent Space:

- Buat ringkasan kesehatan operasional mingguan untuk Ruang Agen saya
- Buat laporan semua kesalahan 4xx dari minggu lalu
- Buat laporan ringkasan insiden selama 30 hari terakhir
- Buat ringkasan aktivitas alarm untuk layanan pembayaran minggu ini
- Buat laporan riwayat penyebaran selama 7 hari terakhir
- Ringkas semua rekomendasi terbuka ke dalam laporan

Pertanyaan informasi sumber daya

Dari halaman mana pun di aplikasi web DevOps Agent Space:

- Berapa banyak fungsi Lambda yang menggunakan Python 3.8?
- Apakah saya memiliki sertifikat yang akan kedaluwarsa?
- Daftar semua tabel DynamoDB dengan penagihan sesuai permintaan
- Tunjukkan kluster EKS dalam produksi
- Fungsi Lambda mana yang belum digunakan dalam 90 hari terakhir?
- Daftar bucket S3 tanpa mengaktifkan versi

- Instans RDS apa yang menjalankan database versi X?

Pertanyaan kesehatan sistem

Dari halaman Topologi atau Respons Insiden:

- Alarm mana yang ditembakkan dalam 24 jam terakhir?
- Ada kesalahan 5xx dalam satu jam terakhir?
- Tunjukkan tren kesalahan Lambda untuk layanan pembayaran
- Apa pemanfaatan CPU untuk cluster ECS saya?
- Apakah ada target yang tidak sehat di penyeimbang beban saya?
- Tunjukkan kepada saya peristiwa pelambatan API Gateway dari kemarin
- Layanan mana yang memiliki tingkat kesalahan tertinggi minggu lalu?
- Beri saya laporan kesehatan secara keseluruhan yang mencakup 24 jam terakhir

Kueri alat observabilitas

Dari Topologi:

- Daftar grup log Splunk
- Tunjukkan metrik Prometheus dan ambang alarm mereka
- Monitor Datadog apa yang dikonfigurasi untuk layanan ini?
- Daftar kebijakan peringatan Relik Baru
- Tunjukkan konfigurasi dasbor Dynatrace

Pertanyaan wawasan investigasi

Dari halaman Respons Insiden:

- Apa penyebab paling umum dari insiden bulan lalu?
- Berapa waktu resolusi rata-rata untuk menyelesaikan investigasi?
- Ringkas investigasi dari minggu lalu dan RCA mereka
- Berapa banyak insiden yang disebabkan oleh DynamoDB throttling?

- Tunjukkan tren investigasi selama kuartal terakhir
- Layanan mana yang memiliki insiden paling sering?

Pertanyaan detail investigasi

Dari halaman Detail Investigasi:

- Log apa yang Anda lihat?
- Hipotesis apa yang Anda jelajahi?
- Seberapa berisiko tindakan mitigasi yang Anda usulkan?
- Apa garis waktu peristiwa selama insiden ini?
- Mengapa Anda menyimpulkan ini adalah akar penyebabnya?
- Bukti apa yang mendukung analisis akar penyebab Anda?
- Siapa yang memberikan kemudi selama penyelidikan Anda?
- Beri aku ringkasan investigasi insiden ini

Pertanyaan kemudi investigasi

Dari halaman Detail Investigasi:

- Fokus pada log untuk layanan pembayaran antara 14:00-15:00 UTC dan perbarui RCA Anda
- Jelajahi hipotesis bahwa pelambatan DynamoDB menyebabkan masalah
- Periksa konfigurasi cluster ECS untuk melihat apakah itu menyebabkan alarm
- Hanya periksa log selama 2 jam terakhir, bukan sehari penuh
- Selidiki lonjakan kesalahan pada pukul 3 sore
- Lihat log API Gateway alih-alih log Lambda

Pertanyaan rekomendasi pencegahan

Dari halaman Pencegahan:

- Apa rekomendasi pencegahan insiden 3 teratas saya?
- Tunjukkan rekomendasi yang akan mencegah insiden yang melibatkan DynamoDB

- Rekomendasi mana yang akan membantu saya mendeteksi masalah latensi permintaan lebih cepat?
- Buat daftar perbaikan observabilitas yang dapat mencegah insiden serupa
- Tunjukkan rekomendasi infrastruktur untuk layanan pembayaran
- Rekomendasi mana yang memiliki dampak tertinggi pada ketahanan sistem?

Mengaktifkan Obrolan di Ruang Agen Anda

Obrolan tersedia di semua aplikasi web DevOps Agent Space. Proses penyiapan tergantung pada apakah Anda memiliki Ruang Agen baru atau yang sudah ada.

Ruang Agen Baru

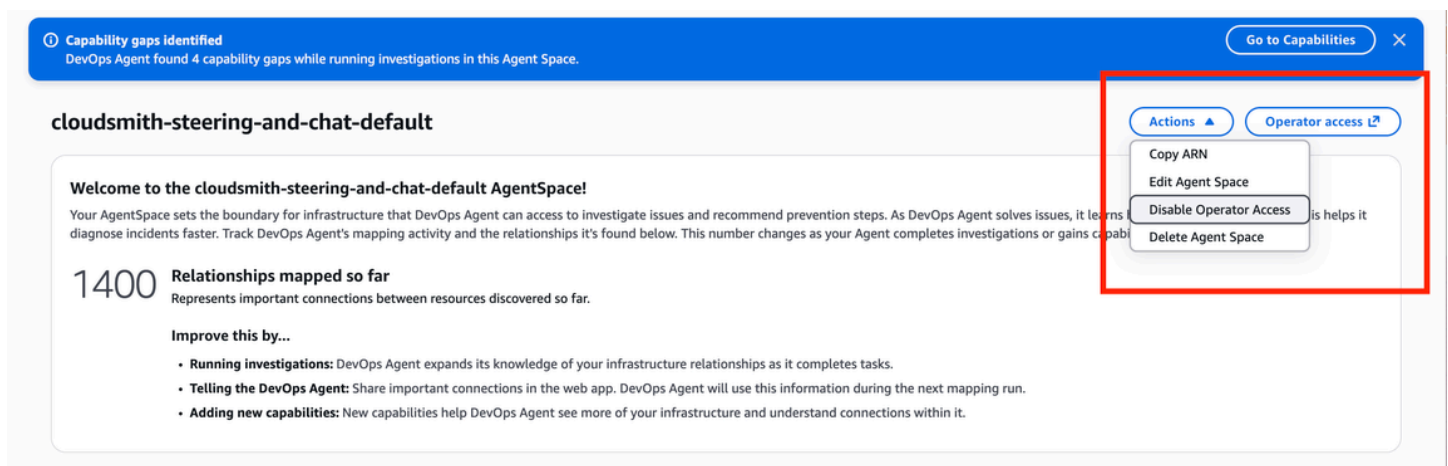
Obrolan diaktifkan secara otomatis saat Anda membuat Ruang Agen baru. Tidak diperlukan konfigurasi tambahan atau pengaturan izin IAM. Setelah Anda mengonfigurasi aplikasi web DevOps Agent Space, Obrolan segera tersedia sebagai panel persisten di sisi kiri halaman mana pun.

Ruang Agen yang Ada

Jika Anda membuat Ruang Agen sebelum Obrolan dirilis, Anda harus mengaktifkan izin IAM yang diperlukan. Anda memiliki dua pilihan:

Opsi 1: Mencabut dan mengaktifkan kembali akses aplikasi operator

Arahkan ke Konsol Admin AWS DevOps Agen, cari dropdown Tindakan di sudut kanan atas, dan nonaktifkan konfigurasi akses operator saat ini.



The screenshot shows the AWS DevOps Agent console interface. At the top, a blue notification bar states "Capability gaps identified" and "DevOps Agent found 4 capability gaps while running investigations in this Agent Space." Below this, the page title is "cloudsmith-steering-and-chat-default". The main content area displays a welcome message and a "Relationships mapped so far" section with a value of 1400. On the right side, there is a "Actions" dropdown menu with the following options: "Copy ARN", "Edit Agent Space", "Disable Operator Access", and "Delete Agent Space". The "Disable Operator Access" option is highlighted with a red box, indicating the current step in the process.

Kemudian aktifkan opsi buat otomatis untuk akses operator.

Connect observability-newrelic-default to IAM Identity Center

IAM Identity Center Instance
Your Web App user access will be managed by the following IAM Identity Center instance
[ssoins-722823a2de611c55](#)

IAM Identity Center Application Role Name
Authenticated Web App users will use the following IAM role to access DevOps Agent

Auto-create a new DevOps Agent role
Create and use a new service role

Assign an existing role
Provided role will be verified by DevOps Agent

Create a new DevOps Agent role using a policy template
Use provided details to create your own role in the IAM Console

Web app role name that will be created
DevOpsAgentRole-WebappIDC-fpwoc9xn

Operator access

IAM Role name for administrator access
This role provides administrator access for setup and configuration of your web app

Auto-create a new DevOps Agent role
Create and use a new service role

Assign an existing role
Provided role will be verified by DevOps Agent

Create a new DevOps Agent role using a policy template
Use provided details to create your own role in the IAM Console

Web app role name that will be created
DevOpsAgentRole-WebappAdmin-zq3mg548

Connect

Configure web app

Ini secara otomatis menerapkan izin IAM yang diperlukan untuk Obrolan bersama dengan semua izin operator lainnya saat ini.

Opsi 2: Tambahkan izin IAM secara manual

Tambahkan izin IAM berikut ke peran akses operator yang ada:

- `aidevops:ListChats`— Lihat riwayat percakapan obrolan
- `aidevops:CreateChat`— Buat percakapan obrolan baru
- `aidevops:SendMessage`— Kirim pesan dan terima tanggapan

Arahkan ke konsol AWS IAM, cari peran operator DevOps Agen, dan tambahkan izin ini ke kebijakan peran. Obrolan menjadi tersedia segera setelah izin ditambahkan.

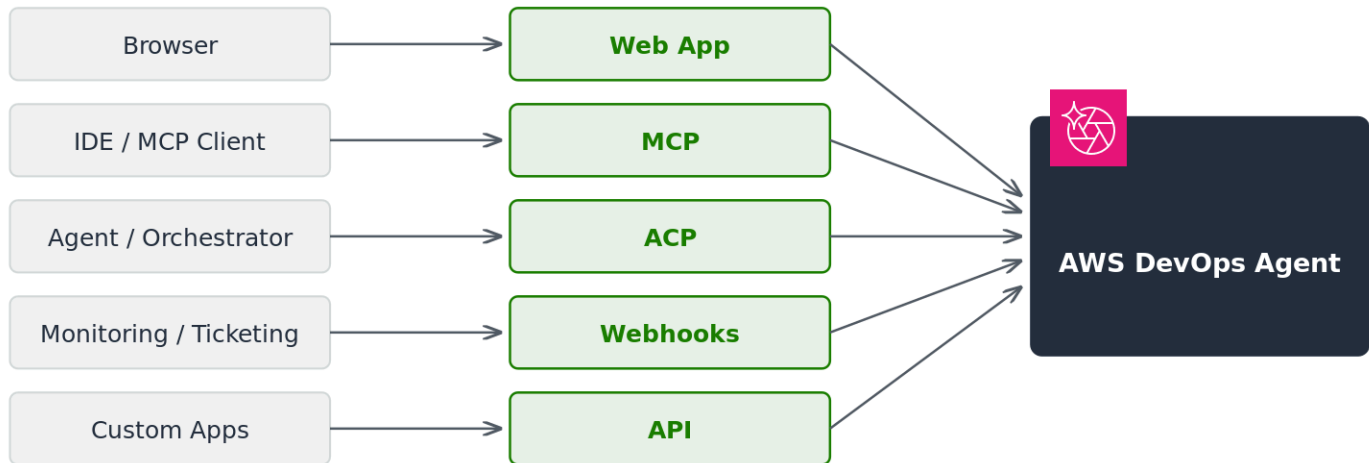
Setelah menyelesaikan salah satu opsi, segarkan aplikasi web DevOps Agent Space Anda dan panel obrolan muncul di sisi kiri halaman mana pun.

Interface dengan Agen DevOps

AWS DevOps Agen mendukung lima metode akses: konsol aplikasi web, integrasi Model Context Protocol (MCP), integrasi Agent Client Protocol (ACP), webhook untuk otomatisasi berbasis peristiwa,

dan akses API langsung. Pilih metode yang paling sesuai dengan alur kerja dan persyaratan teknis Anda.

Diagram berikut menggambarkan metode akses ini dan bagaimana mereka terhubung ke layanan DevOps Agen.



DevOps Aplikasi web agen

Aplikasi web adalah antarmuka utama untuk DevOps Agen. Gunakan obrolan percakapan untuk menyelidiki insiden, menanyakan infrastruktur Anda, dan mengelola rekomendasi. Untuk informasi selengkapnya, lihat [the section called “Apa itu Aplikasi Web DevOps Agen?”](#).

Integrasi Model Context Protocol (MCP)

Anda dapat mengakses kemampuan AWS DevOps Agen langsung dari MCP-compatible klien dan IDE. Gunakan [AWS MCP Server](#) untuk terhubung. Anda dapat menyelidiki insiden, mengoptimalkan biaya, meninjau arsitektur, dan memetakan topologi tanpa meninggalkan lingkungan pengembangan Anda.

[Untuk pengguna Kiro, daya aws-devops-agent khusus tersedia di repositori daya Kiro.](#) Kekuatan ini menghubungkan Kiro ke AWS DevOps Agen melalui AWS MCP Server. Ini memberikan intelijen AI-powered operasional langsung di IDE Anda.

Untuk pengguna [Claude Code](#), [sample-aws-devops-agent-claude-plugin menyediakan plugin pra-konfigurasi yang menghubungkan Claude Code ke Agent](#) melalui MCP Server. AWS DevOps AWS

Integrasi Protokol Klien Agen (ACP)

Anda dapat memanggil AWS DevOps Agen secara terprogram menggunakan Agent [Client Protocol \(ACP\)](#). Untuk implementasi sampel, lihat repositori [sample-aws-devops-agent-acp-mcp](#) di GitHub

Webhook

Webhook memungkinkan sistem eksternal untuk secara otomatis memicu investigasi AWS DevOps Agen. Sistem eksternal seperti platform tiket dan alat pemantauan dapat mengirim permintaan HTTP ketika insiden terjadi. Untuk informasi selengkapnya, lihat [the section called “Memanggil DevOps Agen melalui Webhook”](#).

AWS DevOps Agen API

AWS DevOps Agen menyediakan API untuk akses terprogram ke kemampuan agen. Anda dapat membuat dan mengelola Ruang Agen, memicu investigasi, dan mengambil temuan. Untuk informasi selengkapnya, lihat [Referensi API AWS DevOps Agen](#).

Mengkonfigurasi kemampuan untuk AWS DevOps Agen

AWS DevOps Kemampuan agen memperluas fungsionalitas agen Anda dengan menghubungkannya ke alat dan infrastruktur yang ada. Konfigurasi kemampuan ini untuk memungkinkan penyelidikan insiden yang komprehensif, alur kerja respons otomatis, dan integrasi tanpa batas dengan ekosistem Anda DevOps .

Kemampuan berikut membantu Anda memaksimalkan efektivitas DevOps Agen Anda:

- **AWS EKS Access Setup** - Aktifkan introspeksi cluster Kubernetes, log pod, dan peristiwa cluster untuk lingkungan EKS publik dan pribadi
- **Integrasi Azure** - Hubungkan langganan Azure dan DevOps organisasi Azure untuk menyelidiki sumber daya Azure dan menghubungkan penyebaran Azure dengan insiden DevOps
- **CI/CD Pipeline Integration** - Connect GitHub dan GitLab pipelines untuk menghubungkan penerapan dengan insiden dan melacak perubahan kode selama investigasi
- **Koneksi Server MCP** - Memperluas kemampuan investigasi dengan menghubungkan alat observabilitas eksternal dan sistem pemantauan khusus melalui Protokol Konteks Model
- **Multi-Account AWS Akses** - Konfigurasi AWS akun sekunder untuk menyelidiki sumber daya di seluruh organisasi Anda selama respons insiden
- **Integrasi Sumber Telemetri** - Connect platform pemantauan seperti Datadog, Dynatrace, Grafana, New Relic, dan Splunk untuk akses data observabilitas yang komprehensif
- **Integrasi Tiket dan Obrolan** - Connect ServiceNow, PagerDuty, dan Slack untuk mengotomatiskan alur kerja respons insiden dan memungkinkan kolaborasi tim
- **Konfigurasi Webhook** - Izinkan sistem eksternal untuk secara otomatis memicu investigasi DevOps Agen melalui permintaan HTTP. Untuk detail tentang penyiapan webhook, metode otentikasi, dan format permintaan, lihat [the section called “Memanggil DevOps Agen melalui Webhook”](#)
- **EventBridge Integrasi Amazon** - Menggabungkan AWS DevOps Agen ke dalam aplikasi berbasis peristiwa dengan merutekan peristiwa investigasi dan mitigasi siklus hidup ke target Amazon EventBridge

Anda dapat mengonfigurasi setiap kemampuan secara independen berdasarkan kebutuhan spesifik tim Anda dan tumpukan alat yang ada. Mulailah dengan integrasi yang paling penting untuk alur kerja respons insiden Anda, lalu perluas ke kemampuan tambahan sesuai kebutuhan.

Migrasi dari pratinjau publik ke ketersediaan umum

Jika Anda menggunakan AWS DevOps Agen selama pratinjau publik, Anda harus memperbarui peran IAM Anda sebelum rilis GA. Panduan ini berjalan melalui pembaruan peran pemantauan dan peran operator di akun Anda.

Apa yang berubah

1. [Riwayat obrolan sesuai permintaan selama pratinjau tidak lagi dapat diakses](#)
2. [Kebijakan terkelola baru menggantikan kebijakan yang tersedia selama pratinjau](#)
3. [Agent Spaces mungkin memiliki cakupan akses aplikasi IAM Identity Center yang sudah ketinggalan zaman](#)

Riwayat obrolan sesuai permintaan dari pratinjau publik

Rilis GA memperkenalkan langkah-langkah keamanan tambahan untuk memperkuat kontrol akses untuk riwayat obrolan. Sebagai hasil dari perubahan ini, riwayat obrolan sesuai permintaan dari periode pratinjau publik (sebelum 30 Maret 2026) tidak lagi dapat diakses. Jurnal investigasi dan temuan yang dibuat selama pratinjau publik tidak terpengaruh. Perubahan ini hanya berlaku untuk percakapan obrolan sesuai permintaan.

Kebijakan Terkelola Baru

Untuk GA, AWS berikan kebijakan terkelola baru yang menggantikan kebijakan era pratinjau:

Jenis peran	Menghapus	Tambahkan
Memantau	Kebijakan terkelola AIOpsAssistantPolicy	Kebijakan terkelola AIDevOpsAgentAccessPolicy
Operator (IAM dan IDC)	Kebijakan sebaris	Kebijakan terkelola AIDevOpsOperatorAppAccessPolicy

Selain itu, peran operator memerlukan kebijakan kepercayaan yang diperbarui, dan peran operator IDC memerlukan kebijakan inline baru.

Prasyarat

- Akses ke AWS akun tempat peran DevOps Agen Anda dikonfigurasi (akun primer dan semua akun sekunder)
- Izin IAM untuk mengubah peran, kebijakan, dan hubungan kepercayaan
- ID Ruang Agen, ID AWS akun, dan Wilayah Anda (terlihat di konsol DevOps Agen)

Langkah 1: Perbarui peran pemantauan

Perbarui peran pemantauan di akun utama Anda dan di setiap akun sekunder. Ini adalah peran Primary/Secondary sumber yang dikonfigurasi di bawah tab Kemampuan di ruang agen Anda (contoh primary/secondary peran:DevOpsAgentRole-AgentSpace-3xj2396z).

1. Di konsol DevOps Agen, buka Ruang Agen Anda dan pilih tab Kemampuan.
2. Temukan peran pemantauan untuk Primary/Secondary Sumber Anda (misalnya,DevOpsAgentRole-AgentSpace-3xj2396z) dan pilih Edit.
3. Di bawah kebijakan Izin, hapus kebijakan AI0psAssistantPolicy AWS terkelola.
4. Pilih Tambahkan izin, Lampirkan kebijakan, dan lampirkan kebijakan AIDevOpsAgentAccessPolicy terkelola.
5. Edit kebijakan inline dan ganti isinya dengan yang berikut, ganti ID akun Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateServiceLinkedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<account-id>:role/aws-service-role/resource-explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
```

1. Kebijakan kepercayaan untuk peran pemantauan tidak memerlukan perubahan. Verifikasi itu cocok dengan yang berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:<region>:<account-id>:agentspace/*"
        }
      }
    }
  ]
}
```

- Ulangi langkah 2—6 untuk peran pemantauan di setiap akun sekunder.

Langkah 2: Perbarui peran operator (IAM)

1. Di konsol DevOps Agen, pilih tab Access dan temukan peran operator.
2. Di konsol IAM, hapus kebijakan sebaris yang ada dari peran operator.
3. Pilih Tambahkan izin, Lampirkan kebijakan, dan lampirkan kebijakan `AIDevOpsOperatorAppAccessPolicy` terkelola.
4. Pilih tab Trust relationship dan pilih Edit trust policy. Ganti kebijakan kepercayaan dengan yang berikut, ganti ID akun, Wilayah, dan ID Ruang Agen Anda:

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "aidevops.amazonaws.com"
    },
    "Action": ["sts:AssumeRole", "sts:TagSession"],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<account-id>"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
      }
    }
  }
]
}

```

Langkah 3: Perbarui peran operator (IDC)

Jika Anda menggunakan Pusat Identitas IAM dengan DevOps Agen, perbarui setiap peran operator IDC.

1. Di konsol IAM, buka Peran dan cari **WebappIDC** untuk menemukan peran DevOps Agen IDC Anda (misalnya, DevOpsAgentRole-WebappIDC-<id>).
2. Untuk setiap peran IDC:

sebuah. Hapus kebijakan inline yang ada.

b. Pilih Tambahkan izin, Lampirkan kebijakan, dan lampirkan kebijakan AIDevOpsOperatorAppAccessPolicy terkelola.

c. Pilih tab Trust relationship dan pilih Edit trust policy. Ganti kebijakan kepercayaan dengan yang berikut, ganti ID akun, Wilayah, dan ID Ruang Agen Anda:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",

```

```

    "Principal": {
      "Service": "aidevops.amazonaws.com"
    },
    "Action": ["sts:AssumeRole", "sts:TagSession"],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<account-id>"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
      }
    }
  },
  {
    "Sid": "TrustedIdentityPropagation",
    "Effect": "Allow",
    "Principal": {
      "Service": "aidevops.amazonaws.com"
    },
    "Action": "sts:SetContext",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "<account-id>"
      },
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
      },
      "ForAllValues:ArnEquals": {
        "sts:RequestContextProviders": [
          "arn:aws:iam::aws:contextProvider/IdentityCenter"
        ]
      },
      "Null": {
        "sts:RequestContextProviders": "false"
      }
    }
  }
]
}

```

d. Buat kebijakan inline baru dengan izin berikut, ganti ID akun Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDevOpsAgentSSOAccess",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:DescribeInstance"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowDevOpsAgentIDCUserAccess",
      "Effect": "Allow",
      "Action": "identitystore:DescribeUser",
      "Resource": [
        "arn:aws:identitystore::<account-id>:identitystore/*",
        "arn:aws:identitystore:::user/*"
      ]
    }
  ]
}
```

Hubungkan kembali Pusat Identitas IAM (jika ada)

Ruang Agen yang dibuat selama pratinjau publik mungkin memiliki aplikasi Pusat Identitas IAM yang dikonfigurasi dengan cakupan akses yang sudah ketinggalan zaman. Untuk GA, ruang lingkup yang benar adalah **aidevops:read_write**. Jika aplikasi IAM Identity Center Anda memiliki scope (**awsaidevops:read_write**) sebelumnya, Anda harus memutuskan dan menghubungkan kembali IAM Identity Center.

Cara memeriksa cakupan aplikasi IAM Identity Center Anda

Jalankan perintah AWS CLI berikut untuk memeriksa cakupan pada aplikasi IAM Identity Center Anda. Anda dapat menemukan aplikasi ARN di konsol Pusat Identitas IAM di bawah Aplikasi.

```
aws sso-admin list-application-access-scopes \
  --application-arn arn:aws:sso::<account-id>:application/<instance-id>/<application-id>
```

Output harus menunjukkan ruang lingkup yang benar **aidevops:read_write**:

```
{
  "Scopes": [
    {
      "Scope": "aidevops:read_write"
    }
  ]
}
```

Jika ruang lingkup menunjukkan **awsaidevops:read_write**, itu sudah usang. Ikuti langkah-langkah di bawah ini untuk memperbaruinya.

Cara menghubungkan kembali Pusat Identitas IAM

Cakupan akses pada aplikasi Pusat Identitas IAM yang AWS dikelola tidak dapat diperbarui secara langsung. Anda harus memutuskan dan menyambung kembali:

1. Di konsol AWS DevOps Agen, buka Ruang Agen Anda dan pilih tab Akses.
2. Pilih Putuskan sambungan di samping konfigurasi Pusat Identitas IAM.
3. Konfirmasikan keputusan.
4. Pilih Connect untuk mengatur IAM Identity Center lagi. Layanan ini membuat aplikasi IAM Identity Center baru dengan cakupan yang benar.
5. Tetapkan kembali pengguna dan grup ke aplikasi baru di konsol Pusat Identitas IAM.

Important

Memutuskan sambungan akan menghapus obrolan pengguna individual dan riwayat artefak yang terkait dengan akun pengguna IAM Identity Center. Pengguna harus masuk lagi setelah koneksi ulang.

Verifikasi

Setelah menyelesaikan semua langkah:

1. Kembali ke konsol DevOps Agen dan verifikasi bahwa tidak ada kesalahan izin yang muncul di tab Akses Ruang Agen.

2. Uji aplikasi web operator untuk mengonfirmasi pemuatan dan fungsinya dengan benar.
3. Jika Anda menggunakan IDC, verifikasi bahwa pengguna dapat mengautentikasi dan mengakses pengalaman operator.

Pemecahan masalah

Izin ditolak kesalahan setelah migrasi

- Verifikasi `AI0psAssistantPolicy` bahwa telah dihapus dan `AIDevOpsAgentAccessPolicy` dilampirkan ke peran pemantauan.
- Verifikasi bahwa kebijakan inline lama telah dihapus dan `AIDevOpsOperatorAppAccessPolicy` dilampirkan ke peran operator.
- Periksa apakah kebijakan kepercayaan operator termasuk `sts:TagSession`.
- Konfirmasikan bahwa Anda mengganti semua nilai placeholder (`<account-id>`, `<region>`, `<agentspace-id>`) dengan nilai aktual.

Akun sekunder tidak berfungsi

- Setiap peran pemantauan akun sekunder harus diperbarui secara independen. Masuk ke setiap akun dan ulangi Langkah 1.

Kegagalan otentikasi IDC

- Verifikasi kebijakan kepercayaan IDC mencakup `sts:TagSession` pernyataan `sts:AssumeRole` dan `TrustedIdentityPropagation` pernyataan.
- Konfirmasikan kebijakan inline dengan `sso:ListInstances`, `sso:DescribeInstance`, dan `identitystore:DescribeUser` telah dibuat.

Riwayat obrolan sesuai permintaan hilang setelah migrasi

- Riwayat obrolan sesuai permintaan dari periode pratinjau publik tidak dapat diakses setelah rilis GA. Ini adalah perilaku yang diharapkan karena langkah-langkah keamanan yang ditingkatkan yang diperkenalkan di GA. Jurnal investigasi dan temuan dari pratinjau publik tidak terpengaruh.

AWS Pengaturan akses EKS

Anda dapat mengaktifkan AWS DevOps Agen untuk menyelidiki masalah di kluster Amazon EKS Anda dengan menjalankan `kubectl` perintah hanya-baca terhadap kluster publik dan pribadi. Anda dapat menghubungkan sejumlah kluster EKS ke Ruang Agen yang sama.

Setelah terhubung, agen dapat membantu mendiagnosis masalah operasional di klaster Anda — menjelaskan sumber daya, mengambil log pod, memeriksa peristiwa klaster, memeriksa kesehatan node, dan banyak lagi. Agen tidak dapat membuat, memodifikasi, atau menghapus sumber daya apa pun di klaster Anda.

Prasyarat

Sebelum mengatur akses EKS, pastikan bahwa mode otentikasi kluster EKS Anda menyertakan EKS API. Anda dapat memeriksanya di tab Access di [konsol Amazon EKS](#). Jika mode tidak menyertakan EKS API, pilih mode yang dilakukan sebelum melanjutkan.

Pengaturan

Langkah-langkah ini harus diselesaikan dari [konsol Amazon EKS](#) untuk setiap cluster yang ingin Anda buat entri akses. Anda dapat menemukan ARN peran IAM di Ruang Agen ([the section called “Membuat Ruang Agen”](#) lihat) di bawah Kemampuan > Cloud > Sumber Utama > Edit.

1. Buka tab Access. Jika mode Otentikasi sudah mengatakan EKS API, Anda dapat menambahkan entri akses. Jika tidak, pilih mode yang menyertakan EKS API.
2. Dari tab Access, buat entri akses IAM baru. Salin peran IAM sumber cloud utama Anda ARN dan masukkan sebagai prinsipal IAM untuk entri akses. Klik Berikutnya.
3. Pilih kebijakan AIops AssistantPolicy akses Amazon AWS Terkelola, dan pilih Cluster untuk cakupan akses. (Atau, jika Anda ingin agen hanya mengakses namespace tertentu, pilih Namespace Kubernetes yang diinginkan). Klik Add Policy, dan kemudian klik Next.
4. Tinjau perubahan dan konfirmasi bahwa kebijakan entri akses dan peran IAM yang benar telah dipilih, dan buat entri akses Anda dengan mengklik “Buat”.

Untuk memverifikasi bahwa akses EKS telah dikonfigurasi dengan benar, navigasikan ke Aplikasi Operator dan mulai penyelidikan baru, ajukan pertanyaan kepada agen tentang klaster Anda, seperti “daftar semua pod di namespace default” atau “tunjukkan peristiwa terbaru di klaster saya”.

Pemecahan masalah

Jika agen tidak dapat menjangkau klaster Anda, verifikasi bahwa entri akses menggunakan ARN peran IAM yang benar yang ditampilkan dalam dialog penyiapan dan kebijakan akses `AIopsAssistantPolicyAmazon` dilampirkan.

Menghubungkan Azure

Integrasi Azure memungkinkan AWS DevOps Agen untuk menyelidiki sumber daya di lingkungan Azure Anda dan menghubungkan penerapan DevOps pipeline Azure dengan insiden operasional. Dengan menghubungkan Azure, agen mendapatkan visibilitas ke infrastruktur Azure Anda dan dapat melakukan analisis akar penyebab di kedua AWS sumber daya Azure.

Integrasi Azure terdiri dari dua kemampuan independen:

- **Azure Resources** — Memungkinkan agen untuk menemukan dan menyelidiki sumber daya cloud Azure seperti mesin virtual, cluster Azure Kubernetes Service (AKS), database, dan komponen jaringan. Agen menggunakan Azure Resource Graph untuk menanyakan sumber daya Anda selama penyelidikan insiden.
- **Azure DevOps** - Memungkinkan agen untuk mengakses DevOps repositori Azure dan riwayat eksekusi pipeline. Agen dapat mengkorelasikan perubahan kode dan penerapan dengan insiden untuk membantu mengidentifikasi akar penyebab potensial.

Setiap kemampuan terdaftar di tingkat AWS akun dan kemudian dapat dikaitkan dengan Ruang Agen individu.

Metode pendaftaran

AWS DevOps Agen mendukung dua metode untuk menghubungkan ke Azure:

- **Persetujuan Admin** — Alur berbasis persetujuan yang efisien di mana Anda mengotorisasi aplikasi AWS DevOps Agen Entra di penyewa Azure Anda. Di konsol, ini muncul sebagai opsi Persetujuan Admin. Metode ini memerlukan login dengan akun yang memiliki izin untuk melakukan persetujuan admin di Microsoft Entra ID.
- **Pendaftaran Aplikasi** — Pendekatan yang dikelola sendiri di mana Anda membuat aplikasi Entra Anda sendiri dengan kredensi identitas federasi menggunakan Federasi Identitas Keluar. Di konsol, ini muncul sebagai opsi Pendaftaran Aplikasi. Metode ini cocok ketika Anda memerlukan kontrol lebih besar atas konfigurasi aplikasi atau ketika izin izin admin tidak tersedia.

Kedua metode memberikan kemampuan yang sama. Anda dapat menggunakan salah satu atau kedua metode dalam AWS akun yang sama.

Keterbatasan yang Sudah Diketahui

- **Persetujuan Admin:** satu AWS akun per penyewa Azure - Setiap penyewa Azure hanya dapat memiliki Aplikasi AWS DevOps Agen Entra yang terkait dengan satu akun pada satu AWS waktu. Untuk mengaitkan penyewa yang sama dengan AWS akun yang berbeda, Anda harus membatalkan pendaftaran yang ada terlebih dahulu.
- **Pendaftaran Aplikasi:** aplikasi unik per pendaftaran - Setiap Pendaftaran Aplikasi harus menggunakan aplikasi yang berbeda (ID klien). Anda tidak dapat mendaftarkan beberapa konfigurasi dengan ID klien yang sama.
- **Azure DevOps:** akses kode sumber - DevOps Integrasi Azure menyediakan akses ke riwayat eksekusi pipeline di mana pun kode sumber di-host. Namun, untuk mengakses kode sumber yang sebenarnya, repositori harus dihubungkan secara terpisah melalui penyedia sumber yang didukung (misalnya, [the section called “Menghubungkan GitHub”](#)). Kode sumber yang dihosting di Bitbucket tidak dapat diakses secara langsung melalui integrasi Azure DevOps .

Topik

- [the section called “Menghubungkan Sumber Daya Azure”](#)
- [the section called “Menghubungkan Azure DevOps”](#)

Menghubungkan Sumber Daya Azure

Integrasi Azure Resources memungkinkan AWS DevOps Agen menemukan dan menyelidiki sumber daya dalam langganan Azure Anda selama penyelidikan insiden. Agen menggunakan Azure Resource Graph untuk penemuan sumber daya dan dapat mengakses metrik, log, dan data konfigurasi di seluruh lingkungan Azure Anda.

Integrasi ini mengikuti proses dua langkah: daftarkan Azure di tingkat AWS akun, lalu kaitkan langganan Azure tertentu dengan Ruang Agen individual.

Prasyarat

Sebelum menghubungkan Azure Resources, pastikan Anda memiliki:

- Akses ke konsol AWS DevOps Agen
- Akun Azure dengan akses ke langganan target
- Untuk metode Persetujuan Admin: akun dengan izin untuk melakukan persetujuan admin di Microsoft Entra ID
- Untuk metode Pendaftaran Aplikasi: aplikasi Entra dengan izin untuk mengonfigurasi kredensi identitas federasi, dan Federasi Identitas [Keluar](#) diaktifkan di akun Anda AWS

Catatan: Anda juga dapat memulai pendaftaran dari dalam Ruang Agen. Arahkan ke Sumber sekunder, klik Tambah, dan pilih Azure. Jika Azure Cloud belum terdaftar, konsol memandu Anda melalui pendaftaran terlebih dahulu.

Mendaftarkan Azure Resources melalui Persetujuan Admin

Metode Persetujuan Admin menggunakan alur berbasis persetujuan dengan aplikasi yang dikelola AWS DevOps Agen.

Langkah 1: Mulai pendaftaran

1. Masuk ke Konsol AWS Manajemen dan arahkan ke konsol AWS DevOps Agen
2. Buka halaman Penyedia Kemampuan
3. Temukan bagian Azure Cloud dan klik Daftar
4. Pilih metode pendaftaran Persetujuan Admin

Langkah 2: Selesaikan Persetujuan Admin

1. Tinjau izin yang diminta
2. Klik untuk melanjutkan — Anda diarahkan ke halaman persetujuan admin Microsoft Entra
3. Masuk dengan akun utama pengguna yang memiliki izin untuk melakukan persetujuan admin
4. Meninjau dan memberikan persetujuan untuk aplikasi AWS DevOps Agen

Langkah 3: Otorisasi pengguna lengkap

1. Setelah persetujuan admin, Anda diminta untuk otorisasi pengguna untuk memverifikasi identitas Anda sebagai anggota penyewa yang berwenang
2. Masuk dengan akun milik penyewa Azure yang sama
3. Setelah otorisasi, Anda diarahkan kembali ke konsol AWS DevOps Agen dengan status sukses

Langkah 4: Tetapkan peran

Lihat [Menetapkan peran Azure](#) di bawah ini. Cari AWS DevOps Agen saat memilih anggota.

Mendaftarkan Azure Resources melalui Pendaftaran Aplikasi

Metode Pendaftaran Aplikasi menggunakan aplikasi Entra Anda sendiri dengan kredensi identitas federasi.

Langkah 1: Mulai pendaftaran

1. Di konsol AWS DevOps Agen, buka halaman Penyedia Kemampuan
2. Temukan bagian Azure Cloud dan klik Daftar
3. Pilih metode Pendaftaran Aplikasi

Langkah 2: Buat dan konfigurasi aplikasi Entra Anda

Ikuti petunjuk yang ditampilkan di konsol untuk:

1. Aktifkan Federasi Identitas Keluar di AWS akun Anda (di konsol IAM, buka Pengaturan akun → Federasi Identitas Keluar)
2. Buat aplikasi Entra di Microsoft Entra ID Anda, atau gunakan yang sudah ada
3. Konfigurasi kredensial identitas federasi pada aplikasi

Langkah 3: Berikan detail pendaftaran

Isi formulir pendaftaran dengan:

- ID Penyewa - Pengenal penyewa Azure Anda
- Nama Penyewa - Nama tampilan untuk penyewa
- ID Klien — ID aplikasi (klien) dari aplikasi Entra yang Anda buat
- Audiens - Pengidentifikasi audiens untuk kredensi federasi

Langkah 4: Buat peran IAM

Peran IAM akan dibuat secara otomatis saat Anda mengirimkan pendaftaran melalui konsol. Ini memungkinkan AWS DevOps Agen untuk mengambil kredensi dan memanggil `sts:GetWebIdentityToken`

Langkah 5: Tetapkan peran

Lihat [Menetapkan peran Azure](#) di bawah ini. Cari aplikasi Entra yang Anda buat saat memilih anggota.

Langkah 6: Lengkapi pendaftaran

1. Konfirmasikan konfigurasi di konsol AWS DevOps Agen
2. Klik Submit untuk menyelesaikan pendaftaran

Menetapkan peran Azure

Setelah pendaftaran, berikan akses baca aplikasi ke langganan Azure Anda. Langkah ini sama untuk metode Persetujuan Admin dan Pendaftaran Aplikasi.

1. Di Portal Azure, navigasikan ke langganan target Anda
2. Pergi ke Access Control (IAM)
3. Klik Menambahkan > Tambahkan tugas peran
4. Pilih peran Pembaca dan klik Berikutnya
5. Klik Pilih anggota, cari aplikasi (baik AWS DevOps Agen untuk Persetujuan Admin, atau aplikasi Entra Anda sendiri untuk Pendaftaran Aplikasi)
6. Pilih aplikasi dan klik Review + assign
7. (Opsional) Untuk mengaktifkan agen mengakses kluster Azure Kubernetes Service (AKS), selesaikan pengaturan akses AKS berikut.

Persyaratan Keamanan: Prinsipal layanan harus ditetapkan hanya peran Pembaca (dan secara opsional peran hanya-baca AKS yang tercantum di bawah). Peran Reader berfungsi sebagai batas keamanan yang membatasi agen untuk operasi read-only dan membatasi dampak serangan injeksi prompt tidak langsung. Menetapkan peran dengan izin tulis atau tindakan secara signifikan meningkatkan radius ledakan injeksi cepat dan dapat mengakibatkan kompromi sumber daya Azure. AWS DevOps Agen hanya melakukan operasi baca. Agen tidak memodifikasi, membuat, atau menghapus sumber daya Azure.

Pengaturan akses AKS (opsional)

Langkah 1: Akses tingkat Azure Resource Manager (ARM)

Tetapkan Azure Kubernetes Service Cluster User Role ke aplikasi.

Di Portal Azure, buka Langganan → pilih langganan → Kontrol Akses (IAM) → Tambahkan penetapan peran → pilih Peran Pengguna Cluster Layanan Azure Kubernetes → tetapkan ke aplikasi (baik AWS DevOps Agen untuk Persetujuan Admin, atau aplikasi Entra Anda sendiri untuk Pendaftaran Aplikasi).

Ini mencakup semua kluster AKS dalam langganan. Untuk cakupan ke cluster tertentu, tetapkan pada kelompok sumber daya atau tingkat cluster individu sebagai gantinya.

Langkah 2: Akses API Kubernetes

Pilih satu opsi berdasarkan konfigurasi otentikasi kluster Anda:

Opsi A: Kontrol Akses Berbasis Peran Azure (RBAC) untuk Kubernetes (disarankan)

1. Aktifkan Azure RBAC di cluster jika belum diaktifkan: Azure Portal → Kluster AKS → Pengaturan → Konfigurasi keamanan → Otentikasi dan otorisasi → pilih Azure RBAC
2. Tetapkan peran hanya-baca: Azure Portal → Langganan → pilih langganan → Kontrol Akses (IAM) → Tambahkan penetapan peran → pilih Azure Kubernetes Service RBAC Reader → tetapkan ke aplikasi

Ini mencakup semua kluster AKS dalam langganan.

Opsi B: Direktori Aktif Azure (Azure AD) +Kubernetes RBAC

Gunakan ini jika kluster Anda sudah menggunakan konfigurasi autentikasi Azure AD default dan Anda memilih untuk tidak mengaktifkan Azure RBAC. Ini membutuhkan pengaturan per `clusterkubect1`.

1. Simpan manifes berikut sebagaidevops-agent-reader.yaml:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: devops-agent-reader
rules:
```

```

- apiGroups: [""]
  resources: ["namespaces", "pods", "pods/log", "services", "events", "nodes"]
  verbs: ["get", "list"]
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets", "statefulsets", "daemonsets"]
  verbs: ["get", "list"]
- apiGroups: ["metrics.k8s.io"]
  resources: ["pods", "nodes"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: devops-agent-reader-binding
subjects:
- kind: User
  name: "<SERVICE_PRINCIPAL_OBJECT_ID>"
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: devops-agent-reader
  apiGroup: rbac.authorization.k8s.io

```

1. Ganti <SERVICE_PRINCIPAL_OBJECT_ID> dengan Object ID kepala layanan Anda. Untuk menemukannya: Azure Portal → Entra ID → Aplikasi Perusahaan → cari nama aplikasi (baik AWS DevOps Agen untuk Persetujuan Admin, atau aplikasi Entra Anda sendiri untuk Pendaftaran Aplikasi).
2. Terapkan untuk setiap cluster:

```

az aks get-credentials --resource-group <rg> --name <cluster-name>
kubectl apply -f devops-agent-reader.yaml

```

Catatan: Cluster yang menggunakan akun lokal saja (tanpa Azure AD) tidak didukung. Sebaiknya aktifkan integrasi Azure AD di klaster Anda untuk menggunakan fitur ini.

Peran kustom yang paling tidak memiliki hak istimewa (opsional)

Untuk kontrol akses yang lebih ketat, Anda dapat membuat peran Azure kustom yang hanya digunakan oleh penyedia sumber daya yang digunakan AWS DevOps Agen, alih-alih peran Pembaca yang luas:

```
{
  "Name": "AWS DevOps Agent - Azure Reader",
  "Description": "Least-privilege read-only access for AWS DevOps Agent incident investigations.",
  "Actions": [
    "Microsoft.AlertsManagement/*/read",
    "Microsoft.Compute/*/read",
    "Microsoft.ContainerRegistry/*/read",
    "Microsoft.ContainerService/*/read",
    "Microsoft.ContainerService/managedClusters/commandResults/read",
    "Microsoft.DocumentDB/*/read",
    "Microsoft.Insights/*/read",
    "Microsoft.KeyVault/vaults/read",
    "Microsoft.ManagedIdentity/*/read",
    "Microsoft.Monitor/*/read",
    "Microsoft.Network/*/read",
    "Microsoft.OperationalInsights/*/read",
    "Microsoft.ResourceGraph/resources/read",
    "Microsoft.ResourceHealth/*/read",
    "Microsoft.Resources/*/read",
    "Microsoft.Sql/*/read",
    "Microsoft.Storage/*/read",
    "Microsoft.Web/*/read"
  ],
  "NotActions": [],
  "DataActions": [],
  "NotDataActions": [],
  "AssignableScopes": [
    "/subscriptions/{your-subscription-id}"
  ]
}
```

Mengaitkan langganan dengan Agent Space

Setelah mendaftarkan Azure di tingkat akun, kaitkan langganan tertentu dengan Ruang Agen Anda:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Kemampuan
3. Di bagian Sumber sekunder, klik Tambah
4. Pilih Azure
5. Berikan ID Langganan untuk langganan Azure yang ingin Anda kaitkan

6. Klik Tambah untuk menyelesaikan asosiasi

Anda dapat mengaitkan beberapa langganan dengan Ruang Agen yang sama untuk memberikan visibilitas agen di seluruh lingkungan Azure Anda.

Mengelola koneksi Azure Resources

- Melihat langganan yang terhubung — Di tab Kemampuan, bagian Sumber sekunder mencantumkan semua langganan Azure yang terhubung.
- Menghapus langganan — Untuk memutuskan sambungan langganan dari Ruang Agen, pilih langganan tersebut di daftar Sumber sekunder dan klik Hapus. Ini tidak mempengaruhi pendaftaran tingkat akun.
- Menghapus pendaftaran — Untuk menghapus pendaftaran Azure Cloud sepenuhnya, buka halaman Penyedia Kemampuan dan hapus pendaftaran. Semua asosiasi Ruang Agen harus dihapus terlebih dahulu.

Menghubungkan Azure DevOps

DevOps Integrasi Azure memungkinkan AWS DevOps Agen mengakses repositori dan riwayat eksekusi pipeline di organisasi Azure Anda. DevOps Agen dapat mengkorelasikan perubahan kode dan penerapan dengan insiden operasional untuk membantu mengidentifikasi akar penyebab potensial.

Catatan: DevOps Pipeline Azure dapat menggunakan kode sumber dari Azure Repos,, GitHub atau Bitbucket. DevOps Integrasi Azure menyediakan akses ke riwayat eksekusi pipeline terlepas dari penyedia sumbernya. Namun, untuk mengakses kode sumber yang sebenarnya selama penyelidikan, repositori harus dihubungkan secara terpisah melalui integrasi yang didukung seperti [the section called “Menghubungkan GitHub”](#). Kode sumber di Bitbucket tidak dapat diakses secara langsung melalui integrasi ini.

Integrasi ini mengikuti proses dua langkah: daftarkan Azure DevOps di tingkat AWS akun, lalu kaitkan proyek tertentu dengan Ruang Agen individu.

Prasyarat

Sebelum menghubungkan Azure DevOps, pastikan Anda memiliki:

- Akses ke konsol AWS DevOps Agen

- DevOps Organisasi Azure dengan setidaknya satu proyek yang berisi repositori dan riwayat pipa
- Izin untuk menambahkan pengguna ke organisasi DevOps Azure Anda
- Untuk metode Persetujuan Admin: akun dengan izin untuk melakukan persetujuan admin di Microsoft Entra ID
- Untuk metode Pendaftaran Aplikasi: aplikasi Entra dengan izin untuk mengonfigurasi kredensi identitas federasi, dan Federasi Identitas [Keluar](#) diaktifkan di akun Anda AWS

Catatan: Anda juga dapat memulai pendaftaran dari dalam Ruang Agen. Arahkan ke bagian Pipelines, klik Tambah, dan pilih DevOpsAzure. Jika Azure DevOps belum terdaftar, konsol memandu Anda melalui pendaftaran terlebih dahulu.

Mendaftarkan Azure DevOps melalui Persetujuan Admin

Metode Persetujuan Admin menggunakan alur berbasis persetujuan dengan aplikasi yang dikelola AWS DevOps Agen.

Langkah 1: Mulai pendaftaran

1. Masuk ke Konsol AWS Manajemen dan arahkan ke konsol AWS DevOps Agen
2. Buka halaman Penyedia Kemampuan
3. Temukan DevOps bagian Azure dan klik Daftar
4. Masukkan nama DevOps organisasi Azure Anda saat diminta

Langkah 2: Selesaikan Persetujuan Admin

1. Klik untuk melanjutkan - Anda diarahkan ke halaman persetujuan admin Microsoft Entra
2. Masuk dengan akun utama pengguna yang memiliki izin untuk melakukan persetujuan admin
3. Meninjau dan memberikan persetujuan untuk aplikasi AWS DevOps Agen

Langkah 3: Otorisasi pengguna lengkap

1. Setelah persetujuan admin, Anda diminta untuk otorisasi pengguna untuk memverifikasi identitas Anda sebagai anggota penyewa yang berwenang
2. Masuk dengan akun milik penyewa Azure yang sama
3. Setelah otorisasi, Anda dialihkan kembali ke konsol AWS DevOps Agen dengan status sukses

Langkah 4: Berikan akses di Azure DevOps

Lihat [Pemberian akses di Azure di bawah DevOps](#) ini. Cari AWS DevOps Agen saat menambahkan pengguna.

Mendaftarkan Azure DevOps melalui Pendaftaran Aplikasi

Pendaftaran Aplikasi dibagikan antara Azure Resources dan DevOps Azure. Jika Anda telah menyelesaikan Pendaftaran Aplikasi untuk Azure Resources, Anda dapat melompat ke [Pemberian akses di Azure](#). DevOps

Langkah 1: Mulai Pendaftaran Aplikasi ADO

1. Di konsol AWS DevOps Agen, buka halaman Penyedia Kemampuan
2. Temukan bagian Azure Cloud dan klik Daftar
3. Pilih metode Pendaftaran Aplikasi

Langkah 2: Buat dan konfigurasi aplikasi Entra Anda

Ikuti petunjuk yang ditampilkan di konsol untuk:

1. Aktifkan Federasi Identitas Keluar di AWS akun Anda (di konsol IAM, buka Pengaturan akun → Federasi Identitas Keluar)
2. Buat aplikasi Entra di Microsoft Entra ID Anda, atau gunakan yang sudah ada
3. Konfigurasi kredensial identitas federasi pada aplikasi

Langkah 3: Berikan detail pendaftaran

Isi formulir pendaftaran dengan:

- ID Penyewa — Pengenal penyewa Azure Anda
- Nama Penyewa - Nama tampilan untuk penyewa
- ID Klien — ID aplikasi (klien) dari aplikasi Entra
- Audiens - Pengidentifikasi audiens untuk kredensial federasi

Langkah 4: Buat peran IAM

Peran IAM akan dibuat secara otomatis saat Anda mengirimkan pendaftaran melalui konsol. Ini memungkinkan AWS DevOps Agen untuk mengambil kredensi dan memanggil.

```
sts:GetWebIdentityToken
```

Langkah 5: Lengkapi pendaftaran

1. Konfirmasikan konfigurasi di konsol AWS DevOps Agen
2. Klik Submit untuk menyelesaikan pendaftaran

Langkah 6: Berikan akses di Azure DevOps

Lihat [Pemberian akses di Azure di bawah DevOps](#) ini. Cari aplikasi Entra yang Anda buat selama Pendaftaran Aplikasi saat menambahkan pengguna.

Memberikan akses di Azure DevOps

Setelah pendaftaran, berikan akses aplikasi ke DevOps organisasi Azure Anda. Langkah ini sama untuk metode Persetujuan Admin dan Pendaftaran Aplikasi.

1. Di Azure DevOps, buka Pengaturan Organisasi > Pengguna> Tambah Pengguna
2. Cari aplikasi (baik AWS DevOps Agen untuk Persetujuan Admin, atau aplikasi Entra Anda sendiri untuk Pendaftaran Aplikasi)
3. Mengatur tingkat akses ke Basic
4. Di bawah Tambahkan ke proyek, pilih proyek yang ingin diakses agen
5. Di bawah DevOps Grup Azure, pilih Pembaca Proyek
6. Klik Tambah untuk menyelesaikan

Persyaratan Keamanan: Tetapkan hanya grup Pembaca Proyek. Akses hanya-baca berfungsi sebagai batas keamanan yang membatasi agen untuk operasi hanya-baca dan membatasi dampak serangan injeksi prompt tidak langsung. Menetapkan grup dengan izin tulis atau tindakan secara signifikan meningkatkan radius ledakan injeksi cepat dan dapat mengakibatkan kompromi sumber daya Azure. DevOps

Mengaitkan proyek dengan Agen Ruang

Setelah mendaftarkan Azure DevOps di tingkat akun, kaitkan proyek tertentu dengan Ruang Agen Anda:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Kemampuan
3. Di bagian Pipelines, klik Tambah
4. Pilih Azure DevOps dari daftar penyedia yang tersedia
5. Pilih proyek dari dropdown proyek yang tersedia
6. Klik Tambah untuk menyelesaikan asosiasi

Mengelola koneksi Azure DevOps

- Melihat proyek yang terhubung — Di tab Kemampuan, bagian Pipelines mencantumkan semua proyek Azure DevOps yang terhubung.
- Menghapus proyek — Untuk memutuskan sambungan proyek dari Ruang Agen, pilih di bagian Pipelines dan klik Hapus.
- Menghapus pendaftaran — Untuk menghapus DevOps pendaftaran Azure sepenuhnya, buka halaman Penyedia Kemampuan dan hapus pendaftaran. Semua asosiasi Ruang Agen harus dihapus terlebih dahulu.

Menghubungkan ke CI/CD jaringan pipa

Integrasi pipa CI/CD memungkinkan AWS DevOps Agen untuk memantau penerapan dan mengkorelasikan perubahan kode dengan insiden operasional selama penyelidikan. Dengan menghubungkan CI/CD penyedia Anda, agen dapat melacak peristiwa penyebaran dan mengaitkannya dengan AWS sumber daya untuk membantu mengidentifikasi akar penyebab potensial selama respons insiden.

AWS DevOps Agen mendukung integrasi dengan CI/CD platform populer melalui proses dua langkah:

1. Registrasi tingkat akun — Daftarkan CI/CD penyedia Anda sekali di tingkat akun AWS
2. Koneksi Ruang Agen - Hubungkan proyek atau repositori tertentu ke Ruang Agen individual berdasarkan kebutuhan organisasi Anda

Pendekatan ini memungkinkan Anda untuk berbagi pendaftaran CI/CD penyedia di beberapa Ruang Agen sambil mempertahankan kontrol terperinci atas proyek mana yang dipantau oleh setiap ruang.

CI/CD Penyedia yang didukung

AWS DevOps Agen mendukung CI/CD platform berikut:

- GitHub— Connect repositori [GitHubdari.com](https://github.com) menggunakan aplikasi AWS DevOps Agen GitHub .
- GitLab— Connect project [GitLabdari.com](https://gitlab.com), GitLab instans terkelola, atau penerapan GitLab self-host yang dapat diakses publik.

Topik

- [the section called “Menghubungkan GitHub”](#)
- [the section called “Menghubungkan GitLab”](#)

Menghubungkan GitHub

GitHub integrasi memungkinkan AWS DevOps Agen untuk mengakses repositori kode dan menerima peristiwa penyebaran selama investigasi insiden. Integrasi ini mengikuti proses dua langkah: pendaftaran tingkat akun GitHub, diikuti dengan menghubungkan repositori tertentu ke Ruang Agen individu.

AWS DevOps Agen mendukung GitHub instans.com (SaaS) dan Server GitHub Perusahaan (yang dihosting sendiri).

Prasyarat

Sebelum menghubungkan GitHub, pastikan Anda memiliki:

- Akses ke konsol admin AWS DevOps Agen
- Akun GitHub pengguna atau organisasi dengan izin admin
- Otorisasi untuk menginstal GitHub aplikasi di akun atau organisasi

Untuk GitHub Enterprise Server, Anda juga membutuhkan:

- Instans GitHub Enterprise Server (versi 3.x atau yang lebih baru) dapat diakses melalui HTTPS

- URL HTTPS dari instance Server GitHub Perusahaan Anda (misalnya, `https://github.example.com`)
- (Opsional) Koneksi pribadi, jika instans Server GitHub Perusahaan Anda tidak dapat diakses publik

Mendaftar GitHub (tingkat akun)

GitHub terdaftar di tingkat AWS akun dan dibagikan di antara semua Ruang Agen di akun itu. Anda hanya perlu mendaftar GitHub sekali per AWS akun.

Langkah 1: Arahkan ke penyedia pipa

1. Masuk ke Konsol AWS Manajemen
2. Arahkan ke konsol AWS DevOps Agen
3. Buka tab Kemampuan
4. Di bagian Pipeline, klik Tambah
5. Pilih GitHub dari daftar penyedia yang tersedia

Jika GitHub belum terdaftar, Anda akan diminta untuk mendaftarkannya terlebih dahulu.

Langkah 2: Pilih jenis koneksi

Pada layar “Daftarkan GitHub Akun/Organisasi”, pilih apakah Anda terhubung sebagai pengguna atau organisasi:

- Pengguna — GitHub Akun pribadi Anda dengan nama pengguna dan profil
- Organisasi — GitHub Akun bersama tempat banyak orang dapat berkolaborasi di banyak proyek sekaligus

Jika Anda terhubung ke instance GitHub Enterprise Server, centang kotak centang Use GitHub Enterprise Server dan masukkan URL HTTPS instance Anda (misalnya, `https://github.example.com`).

Jika instans Server GitHub Perusahaan Anda tidak dapat diakses publik, Anda dapat mengonfigurasi sambungan pribadi secara opsional agar AWS DevOps Agen dapat menjangkau instans Anda dengan aman. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#).

Note

Jangan sertakan `/api/v3` atau trailing path apa pun di URL — masukkan hanya URL dasar.

Langkah 3: Siapkan GitHub Aplikasi

Klik Kirim untuk memulai proses penyiapan aplikasi. Langkah selanjutnya berbeda tergantung pada apakah Anda terhubung GitHub ke.com atau GitHub Enterprise Server.

Untuk GitHub .com

1. Anda akan diarahkan GitHub untuk menginstal GitHub aplikasi AWS DevOps Agen.
2. Pilih akun atau organisasi mana yang akan menginstal aplikasi.
3. Aplikasi ini memungkinkan AWS DevOps Agen untuk menerima peristiwa dari repositori yang terhubung, termasuk peristiwa penerapan.

Untuk Server GitHub Perusahaan

GitHub Enterprise Server menggunakan alur Manifes GitHub Aplikasi, yang secara otomatis menyiapkan GitHub Aplikasi baru di instans Anda. Ini melibatkan dua pengalihan ke instance Server GitHub Enterprise Anda.

1. Browser Anda akan diarahkan ke halaman “Buat GitHub Aplikasi” instance GitHub Enterprise Server Anda.
2. Anda akan melihat nama aplikasi yang sudah diisi sebelumnya. Jangan ragu untuk mengubah nama sesuai kebutuhan. Klik Buat GitHub Aplikasi.
3. Anda akan diarahkan kembali ke AWS DevOps Agen, yang menukar kode manifes dengan kredensi aplikasi.

Langkah 4: Pilih repositori dan selesaikan instalasi

1. Anda akan melihat halaman Instal & Otorisasi untuk GitHub Aplikasi.
2. Pilih repositori mana yang memungkinkan aplikasi mengakses:
 - Semua repositori - Berikan akses ke semua repositori saat ini dan masa depan
 - Hanya pilih repositori - Pilih repositori tertentu dari akun atau organisasi Anda

3. Klik Instal & Otorisasi.
4. Anda akan diarahkan kembali ke konsol AWS DevOps Agen, di mana GitHub akan muncul sebagai terdaftar di tingkat akun.

Menghubungkan repositori ke Ruang Agen

Setelah mendaftar GitHub di tingkat akun, Anda dapat menghubungkan repositori tertentu ke Ruang Agen individual:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Kemampuan
3. Di bagian Pipeline, klik Tambah
4. Pilih GitHub dari daftar penyedia yang tersedia
5. Pilih subset repositori yang relevan dengan Ruang Agen ini
6. Klik Tambah untuk menyelesaikan koneksi

Anda dapat menghubungkan kumpulan repositori yang berbeda ke Ruang Agen yang berbeda berdasarkan kebutuhan organisasi Anda.

Memahami GitHub aplikasi

GitHub Aplikasi AWS DevOps Agen:

- Meminta akses ke repositori Anda — Anda dapat meninjau izin tertentu selama instalasi Aplikasi GitHub
- Menerima peristiwa penyebaran dan acara repositori lainnya
- Memungkinkan AWS DevOps Agen untuk mengkorelasikan perubahan kode dengan insiden operasional
- Dapat di-uninstall kapan saja melalui pengaturan Anda GitHub

Untuk Server GitHub Perusahaan, GitHub Aplikasi secara otomatis dibuat pada instans Anda saat pendaftaran. Anda dapat mengelola akses repositori aplikasi atau menghapus instalannya melalui Pengaturan > Aplikasi > Aplikasi Terinstal. GitHub Untuk menghapus definisi aplikasi sepenuhnya, buka Pengaturan > Pengaturan pengembang > GitHub Aplikasi.

GitHub Pembaruan izin aplikasi

AWS DevOps Agen dapat meminta pembaruan izin setelah Anda menginstal GitHub Aplikasi untuk mendukung fitur baru. Ketika ini terjadi:

1. Anda akan menerima pemberitahuan dari GitHub mengenai permintaan pembaruan izin.
2. Tinjau detail pembaruan untuk memahami izin baru apa yang diminta.
3. Terima permintaan untuk memberikan izin yang diperbarui.

Tidak ada perubahan yang diperlukan dalam layanan atau aplikasi Anda. Setelah Anda menerima izin yang diperbarui, token akses instalasi berikutnya yang diminta AWS DevOps Agen GitHub akan secara otomatis menyertakan izin baru.

Note

Sampai Anda menerima pembaruan izin, AWS DevOps Agen terus beroperasi dengan izin yang diberikan sebelumnya. Kemampuan baru yang bergantung pada izin yang diperbarui tidak akan tersedia sampai Anda menyetujui permintaan tersebut.

Mengelola GitHub koneksi

- Memperbarui akses repositori — Untuk mengubah repositori mana yang dapat diakses GitHub aplikasi, buka pengaturan GitHub akun atau organisasi Anda (atau pengaturan instans Server GitHub Perusahaan Anda), navigasikan ke aplikasi yang diinstal, dan ubah konfigurasi GitHub aplikasi Agen. AWS DevOps
- Melihat repositori yang terhubung - Di konsol AWS DevOps Agen, pilih Ruang Agen Anda dan buka tab Kemampuan untuk melihat repositori yang terhubung di bagian Pipeline.
- Menghapus GitHub koneksi - Untuk memutuskan sambungan GitHub dari Ruang Agen, pilih koneksi di bagian Pipeline dan klik Hapus. Untuk menghapus instalasi GitHub aplikasi sepenuhnya, hapus instalannya dari pengaturan GitHub akun atau organisasi Anda. Untuk Server GitHub Perusahaan, karena GitHub Aplikasi dibuat langsung pada instans Anda saat pendaftaran, Anda dapat membersihkan aplikasi secara opsional sepenuhnya dengan melakukan kedua hal berikut:
 - Copot pemasangan aplikasi — Buka Pengaturan > Aplikasi > GitHub Aplikasi Terinstal, klik Konfigurasi pada aplikasi, lalu hapus instalannya.

- Hapus aplikasi — Buka Pengaturan > Pengaturan pengembang > GitHub Aplikasi, pilih aplikasi, buka tab Lanjutan, dan pilih Hapus GitHub Aplikasi. Peringatan: Menghapus GitHub Aplikasi bersifat permanen dan tidak dapat dibatalkan. Jika Anda menghapusnya, Anda harus mendaftarkan ulang GitHub Enterprise Server dari awal di konsol AWS DevOps Agen untuk membuat aplikasi baru.

Menghubungkan GitLab

GitLab integrasi memungkinkan AWS DevOps Agen untuk memantau penyebaran dari GitLab Pipelines untuk menginformasikan penyelidikan kausal selama respons insiden. Integrasi ini mengikuti proses dua langkah: pendaftaran tingkat akun GitLab, diikuti dengan menghubungkan proyek tertentu ke Ruang Agen individu.

Mendaftar GitLab (tingkat akun)

GitLab terdaftar di tingkat AWS akun dan dibagikan di antara semua Ruang Agen di akun itu. Ruang Agen Individu kemudian dapat memilih proyek spesifik mana yang berlaku untuk Ruang Agen mereka.

Langkah 1: Arahkan ke penyedia pipa

1. Masuk ke Konsol AWS Manajemen
2. Arahkan ke konsol AWS DevOps Agen
3. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
4. Temukan GitLab di bagian Penyedia yang tersedia di bawah Pipeline dan klik Daftar

Langkah 2: Konfigurasi GitLab koneksi


Pada halaman GitLab pendaftaran, konfigurasi yang berikut:

Jenis koneksi - Pilih apakah Anda terhubung sebagai pribadi atau grup:

- Pribadi (default) - Akun GitLab pengguna individual Anda dengan nama pengguna dan profil
- Grup — Di GitLab, Anda menggunakan grup untuk mengelola satu atau lebih proyek terkait secara bersamaan

GitLab tipe instans - Pilih jenis GitLab instans yang Anda sambungkan:

- GitLab.com (default) — GitLab Pelayanan publik
- Hosting mandiri yang dapat diakses publik GitLab - Centang kotak Use GitLab self hosted endpoint dan berikan URL ke instans Anda GitLab

 Note

Saat ini, hanya GitLab instans yang dapat diakses publik yang didukung.

Token akses — Berikan token akses GitLab pribadi:

1. Di tab browser terpisah, masuk ke GitLab akun Anda
2. Arahkan ke pengaturan pengguna Anda dan pilih Access Tokens
3. Buat token akses pribadi baru dengan izin berikut:
 - `read_repository`— Diperlukan untuk mengakses konten repositori
 - `read_virtual_registry`— Diperlukan untuk mengakses informasi registri virtual
 - `read_registry`— Diperlukan untuk mengakses informasi registri
 - `api`— Diperlukan untuk akses API baca dan tulis
 - `self_rotate`- Diperlukan untuk memutar token. Fitur ini saat ini tidak didukung oleh AWS DevOps Agen tetapi akan didukung di kemudian hari. Menambahkan sekarang mencegah kebutuhan untuk membuat token baru di masa depan.
4. Setel kedaluwarsa token ke maksimum 365 hari dari tanggal saat ini
5. Salin token yang dihasilkan
6. Kembali ke konsol AWS DevOps Agen
7. Tempelkan token ke bidang “Token Akses”

Langkah 3: Selesaikan pendaftaran

(Opsional) Tag - Tambahkan AWS tag ke GitLab pendaftaran untuk tujuan organisasi.

Klik Berikutnya untuk meninjau konfigurasi Anda, lalu klik Kirim untuk menyelesaikan proses GitLab pendaftaran. Sistem akan memvalidasi token akses Anda dan membuat koneksi.

Menghubungkan proyek ke Ruang Agen

Setelah mendaftar GitLab di tingkat akun, Anda dapat menghubungkan proyek tertentu ke Ruang Agen individu:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen Anda
2. Buka tab Kemampuan
3. Di bagian Pipeline, klik Tambah
4. Pilih GitLab dari daftar penyedia yang tersedia
5. Pilih GitLab proyek yang relevan dengan Ruang Agen Anda
6. Klik Simpan

AWS DevOps Agen akan memantau proyek-proyek ini untuk penyebaran dari GitLab Pipelines untuk menginformasikan penyelidikan kausal.

Mengelola GitLab koneksi

- Memperbarui token akses — Jika token akses Anda kedaluwarsa atau perlu diperbarui, Anda dapat memperbaruinya di konsol AWS DevOps Agen dengan memodifikasi GitLab pendaftaran di tingkat akun.
- Melihat proyek yang terhubung — Di konsol AWS DevOps Agen, pilih Ruang Agen Anda dan buka tab Kemampuan untuk melihat proyek yang terhubung di bagian Pipeline.
- Menghapus GitLab koneksi - Untuk memutuskan GitLab proyek dari Ruang Agen, pilih koneksi di bagian Pipeline dan klik Hapus. Untuk menghapus GitLab pendaftaran sepenuhnya, hapus dari semua Ruang Agen terlebih dahulu, lalu hapus pendaftaran di tingkat akun.

Menghubungkan Server MCP

Server Model Context Protocol (MCP) memperluas kemampuan investigasi AWS DevOps Agen dengan menyediakan akses ke data dari alat observabilitas eksternal Anda, sistem pemantauan khusus, dan sumber data operasional. Panduan ini menjelaskan cara menghubungkan server MCP ke AWS DevOps Agen.

Persyaratan

Sebelum menghubungkan server MCP, pastikan server Anda memenuhi persyaratan ini:

- Protokol transport HTTP Streamable — Hanya server MCP yang mengimplementasikan protokol transport HTTP Streamable yang didukung.
- Dukungan otentikasi — Server MCP Anda harus mendukung salah satu metode otentikasi berikut: OAuth 2.0 (Client Credentials atau 3LO), kunci API/otentikasi berbasis token, atau Signature Version 4 (SigV4). AWS

Pertimbangan keamanan

Saat menghubungkan server MCP ke AWS DevOps Agen, pertimbangkan aspek keamanan berikut:

- Tool allowlisting — Anda harus mengizinkan daftar hanya alat khusus yang dibutuhkan Ruang Agen Anda, daripada mengekspos semua alat dari server MCP Anda. Lihat [Mengonfigurasi alat MCP di Ruang Agen](#) untuk mengetahui cara mengizinkan alat daftar per Ruang Agen.

Harap dicatat bahwa panjang alat maksimum dari setiap alat MCP adalah 64.

- Risiko injeksi cepat - Server MCP khusus dapat menimbulkan risiko tambahan serangan injeksi yang cepat. Lihat [Prompt injection protection: AWS DevOps Agent Security](#) untuk informasi lebih lanjut.
- Alat dan akses hanya-baca — Hanya izinkan daftar alat MCP hanya-baca dan pastikan bahwa kredensial otentikasi hanya diizinkan akses hanya-baca.

Lihat [AWS DevOps Agen Keamanan](#) untuk informasi lebih lanjut tentang injeksi cepat dan model tanggung jawab bersama.

Note

Jika server MCP Anda berada di jaringan pribadi, lihat [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#)

Mendaftarkan server MCP (tingkat akun)

Server MCP terdaftar di tingkat AWS akun dan dibagikan di antara semua Ruang Agen di akun itu. Individual Agent Spaces kemudian dapat memilih alat spesifik mana yang mereka butuhkan dari setiap server MCP.

Langkah 1: Detail server MCP

1. Masuk ke Konsol AWS Manajemen
2. Arahkan ke konsol AWS DevOps Agen
3. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
4. Temukan MCP Server di bagian Penyedia yang tersedia dan klik Daftar
5. Pada halaman detail server MCP, masukkan informasi berikut:
 - Nama - Masukkan nama deskriptif untuk server MCP Anda
 - URL Endpoint - Masukkan URL HTTPS lengkap dari titik akhir server MCP Anda
 - Deskripsi (opsional) - Tambahkan deskripsi untuk membantu mengidentifikasi tujuan server
 - Aktifkan Pendaftaran Klien Dinamis - Pilih kotak centang ini jika Anda ingin mengizinkan AWS DevOps Agen untuk secara otomatis mendaftar dengan server otorisasi server MCP Anda
 - Connect to endpoint menggunakan koneksi pribadi — Pilih kotak centang ini jika Anda ingin AWS DevOps Agen membuat permintaan ke server MCP Anda secara pribadi. Anda dapat memilih koneksi pribadi yang ada atau membuat yang baru. Jika Anda menggunakan OAuth otentikasi, koneksi pribadi berlaku untuk titik akhir server MCP dan titik akhir pertukaran token. Pastikan koneksi pribadi dikonfigurasi dengan alamat host yang dapat merutekan lalu lintas ke kedua titik akhir. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#).
6. Klik Berikutnya

Note

URL endpoint server MCP akan ditampilkan di AWS CloudTrail log di akun Anda.

Langkah 2: Aliran otorisasi

Pilih metode otentikasi untuk server MCP Anda:

OAuth Client Client Client - Jika server MCP Anda menggunakan alur Client OAuth Client Client:

1. Pilih Kredensial OAuth Klien
2. Klik Berikutnya

OAuth 3LO (Three-Legged OAuth) - Jika server MCP Anda menggunakan OAuth 3LO untuk otentikasi:

1. Pilih OAuth 3LO
2. Klik Berikutnya

Kunci API - Jika server MCP Anda menggunakan otentikasi kunci API:

1. Pilih Kunci API
2. Klik Berikutnya

AWS SiGv4 - Jika server MCP Anda menggunakan otentikasi AWS Signature Version 4:

1. Pilih AWS SiGv4
2. Klik Berikutnya

Langkah 3: Konfigurasi otorisasi

Konfigurasi parameter otorisasi tambahan berdasarkan metode otentikasi yang dipilih:

Untuk Kredensi OAuth Klien:

1. ID Klien — Masukkan ID klien OAuth klien
2. Rahasia Klien — Masukkan rahasia OAuth klien klien
3. URL Exchange — Masukkan URL titik akhir pertukaran OAuth token
4. Parameter Pertukaran — Masukkan parameter pertukaran OAuth token untuk otentikasi dengan layanan
5. Tambahkan Lingkup - Tambahkan OAuth cakupan untuk otentikasi
6. Klik Berikutnya

Untuk OAuth 3LO:

1. ID Klien — Masukkan ID klien OAuth klien
2. Rahasia Klien — Masukkan rahasia OAuth klien dari klien jika diperlukan oleh OAuth klien Anda
3. URL Exchange — Masukkan URL titik akhir pertukaran OAuth token

4. URL Otorisasi - Masukkan URL titik akhir OAuth otorisasi
5. Code Challenge Support - Pilih kotak centang ini jika OAuth klien Anda mendukung tantangan kode
6. Tambahkan Lingkup - Tambahkan OAuth cakupan untuk otentikasi
7. Klik Berikutnya

Untuk Kunci API:

1. Masukkan nama kunci API
2. Masukkan nama header yang akan berisi kunci API dalam permintaan
3. Masukkan nilai kunci API Anda
4. Klik Berikutnya

Untuk AWS SigV4:

AWS Otentikasi SigV4 memungkinkan AWS DevOps Agen untuk terhubung ke server MCP yang menggunakan AWS Signature Versi 4 untuk penandatanganan permintaan. Ini berguna untuk server MCP yang dihosting di belakang Amazon API Gateway atau AWS layanan lain yang mendukung otentikasi SigV4.

Catatan: Koneksi pribadi tidak didukung untuk server MCP yang menggunakan otentikasi SigV4. Titik akhir server MCP Anda harus dapat diakses publik. Untuk server MCP di jaringan pribadi menggunakan metode otentikasi lainnya, lihat [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#)

1. Konfigurasi peran IAM - Pilih salah satu opsi berikut:
 - Gunakan peran yang ada — Pilih peran IAM yang ada dari dropdown. Peran tersebut harus memiliki kebijakan kepercayaan yang memungkinkan prinsipal layanan AWS DevOps Agen untuk mengasumsikan peran tersebut (lihat [Membuat peran IAM untuk otentikasi SigV4](#)).
 - Buat peran baru secara manual — Ikuti step-by-step petunjuk yang ditampilkan di konsol untuk membuat peran IAM baru dengan kebijakan kepercayaan yang benar.
2. AWS Region — Masukkan AWS Region untuk penandatanganan SigV4 (misalnya, us-east-1). Untuk menggunakan penandatanganan multi-wilayah SigV4a, masukkan. *
3. Nama Layanan — Masukkan nama AWS layanan untuk penandatanganan SigV4 (misalnya, execute-api untuk API Gateway).

4. Kustom Header (opsional) - Tambahkan hingga 10 pasangan header kunci-nilai kustom untuk disertakan dengan setiap permintaan ditandatangani.
5. Klik Berikutnya

Langkah 4: Tinjau dan kirimkan

1. Tinjau semua detail konfigurasi server MCP
2. Klik Submit untuk menyelesaikan pendaftaran
3. AWS DevOps Agen akan memvalidasi koneksi ke server MCP Anda
4. Setelah validasi berhasil, server MCP Anda akan terdaftar di tingkat akun

Mengkonfigurasi alat MCP di Ruang Agen

Setelah mendaftarkan server MCP di tingkat akun, Anda dapat mengonfigurasi alat mana dari server tersebut yang tersedia untuk Ruang Agen tertentu:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen Anda
2. Buka tab Kemampuan
3. Di bagian MCP Server, klik Tambah
4. Pilih server MCP terdaftar yang ingin Anda sambungkan ke Ruang Agen ini
5. Konfigurasi alat mana dari server MCP ini yang harus tersedia untuk Ruang Agen:
 - Izinkan semua alat - Membuat semua alat dari server MCP tersedia
 - Pilih alat khusus - Memungkinkan Anda memilih alat mana yang akan diizinkan
6. Klik Tambah untuk menghubungkan server MCP ke Ruang Agen Anda

AWS DevOps Agen sekarang akan dapat menggunakan alat yang diizinkan dari server MCP Anda selama penyelidikan di Ruang Agen ini.

Mengelola koneksi server MCP

Memperbarui kredensi otentikasi — Jika kredensi otentikasi Anda perlu diperbarui, Anda harus mendaftarkan ulang server MCP Anda. Arahkan ke halaman Penyedia Kemampuan di konsol AWS DevOps Agen, cari server MCP Anda, hapus semua asosiasi aktif, dan klik Deregister. Selanjutnya, daftarkan server MCP Anda dengan kredensi otentikasi baru dan buat ulang asosiasi yang diperlukan dengan Ruang Agen Anda.

Melihat server MCP yang terhubung — Untuk melihat semua server MCP yang terhubung ke Ruang Agen Anda, pilih Ruang Agen Anda, buka tab Kemampuan, dan periksa bagian Server MCP. Anda juga dapat memperbarui alat yang dipilih di sini.

Menghapus koneksi server MCP - Untuk memutuskan sambungan server MCP dari Ruang Agen, pilih server di bagian MCP Server dan klik Hapus. Untuk menghapus pendaftaran server MCP sepenuhnya, hapus dari semua Ruang Agen terlebih dahulu, lalu hapus pendaftaran tingkat akun.

Membuat peran IAM untuk otentikasi SiGv4

Saat menggunakan otentikasi AWS SigV4, AWS DevOps Agen mengasumsikan peran IAM di akun Anda untuk menandatangani permintaan ke server MCP Anda. Peran ini harus memiliki kebijakan kepercayaan yang memungkinkan prinsipal layanan AWS DevOps Agen (`aidevops.amazonaws.com`) untuk menganggapnya, dengan perlindungan wakil yang membingungkan.

Kebijakan kepercayaan

Buat peran IAM dengan kebijakan kepercayaan berikut. Ganti `REGION` dengan AWS Wilayah Anda (misalnya, `us-east-1`) dan `ACCOUNT_ID` dengan ID AWS akun Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "ACCOUNT_ID"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:REGION:ACCOUNT_ID:service/*"
        }
      }
    }
  ]
}
```

Kebijakan kepercayaan mencakup kondisi berikut untuk mencegah [masalah wakil yang membingungkan](#):

- `aws:SourceAccount`— Membatasi asumsi peran untuk permintaan yang berasal dari akun Anda. AWS
- `aws:SourceArn`— Membatasi asumsi peran untuk permintaan yang berasal dari sumber daya layanan AWS DevOps Agen di akun Anda.

Kebijakan izin

Lampirkan kebijakan izin ke peran yang memberikan izin minimum yang diperlukan untuk memanggil server MCP Anda. Misalnya, jika server MCP Anda di-host di belakang Amazon API Gateway, peran tersebut harus memiliki `execute-api:Invoke` izin pada sumber daya API Gateway.

Penandatanganan multi-wilayah (Sigv4a)

Jika server MCP Anda digunakan di beberapa AWS Wilayah, Anda dapat menggunakan [Sigv4a \(Sigv4a Versi 4a\) untuk penandatanganan](#) multi-wilayah. Untuk mengaktifkan ini, masukkan * sebagai AWS Wilayah saat mengonfigurasi otorisasi SigV4. Sigv4a menggunakan penandatanganan asimetris, yang memungkinkan satu permintaan yang ditandatangani valid di beberapa Wilayah.

Topik terkait

- Keamanan di AWS DevOps Agen
- Menyiapkan Ruang Agen
- Perlindungan Injeksi yang Cepat

Menghubungkan beberapa AWS Akun

AWS Akun sekunder memungkinkan AWS DevOps Agen untuk menyelidiki sumber daya di beberapa AWS akun di organisasi Anda. Ketika aplikasi Anda menjangkau beberapa akun, menambahkan akun sekunder memastikan agen memiliki visibilitas ke semua sumber daya yang relevan selama investigasi insiden. Akses yang lebih besar ke akun dan sumber daya yang menyusun aplikasi memastikan akurasi investigasi yang lebih besar.

Prasyarat

Sebelum menambahkan AWS akun sekunder, pastikan Anda memiliki:

- Akses ke konsol AWS DevOps Agen di akun utama
- Akses administratif ke AWS akun sekunder
- Izin IAM untuk membuat peran di akun sekunder

Menambahkan AWS akun sekunder

Selain langkah-langkah di bawah ini, Anda dapat menggunakan [the section called “AWS DevOps Panduan orientasi Agen CLI”](#) untuk menambahkan akun sekunder secara terprogram.

Langkah 1: Mulai konfigurasi akun sekunder

1. Masuk ke Konsol AWS Manajemen dan arahkan ke konsol AWS DevOps Agen
2. Pilih Ruang Agen Anda
3. Buka tab Kemampuan
4. Di bagian Cloud, cari subbagian Sumber sekunder
5. Klik Tambah

Langkah 2: Tentukan nama peran

1. Di bidang Nama peran Anda, masukkan nama untuk peran yang akan Anda buat di akun sekunder
2. Perhatikan nama ini—Anda akan menggunakannya lagi saat membuat peran di akun sekunder
3. Salin kebijakan kepercayaan yang disediakan di konsol dan simpan di ruang awal

Langkah 3: Buat peran di akun sekunder

1. Buka tab browser baru dan masuk ke konsol IAM di akun sekunder AWS
2. Arahkan ke IAM > Peran > Buat peran
3. Pilih Kebijakan kepercayaan khusus
4. Tempelkan kebijakan kepercayaan yang Anda salin dari Langkah 2
5. Klik Berikutnya

Langkah 4: Lampirkan kebijakan AWS terkelola

1. Di bagian Kebijakan izin, cari AIDevOpsAgentAccessPolicy

2. Pilih kotak centang di samping kebijakan AIDevOpsAgentAccessPolicyterkelola
3. Klik Berikutnya

Langkah 5: Beri nama dan buat peran

1. Di bidang Nama peran, masukkan nama peran yang sama yang Anda berikan di Langkah 2
2. (Opsional) Tambahkan deskripsi untuk membantu mengidentifikasi tujuan peran
3. Tinjau kebijakan kepercayaan dan izin terlampir
4. Klik Buat peran

Langkah 6: Lampirkan kebijakan inline

1. Di konsol IAM, cari dan pilih peran yang baru saja Anda buat
2. Buka tab Izin
3. Klik Tambahkan izin > Buat kebijakan sebaris
4. Beralih ke tab JSON
5. Rekatkan kebijakan yang Anda simpan di Langkah 2
6. Tempelkan kebijakan ke editor JSON di konsol IAM
7. Klik Berikutnya
8. Berikan nama untuk kebijakan inline (misalnya, "DevOpsAgentInlinePolicy")
9. Klik Buat kebijakan

Langkah 7: Selesaikan konfigurasi

1. Kembali ke konsol AWS DevOps Agen di akun utama
2. Klik Berikutnya untuk menyelesaikan konfigurasi akun sekunder
3. Verifikasi status koneksi ditampilkan sebagai Aktif

Memahami kebijakan yang diperlukan

AWS DevOps Agen membutuhkan tiga komponen kebijakan untuk mengakses sumber daya di akun sekunder:

- Kebijakan kepercayaan — Memungkinkan AWS DevOps Agen di akun utama untuk mengambil peran dalam akun sekunder. Ini membangun hubungan kepercayaan antar akun.
- `AIDevOpsAgentAccessPolicy` (kebijakan AWS terkelola) — Menyediakan izin baca-saja inti yang dibutuhkan AWS DevOps Agen untuk menyelidiki sumber daya di akun sekunder. Kebijakan ini dikelola oleh AWS dan diperbarui saat kemampuan baru ditambahkan.
- Kebijakan sebaris - Memberikan izin tambahan khusus untuk konfigurasi Ruang Agen Anda. Kebijakan ini dibuat berdasarkan pengaturan Ruang Agen Anda dan dapat mencakup izin untuk integrasi atau fitur tertentu.

Di akun utama, Peran IAM AWS DevOps Agen harus dapat mengambil peran yang dibuat di akun sekunder.

Mengelola akun sekunder

- Melihat akun yang terhubung — Di tab Kemampuan, subbagian Sumber sekunder mencantumkan semua akun sekunder yang terhubung dengan status koneksinya.
- Memperbarui peran IAM — Jika Anda perlu mengubah izin, perbarui kebijakan inline yang dilampirkan pada peran di akun sekunder. Perubahan akan diterapkan segera.
- Menghapus akun sekunder — Untuk memutuskan akun sekunder, pilih akun tersebut di daftar Sumber sekunder dan klik Hapus. Ini tidak menghapus peran IAM di akun sekunder.

Menghubungkan sumber telemetri

AWS DevOps Agen menyediakan tiga cara untuk terhubung ke sumber telemetri Anda.

Integrasi 2 arah bawaan

Saat ini, AWS DevOps Agen mendukung pengguna Dynatrace dengan integrasi 2 arah bawaan yang memungkinkan hal berikut:

- Pemetaan sumber daya topologi - AWS DevOps Agen akan menambah Topologi Ruang DevOps Agen Anda dengan entitas dan hubungan yang tersedia melalui server MCP Dynatrace yang dihosting Agen. AWS DevOps
- Pemicu Investigasi Otomatis - Alur Kerja Dynatrace dapat dikonfigurasi untuk memicu Investigasi resolusi insiden dari Masalah Dynatrace.

- Introspeksi telemetri - AWS DevOps Agen dapat mengintrospeksi telemetri Dynatrace saat menyelidiki masalah melalui server MCP Dynatrace yang dihosting Agen. AWS DevOps
- Pembaruan status - AWS DevOps Agen akan mempublikasikan temuan investigasi utama, analisis akar penyebab, dan rencana mitigasi yang dihasilkan ke antarmuka pengguna Dynatrace.

Untuk mempelajari tentang integrasi 2 arah, lihat

- [the section called “Menghubungkan Dynatrace”](#)

Integrasi 1 arah bawaan

Saat ini, AWS DevOps Agen mendukung AWS CloudWatch, Amazon S3, Datadog, Grafana, New Relic, dan Splunk pengguna dengan built-in, integrasi 1 arah.

Praktik terbaik keamanan: Saat mengonfigurasi kredensial untuk integrasi 1 arah bawaan, kami merekomendasikan pelingkupan kunci dan token API ke akses hanya-baca. AWS DevOps Agen menggunakan kredensial ini hanya untuk introspeksi telemetri dan tidak memerlukan akses tulis ke penyedia telemetri Anda.

Integrasi 1 arah AWS CloudWatch bawaan tidak memerlukan pengaturan tambahan dan memungkinkan yang berikut:

- Pemetaan sumber daya topologi - AWS DevOps Agen akan menambah Topologi Ruang DevOps Agen Anda dengan entitas dan hubungan yang tersedia melalui akun cloud primer dan sekunder yang dikonfigurasi. AWS
- Introspeksi telemetri - AWS DevOps Agen dapat mengintrospeksi AWS CloudWatch telemetri saat menyelidiki masalah melalui peran IAM yang disediakan selama konfigurasi akun cloud primer dan sekunder. AWS

Integrasi 1 arah bawaan Amazon S3 memungkinkan hal berikut:

- Introspeksi telemetri - AWS DevOps Agen dapat membaca objek dari ember Amazon S3 saat menyelidiki masalah. Ini berguna untuk mengakses log, file konfigurasi, dan artefak lain yang disimpan di S3.

Untuk menggunakan integrasi Amazon S3, tambahkan `s3:GetObject` dan `s3:ListBucket` izin ke peran IAM DevOps Agen. Mengikuti prinsip hak istimewa terkecil, lingkup izin ini hanya ke bucket

S3 tertentu yang perlu diakses agen. Untuk informasi selengkapnya tentang mengonfigurasi izin IAM, lihat [the section called “DevOps Izin Agen IAM”](#)

Datadog, Grafana, New Relic, dan Splunk built-in, integrasi 1 arah memerlukan pengaturan dan mengaktifkan yang berikut:

- Pemicu Investigasi Otomatis - Peristiwa Datadog, Grafana, Relik Baru, dan Splunk dapat dikonfigurasi untuk memicu Investigasi resolusi insiden Agen melalui webhook AWS DevOps Agen. AWS DevOps
- Introspeksi telemetry - AWS DevOps Agen dapat mengintrospeksi telemetry Datadog, Grafana, Relik Baru, dan Splunk saat menyelidiki masalah melalui server MCP jarak jauh masing-masing penyedia.

Untuk mempelajari tentang integrasi 1 arah, lihat berikut ini:

- [the section called “Menghubungkan DataDog”](#)
- [the section called “Menghubungkan Grafana”](#)
- [the section called “Menghubungkan Relik Baru”](#)
- [the section called “Menghubungkan Splunk”](#)

Bring-your-own sumber telemetry

Untuk sumber telemetry lainnya, termasuk metrik Prometheus, Anda dapat memanfaatkan dukungan AWS DevOps Agen untuk integrasi server webhook dan MCP.

Untuk mempelajari tentang bring-your-own integrasi, lihat berikut ini

- [the section called “Memanggil DevOps Agen melalui Webhook”](#)
- [the section called “Menghubungkan Server MCP”](#)

Menghubungkan Dynatrace

Built-in, Integrasi 2 arah

Saat ini, AWS DevOps Agen mendukung pengguna Dynatrace dengan integrasi 2 arah bawaan yang memungkinkan hal berikut:

- Pemetaan sumber daya topologi - AWS DevOps Agen akan menambah Topologi Ruang DevOps Agen Anda dengan entitas dan hubungan yang tersedia darinya dari lingkungan Dynatrace Anda.
- Pemicu Investigasi Otomatis - Alur Kerja Dynatrace dapat dikonfigurasi untuk memicu Investigasi resolusi insiden dari Masalah Dynatrace.
- Introspeksi telemetry - AWS DevOps Agen dapat mengintrospeksi telemetry Dynatrace saat menyelidiki masalah melalui server MCP Dynatrace. AWS DevOps Agent-hosted
- Pembaruan status - AWS DevOps Agen akan mempublikasikan temuan investigasi utama, analisis akar penyebab, dan rencana mitigasi yang dihasilkan ke antarmuka pengguna Dynatrace.

Prasyarat

Integrasi AWS DevOps Agen dengan Dynatrace membutuhkan SaaS Dynatrace. Integrasi bergantung pada fitur platform Dynatrace (Alur Kerja, AppEngine aplikasi termasuk aplikasi Agen SRE, dan klien OAuth) yang hanya tersedia di lingkungan Dynatrace SaaS.

Dynatrace Managed (lokal) tidak didukung, dan Dynatrace tidak memiliki rencana untuk membawa fitur platform ini ke Managed. Jika Anda menjalankan Dynatrace Managed, Anda harus meningkatkan ke Dynatrace SaaS sebelum menghubungkannya ke Agen. AWS DevOps Lihat [Memutakhirkan dari Dynatrace Dikelola ke SaaS](#).

Orientasi

Proses Orientasi

Orientasi sistem observabilitas Dynatrace Anda melibatkan tiga tahap:

1. Connect - Buat koneksi ke Dynatrace dengan mengonfigurasi kredensi akses akun, dengan semua lingkungan yang mungkin Anda perlukan
2. Aktifkan - Aktifkan Dynatrace di ruang Agen tertentu dengan lingkungan Dynatrace tertentu
3. Konfigurasi lingkungan Dynatrace Anda - gunakan aplikasi Dynatrace SRE Agents untuk menyelesaikan koneksi dalam 2 klik

Langkah 1: Connect

Membangun koneksi ke lingkungan Dynatrace Anda

Konfigurasi

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Temukan Dynatrace di bagian Penyedia yang tersedia di bawah Telemetri dan klik Daftar
3. Buat klien OAuth di Dynatrace, dengan izin terperinci.
 - a. Lihat dokumentasi [Dynatrace](#)
 - b. Saat siap tekan berikutnya
 - c. Anda dapat menghubungkan beberapa lingkungan Dynatrace dan ruang lingkup yang lebih baru ke lingkungan tertentu untuk setiap Ruang DevOps Agen yang mungkin Anda miliki.
4. Masukkan detail Dynatrace Anda dari pengaturan klien OAuth:
 - Nama Klien
 - ID Klien
 - Rahasia Klien
 - Akun URN
5. Klik Berikutnya
6. Tinjau dan tambahkan

Langkah 2: Aktifkan

Aktifkan Dynatrace di ruang Agen tertentu dan konfigurasi pelingkupan yang sesuai

Konfigurasi

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan
2. Pilih tab Kemampuan
3. Temukan bagian Telemetri, Tekan Tambah
4. Anda akan melihat Dynatrace dengan status 'Terdaftar'. Klik tambahkan untuk menambahkan ini ke ruang agen Anda
5. Dynatrace Environment ID - Berikan ID lingkungan Dynatrace yang ingin Anda kaitkan dengan ruang agen ini. DevOps
6. Masukkan satu atau beberapa ID Entitas Dynatrace - DevOps agen bantuan ini menemukan sumber daya Anda yang paling penting, contohnya mungkin layanan atau aplikasi. Jika Anda tidak yakin, Anda dapat menekan hapus.

7. Tinjau dan tekan Simpan
8. Salin URL Webhook dan Rahasia Webhook. Anda akan menggunakan ini di aplikasi Dynatrace SRE Agents untuk menyelesaikan koneksi. Lihat [bagian Memulai](#) untuk detailnya.

Langkah 3: Konfigurasi lingkungan Dynatrace Anda

Untuk menyelesaikan penyiapan Dynatrace Anda, gunakan aplikasi Dynatrace SRE Agents untuk mengonfigurasi sisi Dynatrace integrasi dalam 2 klik — tidak diperlukan pengaturan alur kerja manual. Untuk detailnya, lihat [bagian Memulai](#).

Skema Acara yang Didukung

AWS DevOps Agen mendukung dua jenis peristiwa dari Dynatrace menggunakan webhook. Skema acara yang didukung didokumentasikan di bawah ini:

Peristiwa Insiden

Peristiwa insiden digunakan untuk memicu penyelidikan. Skema acara adalah:

```
{
  "event.id": string;
  "event.status": "ACTIVE" | "CLOSED";
  "event.status_transition": string;
  "event.description": string;
  "event.name": string;
  "event.category": "AVAILABILITY" | "ERROR" | "SLOWDOWN" | "RESOURCE_CONTENTION" |
"CUSTOM_ALERT" | "MONITORING_UNAVAILABLE" | "INFO";
  "event.start"?: string;
  "affected_entity_ids"?: string[];
}
```

Acara Mitigasi

Peristiwa mitigasi digunakan untuk memicu pembuatan laporan mitigasi untuk penyelidikan pada langkah selanjutnya. Skema acara adalah:

```
{
  "task_id": string;
  "task_version": number;
  "event.type": "mitigation_request";
}
```

```
}
```

Penghapusan

Sumber telemetri terhubung pada dua tingkat di tingkat ruang agen dan di tingkat akun. Untuk menghapusnya sepenuhnya, Anda harus terlebih dahulu menghapus dari semua ruang agen di mana ia digunakan dan kemudian dapat tidak terdaftar.

Langkah 1: Hapus dari ruang agen

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetry
4. Pilih Dynatrace
5. Tekan hapus

Langkah 2: Deregister dari akun

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Gulir ke bagian Saat ini terdaftar.
3. Periksa jumlah ruang agen adalah nol (jika tidak ulangi Langkah 1 di atas di ruang agen Anda yang lain)
4. Tekan Deregister di sebelah Dynatrace

Menghubungkan DataDog

Built-in, Integrasi 1 arah

Saat ini, AWS DevOps Agen mendukung pengguna Datadog dengan integrasi 1 arah bawaan, memungkinkan yang berikut:

- Pemicu Investigasi Otomatis - Peristiwa datadog dapat dikonfigurasi untuk memicu Investigasi resolusi insiden AWS DevOps Agen melalui webhook Agen. AWS DevOps
- Introspeksi telemetri - AWS DevOps Agen dapat mengintrospeksi telemetri Datadog saat menyelidiki masalah melalui server MCP jarak jauh masing-masing penyedia.

Orientasi

Langkah 1: Connect

Buat koneksi ke endpoint MCP jarak jauh Datadog Anda dengan kredensi akses akun

Konfigurasi

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Temukan Datadog di bagian Penyedia yang tersedia di bawah Telemetri dan klik Daftar
3. Masukkan detail server MCP Datadog Anda:
 - Nama Server - Pengidentifikasi unik (misalnya, my-datadog-server)
 - URL Endpoint - Titik akhir server MCP Datadog Anda. URL endpoint bervariasi tergantung pada situs Datadog Anda. Lihat tabel endpoint situs Datadog di bawah ini.
 - Deskripsi - Deskripsi server opsional
4. Klik Berikutnya
5. Tinjau dan kirimkan

Titik akhir situs Datadog

URL endpoint MCP bervariasi tergantung pada situs Datadog Anda. Untuk mengidentifikasi situs Anda, periksa URL di browser Anda saat masuk ke Datadog, atau lihat [Akses situs Datadog](#).

Situs Datadog	Domain Situs	URL Titik Akhir MCP
US1 (default)	datadoghq.com	https://mcp.datadoghq.com/api/unstable/mcp-server/mcp
US3	us3.datadoghq.com	https://mcp.us3.datadoghq.com/api/unstable/mcp-server/mcp
US5	us5.datadoghq.com	https://mcp.us5.datadoghq.com/api/un

Situs Datadog	Domain Situs	URL Titik Akhir MCP
		stable/mcp-server/mcp
EU1	datadoghq.eu	https://mcp.datadoghq.eu/api/unstable/mcp-server/mcp
AP1	ap1.datadoghq.com	https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp
AP2	ap2.datadoghq.com	https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp

Otorisasi

Otorisasi OAuth lengkap dengan:

- Otorisasi sebagai pengguna Anda di halaman Datadog OAuth
- Jika tidak masuk, klik Izinkan, masuk, lalu otorisasi

Setelah dikonfigurasi, Datadog menjadi tersedia di semua ruang Agen.

Langkah 2: Aktifkan

Aktifkan DataDog di ruang Agen tertentu dan konfigurasi pelingkupan yang sesuai

Konfigurasi

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan (jika Anda belum membuat ruang agen lihat [the section called “Membuat Ruang Agen”](#))
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetry

4. Tekan Tambah
5. Pilih Datadog
6. Selanjutnya
7. Tinjau dan tekan Simpan
8. Salin URL Webhook dan Kunci API

Langkah 3: Konfigurasi webhooks

Menggunakan URL Webhook dan API Key Anda dapat mengonfigurasi Datadog untuk mengirim peristiwa untuk memicu penyelidikan, misalnya dari alarm.

Webhook Datadog menggunakan otentikasi token pembawa. Untuk format permintaan webhook lengkap, skema payload, dan kode contoh, lihat [the section called “Memanggil DevOps Agen melalui Webhook”](#) Gunakan contoh Versi 2 (Bearer token authentication), atur Authorization: Bearer <Token> header dengan API Key dari Langkah 2.

Kirim webhook dengan Datadog <https://docs.datadoghq.com/integrations/webhooks/> (catatan pilih tidak ada otorisasi dan sebagai gantinya gunakan opsi header kustom).

Pelajari lebih lanjut: [Datadog Remote MCP](#) Server

Penghapusan

Sumber telemetry terhubung pada dua tingkat di tingkat ruang agen dan di tingkat akun. Untuk menghapusnya sepenuhnya, Anda harus terlebih dahulu menghapus dari semua ruang agen di mana ia digunakan dan kemudian dapat tidak terdaftar.

Langkah 1: Hapus dari ruang agen

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetry
4. Pilih Datadog
5. Tekan hapus

Langkah 2: Deregister dari akun

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)

2. Gulir ke bagian Saat ini terdaftar.
3. Periksa jumlah ruang agen adalah nol (jika tidak ulangi Langkah 1 di atas di ruang agen Anda yang lain)
4. Tekan Deregister di sebelah Datadog

Menghubungkan Grafana

Integrasi Grafana memungkinkan AWS DevOps Agen untuk menanyakan metrik, dasbor, dan data peringatan dari instans Grafana Anda selama penyelidikan insiden. Integrasi ini mengikuti proses dua langkah: pendaftaran Grafana tingkat akun, diikuti dengan menghubungkannya ke Ruang Agen individu.

Untuk meningkatkan keamanan, integrasi Grafana hanya mengaktifkan alat hanya-baca. Alat tulis dinonaktifkan dan tidak dapat diaktifkan. Ini berarti agen dapat menanyakan dan membaca data dari instance Grafana Anda tetapi tidak dapat membuat, memodifikasi, atau menghapus sumber daya Grafana apa pun seperti dasbor, peringatan, atau anotasi. Untuk informasi selengkapnya, lihat [Keamanan di AWS DevOps Agen](#).

Persyaratan Grafana

Sebelum menghubungkan Grafana, pastikan Anda memiliki:

- Grafana versi 9.0 atau yang lebih baru. Beberapa fitur, terutama operasi terkait sumber data, mungkin tidak berfungsi dengan benar dengan versi sebelumnya karena titik akhir API yang hilang.
- Instans Grafana dapat diakses melalui HTTPS. Endpoint jaringan publik dan pribadi didukung. Dengan konektivitas jaringan pribadi, instans Grafana Anda dapat di-host di dalam VPC tanpa akses internet publik. Lihat perinciannya di [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#).
- Akun layanan Grafana dengan token akses yang memiliki izin baca yang sesuai

Mendaftarkan Grafana (tingkat akun)

Grafana terdaftar di tingkat AWS akun dan dibagikan di antara semua Ruang Agen di akun itu.

Langkah 1: Konfigurasi Grafana

1. Masuk ke Konsol AWS Manajemen

2. Arahkan ke konsol AWS DevOps Agen
3. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
4. Temukan Grafana di bagian Penyedia yang tersedia di bawah Telemetri dan klik Daftar
5. Pada halaman Konfigurasi Grafana, masukkan informasi berikut:
 - Nama Layanan (wajib) — Masukkan nama deskriptif untuk server Grafana Anda menggunakan karakter alfanumerik, tanda hubung, dan garis bawah saja. Misalnya, `my-grafana-server`.
 - URL Grafana (wajib) — Masukkan URL HTTPS lengkap instance Grafana Anda. Misalnya, `https://myinstance.grafana.net`.
 - Token Akses Akun Layanan (wajib) — Masukkan token akses akun layanan Grafana. Token biasanya dimulai dengan `lsa_`. Untuk membuat token akun layanan, navigasikan ke instans Grafana Anda, buka Administrasi > Akun layanan, buat akun layanan dengan peran Viewer, dan buat token.
 - Deskripsi (opsional) — Tambahkan deskripsi untuk membantu mengidentifikasi tujuan server. Misalnya, `Production Grafana server for monitoring`.
6. (Opsional) Tambahkan AWS tag ke pendaftaran untuk tujuan organisasi.
7. Klik Berikutnya

Langkah 2: Tinjau dan kirimkan pendaftaran Grafana

1. Tinjau semua detail konfigurasi Grafana
2. Klik Submit untuk menyelesaikan pendaftaran
3. Setelah pendaftaran berhasil, Grafana muncul di bagian Saat ini terdaftar di halaman Penyedia Kemampuan

Menambahkan Grafana ke Ruang Agen

Setelah mendaftarkan Grafana di tingkat akun, Anda dapat menghubungkannya ke Ruang Agen individual:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Kemampuan
3. Di bagian Telemetri, klik Tambah
4. Pilih Grafana dari daftar penyedia yang tersedia
5. Klik Simpan

Mengkonfigurasi webhook peringatan Grafana

Anda dapat mengonfigurasi Grafana untuk secara otomatis memicu investigasi AWS DevOps Agen saat peringatan menyala dengan mengirimkan webhook melalui titik kontak Grafana. Untuk detail tentang metode otentikasi webhook dan manajemen kredensi, lihat. [the section called “Memanggil DevOps Agen melalui Webhook”](#)

Langkah 1: Buat template notifikasi khusus

Dalam instance Grafana Anda, navigasikan ke Alerting > Contact points > Notification templates dan buat template baru dengan konten berikut:

```
{{ define "devops-agent-payload" }}
{
  "eventType": "incident",
  "incidentId": "{{ (index .Alerts 0).Labels.alertname }}-{{ (index .Alerts
0).Fingerprint }}",
  "action": "{{ if eq .Status "resolved" }}resolved{{ else }}created{{ end }}",
  "priority": "{{ if eq .Status "resolved" }}MEDIUM{{ else }}HIGH{{ end }}",
  "title": "{{ (index .Alerts 0).Labels.alertname }}",
  "description": "{{ (index .Alerts 0).Annotations.summary }}",
  "service": "{{ if (index .Alerts 0).Labels.job }}{{ (index .Alerts 0).Labels.job }}
{{ else }}grafana{{ end }}",
  "timestamp": "{{ (index .Alerts 0).StartsAt }}",
  "data": {
    "metadata": {
      {{ range $k, $v := (index .Alerts 0).Labels }}
      "{{ $k }}": "{{ $v }}",
      {{ end }}
      "_source": "grafana"
    }
  }
}
{{ end }}
```

Template ini memformat peringatan Grafana ke dalam struktur payload webhook yang diharapkan oleh Agen. AWS DevOps Ini memetakan label peringatan, anotasi, dan status ke dalam bidang yang sesuai, dan mencakup semua label peringatan sebagai metadata.

Catatan: Template ini hanya memproses peringatan pertama dalam grup. Grafana mengelompokkan beberapa peringatan penembakan menjadi satu notifikasi secara default. Untuk memastikan setiap peringatan dikirim satu per satu, konfigurasi

kebijakan notifikasi Anda untuk dikelompokkan menurut `alertrname`. Selain itu, template ini tidak luput dari karakter JSON khusus dalam nilai label atau anotasi. Pastikan label peringatan dan summary anotasi tidak mengandung karakter seperti tanda kutip ganda atau baris baru, yang akan menghasilkan JSON yang tidak valid.

Langkah 2: Buat titik kontak webhook

1. Di Grafana, navigasikan ke Peringatan > Titik kontak dan klik Tambahkan titik kontak
2. Pilih Webhook sebagai tipe integrasi
3. Atur URL ke titik akhir webhook AWS DevOps Agen
4. Di bawah pengaturan Webhook Opsional, konfigurasi header otentikasi berdasarkan jenis webhook Anda. Lihat [metode otentikasi Webhook](#) untuk detailnya.
5. Atur bidang Custom Payload untuk menggunakan template kustom Anda:

```
{ "template": "devops-agent-payload" }
```
6. Klik Simpan titik kontak

Langkah 3: Tetapkan titik kontak ke kebijakan pemberitahuan

1. Arahkan ke Peringatan > Kebijakan pemberitahuan
2. Mengedit kebijakan yang sudah ada atau membuat yang baru
3. Atur titik kontak ke titik kontak webhook yang Anda buat
4. Klik Simpan kebijakan

Saat peringatan yang cocok diaktifkan, Grafana akan mengirimkan muatan yang diformat ke AWS DevOps Agen, yang akan memulai penyelidikan secara otomatis.

Batasan

- ClickHouse alat sumber ClickHouse data — alat sumber data saat ini tidak didukung.
- Pencegahan insiden proaktif — saat ini [the section called “Pencegahan insiden proaktif”](#) tidak menggunakan alat Grafana. Support direncanakan untuk rilis di masa depan.

Pertimbangan Grafana yang Dikelola Amazon

Jika Anda menggunakan [Grafana Terkelola Amazon](#) (AMG), perhatikan batasan berikut:

- Titik kontak Webhook tidak didukung - AMG saat ini tidak mendukung titik kontak webhook dalam konfigurasi peringatannya. Anda tidak dapat menggunakan AMG untuk mengirim webhook peringatan langsung ke Agen. AWS DevOps Untuk detailnya, lihat [Memperingatkan titik kontak di Grafana Terkelola Amazon](#).
- Kedaluwarsa token akun layanan — Token akun layanan AMG memiliki masa kedaluwarsa maksimum 30 hari. Anda perlu memutar token dan memperbarui pendaftaran Grafana Anda di AWS DevOps Agen sebelum kedaluwarsa. Lihat [Mengelola koneksi Grafana](#) untuk mengetahui cara memperbarui kredensial. Untuk detail tentang batas token AMG, lihat [Akun layanan di Grafana Terkelola Amazon](#).

Mengelola koneksi Grafana

- Memperbarui kredensi — Jika token akun layanan Anda kedaluwarsa atau perlu diperbarui, deregister Grafana dari halaman Penyedia Kemampuan dan daftar ulang dengan token baru.
- Melihat instans yang terhubung — Di konsol AWS DevOps Agen, pilih Ruang Agen Anda dan buka tab Kemampuan untuk melihat sumber telemetri yang terhubung.
- Menghapus Grafana — Untuk memutuskan Grafana dari Ruang Agen, pilih Grafana di bagian Telemetri dan klik Hapus. Untuk menghapus pendaftaran sepenuhnya, hapus terlebih dahulu dari semua Ruang Agen, lalu deregister dari halaman Penyedia Kemampuan.

Menghubungkan Relik Baru

Built-in, Integrasi 1 arah

Saat ini, AWS DevOps Agen mendukung pengguna New Relic dengan integrasi 1 arah bawaan, memungkinkan yang berikut:

- Pemicu Investigasi Otomatis - Peristiwa Relik Baru dapat dikonfigurasi untuk memicu Investigasi resolusi insiden AWS DevOps Agen melalui AWS DevOps webhook Agen.
- Introspeksi telemetri - AWS DevOps Agen dapat mengintrospeksi telemetri Relik Baru saat menyelidiki masalah melalui server MCP jarak jauh masing-masing penyedia.

Orientasi

Langkah 1: Connect

Buat koneksi ke titik akhir MCP jarak jauh New Relic Anda dengan kredensi akses akun

Silakan gunakan Pengguna Platform Lengkap (bukan Basic/Core) di Peninggalan baru untuk mengaktifkan alat MCP Relik Baru.

Konfigurasi

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Temukan Relik Baru di bagian Penyedia yang tersedia di bawah Telemetri dan klik Daftar
3. Ikuti petunjuk untuk mendapatkan New Relic API Key
4. Masukkan detail Kunci API server MCP Relic Baru Anda:
 - ID Akun: Masukkan ID akun New Relic Anda yang diperoleh di atas
 - Kunci API: Masukkan Kunci API yang diperoleh di atas
 - Pilih wilayah AS atau UE berdasarkan di mana akun New Relic Anda berada.
5. Klik Tambah

Langkah 2: Aktifkan

Aktifkan New Relic di ruang Agen tertentu dan konfigurasi pelingkupan yang sesuai

Konfigurasi

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan (jika Anda belum membuat ruang agen lihat [the section called "Membuat Ruang Agen"](#))
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetri
4. Tekan Tambah
5. Pilih New Relic
6. Selanjutnya
7. Tinjau dan tekan Simpan
8. Salin URL Webhook dan Kunci API

Langkah 3: Konfigurasi webhooks

Menggunakan URL Webhook dan API Key Anda dapat mengonfigurasi New Relic untuk mengirim peristiwa untuk memicu penyelidikan, misalnya dari alarm. Untuk detail selengkapnya tentang menyiapkan webhook, lihat [Mengubah webhook pelacakan](#).

Webhook Relic baru menggunakan otentikasi token pembawa. Untuk format permintaan webhook lengkap, skema payload, dan kode contoh, lihat [the section called “Memanggil DevOps Agen melalui Webhook”](#) Gunakan contoh Versi 2 (Bearer token authentication), atur `Authorization: Bearer <Token>` header dengan API Key dari Langkah 2.

Kirim webhook dengan New Relic. <https://newrelic.com/instant-observability/webhook-notifications> Anda dapat memilih token Pembawa untuk jenis otorisasi, atau memilih tidak ada otorisasi dan menambahkan header sebagai kustom `Authorization: Bearer <Token>` sebagai gantinya.

Pelajari lebih lanjut: <https://docs.newrelic.com/docs/agentic-ai/mcp/overview/>

Penghapusan

Sumber telemetri terhubung pada dua tingkat di tingkat ruang agen dan di tingkat akun. Untuk menghapusnya sepenuhnya, Anda harus terlebih dahulu menghapus dari semua ruang agen di mana ia digunakan dan kemudian dapat tidak terdaftar.

Langkah 1: Hapus dari ruang agen

1. Dari halaman ruang agen, pilih ruang agen dan tekan detail tampilan
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetri
4. Pilih New Relic
5. Tekan hapus

Langkah 2: Deregister dari akun

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Gulir ke bagian Saat ini terdaftar.
3. Periksa jumlah ruang agen adalah nol (jika tidak ulangi Langkah 1 di atas di ruang agen Anda yang lain)

4. Tekan Deregister di sebelah New Relic

Menghubungkan Splunk

Built-in, Integrasi 1 arah

Saat ini, AWS DevOps Agen mendukung pengguna Splunk dengan integrasi 1 arah bawaan, memungkinkan yang berikut:

- Pemicu Investigasi Otomatis - Peristiwa splunk dapat dikonfigurasi untuk memicu Investigasi resolusi insiden AWS DevOps Agen melalui AWS DevOps webhook Agen.
- Introspeksi telemetri - AWS DevOps Agen dapat mengintrospeksi telemetri Splunk saat menyelidiki masalah melalui server MCP jarak jauh masing-masing penyedia.

Prasyarat

Mendapatkan token API Splunk

Anda akan memerlukan URL dan token MCP untuk menghubungkan Splunk.

Langkah-langkah Administrator Splunk

Administrator Splunk Anda perlu melakukan langkah-langkah berikut:

- aktifkan [akses REST API](#)
- [aktifkan otentikasi token](#) pada penerapan.
- buat peran baru 'mcp_user', peran baru tidak perlu memiliki kemampuan apa pun.
- menetapkan peran 'mcp_user' untuk setiap pengguna pada penyebaran yang berwenang untuk menggunakan server MCP.
- buat token untuk pengguna yang berwenang dengan audiens sebagai 'mcp' dan atur kedaluwarsa yang sesuai, jika pengguna tidak memiliki izin untuk membuat token sendiri.

Langkah Pengguna Splunk

Pengguna Splunk perlu melakukan langkah-langkah berikut:

- Dapatkan token yang sesuai dari Administrator Splunk atau buat sendiri, jika mereka memiliki izin. Audiens untuk token harus 'mcp'.

Orientasi

Langkah 1: Connect

Buat koneksi ke titik akhir MCP jarak jauh Splunk Anda dengan kredensi akses akun

Konfigurasi

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Temukan Splunk di bagian Penyedia yang tersedia di bawah Telemetry dan klik Daftar
3. Masukkan detail server MCP Splunk Anda:
 - Nama Server - Pengidentifikasi unik (misalnya, my-splunk-server)
 - URL titik akhir - Titik akhir server MCP Splunk Anda:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/  
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Deskripsi - Deskripsi server opsional
- Nama Token - Nama token pembawa untuk otentikasi: my-splunk-token
- Nilai Token - Nilai token pembawa untuk otentikasi

Langkah 2: Aktifkan

Aktifkan Splunk di ruang Agen tertentu dan konfigurasi pelingkupan yang sesuai

Konfigurasi

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan (jika Anda belum membuat ruang agen lihat [the section called "Membuat Ruang Agen"](#))
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetry
4. Tekan Tambah
5. Pilih Splunk
6. Selanjutnya
7. Tinjau dan tekan Simpan
8. Salin URL Webhook dan Kunci API

Langkah 3: Konfigurasi webhooks

Menggunakan URL Webhook dan API Key Anda dapat mengonfigurasi Splunk untuk mengirim peristiwa untuk memicu penyelidikan, misalnya dari alarm.

Webhook splunk menggunakan otentikasi token pembawa. Untuk format permintaan webhook lengkap, skema payload, dan kode contoh, lihat. [the section called “Memanggil DevOps Agen melalui Webhook”](#) Gunakan contoh Versi 2 (Bearer token authentication), atur Authorization: Bearer <Token> header dengan API Key dari Langkah 2.

Kirim webhook dengan Splunk <https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action>(perhatikan pilih tidak ada otorisasi dan sebagai gantinya gunakan opsi header khusus)

Pelajari lebih lanjut:

- Dokumentasi Server MCP Splunk: <https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform>
- Persyaratan dan batasan akses untuk REST API Splunk Cloud Platform: <https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud>
- Kelola token otentikasi di Splunk Cloud Platform: <https://help.splunk.com/en/splunk-cloud-platform/administer/manage-users-and-security/9.3.2411/authenticate-into-the-splunk-platform-with-tokens/manage-or-delete-authentication-tokens>
- Buat dan kelola peran dengan Splunk Web: <https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles>

Penghapusan

Sumber telemetry terhubung pada dua tingkat di tingkat ruang agen dan di tingkat akun. Untuk menghapusnya sepenuhnya, Anda harus terlebih dahulu menghapusnya dari semua ruang agen tempat ia digunakan dan kemudian dapat tidak terdaftar.

Langkah 1: Hapus dari ruang agen

1. Dari halaman spasi agen, pilih ruang agen dan tekan detail tampilan
2. Pilih tab Kemampuan
3. Gulir ke bawah ke bagian Telemetry
4. Pilih Splunk

5. Tekan hapus

Langkah 2: Deregister dari akun

1. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
2. Gulir ke bagian Saat ini terdaftar.
3. Periksa jumlah ruang agen adalah nol (jika tidak ulangi Langkah 1 di atas di ruang agen Anda yang lain)
4. Tekan Deregister di sebelah Splunk

Menghubungkan ke tiket dan obrolan

AWS DevOps Agen dirancang untuk bertindak sebagai anggota tim Anda dengan berpartisipasi dalam saluran komunikasi tim Anda yang ada. Anda dapat menghubungkan DevOps Agen ke sistem tiket dan mengkhawatirkan, seperti ServiceNow dan, untuk secara otomatis meluncurkan investigasi dari tiket insiden PagerDuty, mempercepat respons insiden dalam alur kerja yang ada untuk mengurangi mean time to recover (MTTR). Anda juga dapat menghubungkan DevOps Agen Anda ke sistem kolaborasi tim seperti Slack untuk menerima ringkasan aktivitas dari DevOps Agen Anda di saluran obrolan.

Untuk mempelajari tentang menghubungkan integrasi tiket dan obrolan, lihat yang berikut ini:

- [the section called “Menghubungkan PagerDuty”](#)
- [the section called “Menghubungkan ServiceNow”](#)
- [the section called “Menghubungkan Slack”](#)

Menghubungkan PagerDuty

PagerDuty integrasi memungkinkan AWS DevOps Agen untuk mengakses dan memperbarui data insiden, jadwal panggilan, dan informasi layanan dari PagerDuty akun Anda selama investigasi insiden dan respons otomatis. Integrasi ini menggunakan OAuth 2.0 untuk otentikasi aman.

Important

AWS DevOps Agen hanya mendukung PagerDuty OAuth 2.0 (OAuthScoped) yang lebih baru. Warisan PagerDuty OAuth dengan uri pengalihan tidak didukung.

PagerDuty persyaratan

Sebelum menghubungkan PagerDuty, pastikan Anda memiliki:

- PagerDuty Akun dengan ID OAuth klien dan rahasia klien Anda
- Subdomain PagerDuty akun Anda (misalnya, jika PagerDuty URL Anda `https://your-company.pagerduty.com`, subdomain adalah `your-company`)

Mendaftar PagerDuty

PagerDuty terdaftar di tingkat AWS akun dan dibagikan di antara semua Ruang Agen di akun itu.

Langkah 1: Konfigurasi akses di PagerDuty

1. Masuk ke Konsol AWS Manajemen
2. Arahkan ke konsol AWS DevOps Agen
3. Buka halaman Penyedia Kemampuan (dapat diakses dari navigasi samping)
4. Temukan PagerDuty di bagian Penyedia yang tersedia di bawah Komunikasi dan klik Daftar
5. Ikuti pengaturan terpandu pada PagerDuty halaman Konfigurasi akses di:

Periksa wilayah layanan dan subdomain Anda:

- Cakupan akun — Pilih PagerDuty wilayah Anda (AS atau UE) dan masukkan PagerDuty subdomain Anda. Misalnya, jika PagerDuty URL Anda `https://your-company.pagerduty.com`, masukkan `your-company`.

Buat aplikasi baru di PagerDuty:

- Di tab browser terpisah, masuk ke PagerDuty dan navigasikan ke Integrasi > Pendaftaran Aplikasi
- Buat aplikasi baru menggunakan OAuth 2.0 Scoped OAuth
- Di bawah Izin, berikan cakupan minimum yang diperlukan berikut: `incidents.read`, dan `incidents.write services.read`
- Aktifkan Integrasi Acara untuk memungkinkan komunikasi dua arah antara Agen dan AWS DevOps PagerDuty

Konfigurasi OAuth kredensial:

- Ruang lingkup izin - Cakupan minimum yang diperlukan adalah:`incidents.read`,
`incidents.write` `services.read`
- Nama klien — Masukkan nama deskriptif untuk klien Anda OAuth
- ID Klien — Masukkan ID OAuth klien dari pendaftaran PagerDuty aplikasi Anda
- Rahasia klien — Masukkan rahasia OAuth klien dari pendaftaran PagerDuty aplikasi Anda

Langkah 2: Tinjau dan kirimkan PagerDuty pendaftaran

1. Tinjau semua detail PagerDuty konfigurasi
2. Klik Submit untuk menyelesaikan pendaftaran
3. Setelah pendaftaran berhasil, PagerDuty muncul di bagian Saat ini terdaftar di halaman Penyedia Kemampuan

PagerDuty Menambah Ruang Agen

Setelah mendaftar PagerDuty di tingkat akun, Anda dapat menghubungkannya ke Ruang Agen individual:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Kemampuan
3. Di bagian Komunikasi, klik Tambah
4. Pilih PagerDuty dari daftar penyedia yang tersedia
5. Klik Simpan

Mengelola PagerDuty koneksi

- Memperbarui kredensi — Jika OAuth kredensial Anda perlu diperbarui, deregister PagerDuty dari halaman Penyedia Kemampuan dan daftar ulang dengan kredensi baru.
- Melihat koneksi — Di konsol AWS DevOps Agen, pilih Ruang Agen Anda dan buka tab Kemampuan untuk melihat integrasi komunikasi yang terhubung.
- Menghapus PagerDuty - Untuk memutuskan sambungan PagerDuty dari Ruang Agen, pilih di bagian Komunikasi dan klik Hapus. Untuk menghapus pendaftaran sepenuhnya, hapus terlebih dahulu dari semua Ruang Agen, lalu deregister dari halaman Penyedia Kemampuan.

Dukungan Webhook

AWS DevOps Agen hanya mendukung PagerDuty webhook V3. Versi webhook sebelumnya tidak didukung.

Untuk informasi selengkapnya tentang langganan webhook PagerDuty V3, lihat [Ikhtisar Webhooks](#) di dokumentasi pengembang. PagerDuty

Menghubungkan ServiceNow

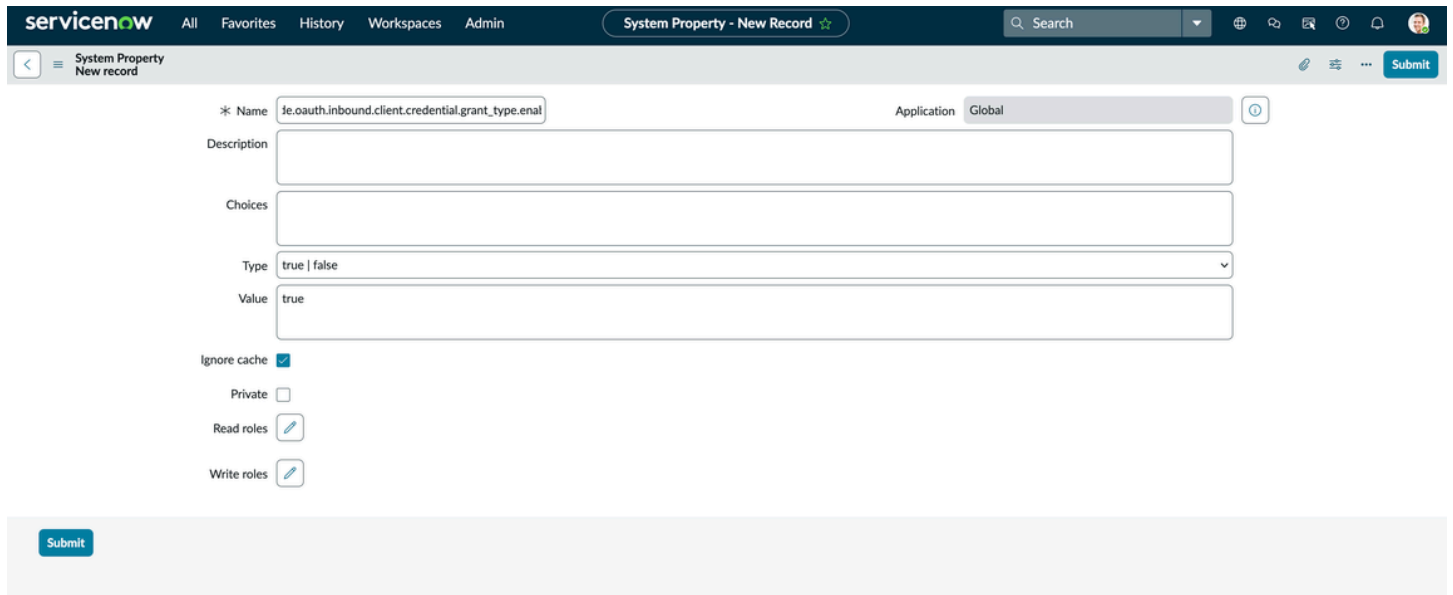
Tutorial ini memandu Anda untuk menghubungkan ServiceNow instance ke AWS DevOps Agen untuk memungkinkannya memulai investigasi respons insiden secara otomatis saat tiket dibuat dan memposting temuan utamanya ke dalam tiket asal. Ini juga berisi contoh cara mengonfigurasi ServiceNow instans Anda untuk mengirim hanya tiket tertentu ke Ruang DevOps Agen dan cara mengatur perutean tiket di beberapa Ruang Agen. DevOps

Pengaturan Awal

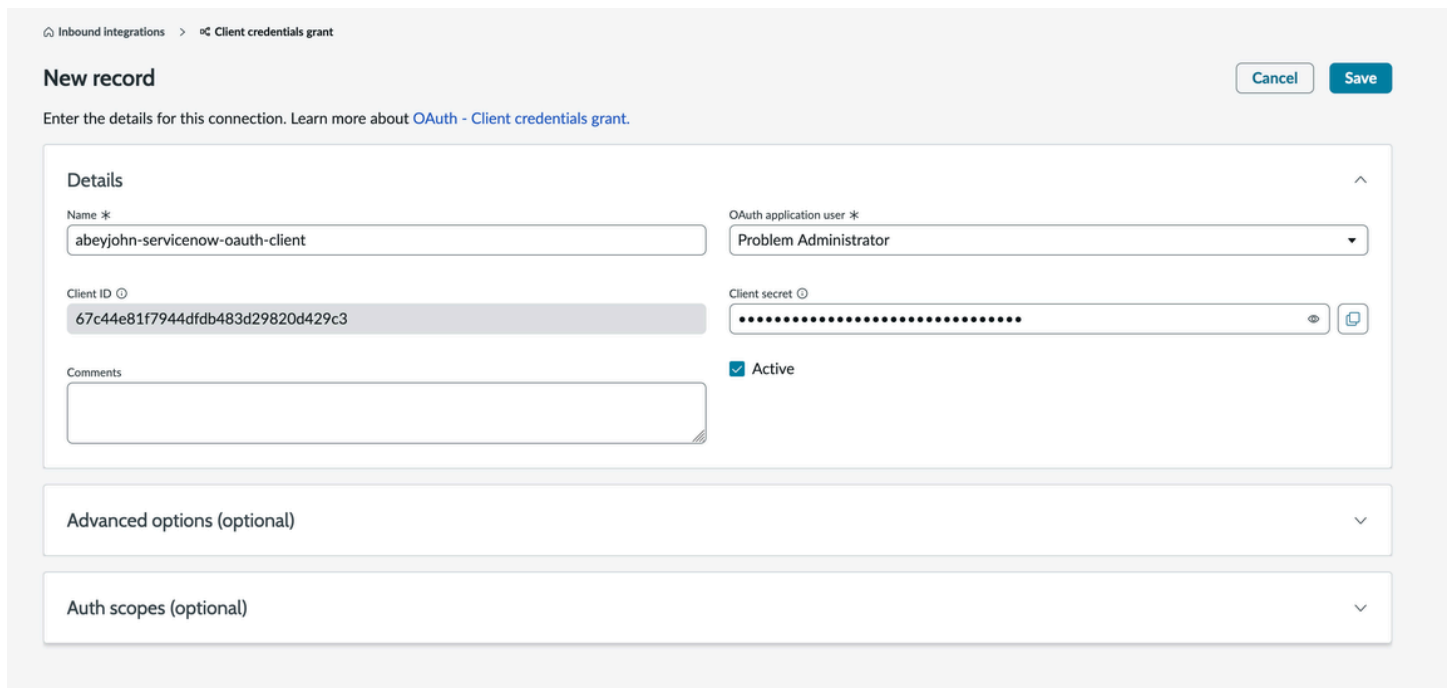
Langkah pertama adalah membuat klien ServiceNow aplikasi OAuth yang AWS DevOps dapat digunakan untuk mengakses instance Anda ServiceNow .

Buat klien aplikasi ServiceNow OAuth

1. Aktifkan properti sistem kredensi klien instans Anda
 - a. Cari `sys_properties.list` di kotak pencarian filter dan kemudian tekan enter (itu tidak akan menampilkan opsi tetapi menekan enter berfungsi)
 - b. Pilih Baru
 - c. Tambahkan nama as `glide.oauth.inbound.client.credential.grant_type.enabled` dan nilai ke true dengan type as true | false



1. Arahkan ke System OAuth > Application Registry dari kotak pencarian filter
2. Pilih “Baru”> “Pengalaman Integrasi Masuk Baru” > “Integrasi Baru” > “OAuth - Hibah Kredensial Klien”
3. Pilih nama dan atur pengguna aplikasi OAuth ke “Administrator Masalah”, klik “Simpan”



Konfigurasi Aturan ServiceNow Bisnis Anda

Setelah Anda membuat konektivitas, Anda harus mengonfigurasi aturan bisnis ServiceNow untuk mengirim tiket ke Ruang DevOps Agen Anda.

1. Arahkan ke **Langganan Aktivitas** → **Administrasi** → **Aturan Bisnis**, dan klik **Baru**.
2. Setel bidang “Tabel” ke “Insiden [insiden]”, centang kotak “Lanjutan”, dan setel aturan untuk dijalankan setelah **Sisipkan**, **Perbarui**, dan **Hapus**.

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met. [More Info](#)

Name: CloudSmith Integration
 Table: Incident [insiden]
 Application: Global
 Active:
 Advanced:

When to run | Actions | Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When: after
 Order: 100

Filter Conditions: [Add Filter Condition](#) [Add OR Clause](#)
 -- choose field -- -- oper -- -- value --

Role conditions: [\[Pencil Icon\]](#)

Insert:
 Update:
 Delete:
 Query:

[Submit](#)

1. Arahkan ke tab “Advanced” dan tambahkan skrip webhook berikut, masukkan rahasia webhook dan URL Anda di mana ditunjukkan, dan klik **Kirim**.

```
(function executeRule(current, previous /*null when async*/ ) {

    var WEBHOOK_CONFIG = {
        webhookSecret: GlideStringUtil.base64Encode('<<< INSERT WEBHOOK SECRET HERE >>>'),
        webhookUrl: '<<< INSERT WEBHOOK URL HERE >>>'
    };

    function generateHMACSignature(payloadString, secret) {
```

```
    try {
        var mac = new GlideCertificateEncryption();
        var signature = mac.generateMac(secret, "HmacSHA256", payloadString);
        return signature;
    } catch (e) {
        gs.error('HMAC generation failed: ' + e);
        return null;
    }
}

function callWebhook(payload, config) {
    try {
        var timestamp = new Date().toISOString();
        var payloadString = JSON.stringify(payload);
        var payloadWithTimestamp = `${timestamp}:${payloadString}`;

        var signature = generateHMACSignature(payloadWithTimestamp,
config.webhookSecret);

        if (!signature) {
            gs.error('Failed to generate signature');
            return false;
        }

        gs.info('Generated signature: ' + signature);

        var request = new sn_ws.RESTMessageV2();
        request.setEndpoint(config.webhookUrl);
        request.setHttpMethod('POST');

        request.setRequestHeader('Content-Type', 'application/json');
        request.setRequestHeader('x-amzn-event-signature', signature);
        request.setRequestHeader('x-amzn-event-timestamp', timestamp);

        request.setRequestBody(payloadString);

        var response = request.execute();
        var httpStatus = response.getStatusCode();
        var responseBody = response.getBody();

        if (httpStatus >= 200 && httpStatus < 300) {
            gs.info('Webhook sent successfully. Status: ' + httpStatus);
            return true;
        } else {
```

```
        gs.error('Webhook failed. Status: ' + httpStatus + ', Response: ' +
responseBody);
        return false;
    }

    } catch (ex) {
        gs.error('Error sending webhook: ' + ex.getMessage());
        return false;
    }
}

function createReference(field) {
    if (!field || field.nil()) {
        return null;
    }

    return {
        link: field.getLink(true),
        value: field.toString()
    };
}

function getStringValue(field) {
    if (!field || field.nil()) {
        return null;
    }
    return field.toString();
}

function getIntValue(field) {
    if (!field || field.nil()) {
        return null;
    }
    var val = parseInt(field.toString());
    return isNaN(val) ? null : val;
}

var eventType = (current.operation() == 'insert') ? "create" : "update";

var incidentEvent = {
    eventType: eventType.toString(),
    sysId: current.sys_id.toString(),
    priority: getStringValue(current.priority),
    impact: getStringValue(current.impact),
```

```
        active: getStringValue(current.active),
        urgency: getStringValue(current.urgency),
        description: getStringValue(current.description),
        shortDescription: getStringValue(current.short_description),
        parent: getStringValue(current.parent),
        incidentState: getStringValue(current.incident_state),
        severity: getStringValue(current.severity),
        problem: createReference(current.problem),
        additionalContext: {}
    };

    incidentEvent.additionalContext = {
        number: current.number.toString(),
        opened_at: getStringValue(current.opened_at),
        opened_by: current.opened_by.nil() ? null :
current.opened_by.getDisplayValue(),
        assigned_to: current.assigned_to.nil() ? null :
current.assigned_to.getDisplayValue(),
        category: getStringValue(current.category),
        subcategory: getStringValue(current.subcategory),
        knowledge: getStringValue(current.knowledge),
        made_sla: getStringValue(current.made_sla),
        major_incident: getStringValue(current.major_incident)
    };

    for (var key in incidentEvent.additionalContext) {
        if (incidentEvent.additionalContext[key] === null) {
            delete incidentEvent.additionalContext[key];
        }
    }

    gs.info(JSON.stringify(incidentEvent, null, 2)); // Pretty print for logging only

    if (WEBHOOK_CONFIG.webhookUrl && WEBHOOK_CONFIG.webhookSecret) {
        callWebhook(incidentEvent, WEBHOOK_CONFIG);
    } else {
        gs.info('Webhook not configured.');
```

```
    }
})(current, previous);
```

Jika Anda memilih untuk mendaftarkan ServiceNow koneksi Anda dari halaman Penyedia Kemampuan, Anda sekarang harus menavigasi ke Ruang DevOps Agen yang ingin Anda selidiki

tiket ServiceNow insiden, pilih Kemampuan → Komunikasi, lalu daftarkan ServiceNow instans yang Anda daftarkan di halaman Penyedia Kemampuan. Sekarang, semuanya harus diatur, dan semua insiden di mana penelepon diatur ke “Administrator Masalah” (untuk meniru izin yang Anda berikan kepada klien AWS DevOps OAuth) akan memicu penyelidikan respons insiden di Ruang Agen yang dikonfigurasi. DevOps Anda dapat menguji ini dengan membuat insiden baru ServiceNow dan menyetel bidang Penelepon insiden tersebut sebagai “Administrator Masalah.”

The screenshot shows the ServiceNow 'Incident - Create' form. The form includes the following fields and controls:

- Number:** INC0010001
- Caller:** Problem Administrator
- Watch list:** Includes lock and refresh icons.
- Short description:** Investigate the CloudWatch alarm [ALARM] [us-east-1] abeyjohn-AlarmsAlwaysRed
- Opened:** 2025-11-14 12:45:19
- Urgency:** 3 - Low
- State:** New
- Buttons:** Submit and Resolve
- Comments:** A large text area for customer-visible comments.
- Related Search Results:** A button to view related search results.

ServiceNow pembaruan tiket

Selama semua Investigasi respons insiden yang dipicu, DevOps Agen Anda akan memberikan pembaruan temuan utamanya, analisis akar penyebab, dan rencana mitigasi ke dalam tiket asal. Temuan agen diposting ke komentar insiden, dan saat ini kami hanya akan memposting catatan agen jenis `finding`, `cause`, `investigation_summary`, `mitigation_summary`, dan pembaruan status investigasi (misalnya `AWS DevOps Agent started/finished its investigation`).

Contoh perutean tiket dan orkestrasi

Skenario: Memfilter insiden mana yang dikirim ke Ruang Agen DevOps

Ini adalah skenario sederhana tetapi membutuhkan beberapa konfigurasi ServiceNow untuk membuat bidang ServiceNow untuk melacak sumber insiden. Untuk tujuan contoh ini, buat bidang Sumber (`u_source`) baru menggunakan pembuat formulir SNOW. Ini akan memungkinkan melacak sumber insiden dan menggunakannya untuk merutekan permintaan dari sumber tertentu ke Ruang DevOps Agen. Perutean dilakukan dengan membuat Aturan Bisnis Layanan Sekarang dan di tab Kapan menjalankan pengaturan “Kapan” pemacu dan “Kondisi Filter.” Dalam contoh ini kondisi filter diatur sebagai berikut:

Business Rule
New record

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met. [More Info](#)

Name: Application:

Table: Active:

Advanced:

When to run | Actions | Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When: Insert:

Order: Update:

Delete:

Query:

Filter Conditions:

Role conditions:

Skenario: Insiden perutean di beberapa Ruang Agen DevOps

Contoh ini menunjukkan bagaimana memicu Investigasi di Ruang DevOps Agen B ketika urgensinya adalah 1, kategori adalah Software, atau Layanan adalah AWS, dan memicu Investigasi di Ruang DevOps Agen A ketika layanan tersebut AWS, dan sumbernya Dynatrace.

Skenario ini dapat dicapai dengan dua cara. Skrip webhook itu sendiri dapat diperbarui untuk memasukkan logika bisnis ini. Dalam skenario ini kami akan menunjukkan bagaimana mencapainya dengan Aturan ServiceNow Bisnis, untuk transparansi dan menyederhanakan debugging. Routing dilakukan dengan membuat dua Aturan Bisnis Service Now.

- Buat Aturan Bisnis ServiceNow untuk Ruang DevOps Agen A dan buat kondisi menggunakan pembuat kondisi untuk hanya mengirim peristiwa berdasarkan kondisi yang kami tentukan.

Business Rule
New record

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met. [More Info](#)

Name: Application:

Table: Active:

Advanced:

Submit

When to run | Actions | Advanced

Specify whether the business rule should run on **Insert** or **Update**. Use **Filter Conditions** to specify under which conditions the business rule should run.

When: Insert:

Order: Update:

Delete:

Query:

Filter Conditions:

All of these conditions must be met

Urgency is 1 - High

Category is Software

or Service is AWS

Role conditions:

- Selanjutnya, buat Aturan Bisnis lain di ServiceNow untuk AgentSpace B yang aturan bisnisnya hanya akan dipicu ketika Layanan AWS dan sumbernya adalah Dynatrace.

Business Rule
New record

A business rule is a server-side script that runs when a record is displayed, inserted, deleted, or when a table is queried. Use business rules to automatically change values in form fields when the specified conditions are met. [More Info](#)

Name: Send events to Agent Space B
Table: Incident [incident]

Application: Global
Active:
Advanced:

When to run: before
Order: 100

Filter Conditions: Add Filter Condition Add OR Clause
All of these conditions must be met
Service is AWS
Source(u_integ_source) contains Dynatrace

Role conditions:

Insert:
Update:
Delete:
Query:

Submit

Sekarang, ketika Anda membuat Insiden baru yang cocok dengan kondisi yang ditentukan, itu akan memicu penyelidikan pada DevOps Agen Space A atau DevOps Agen Space B, memberi Anda kontrol halus atas perutean insiden.

Menghubungkan Slack

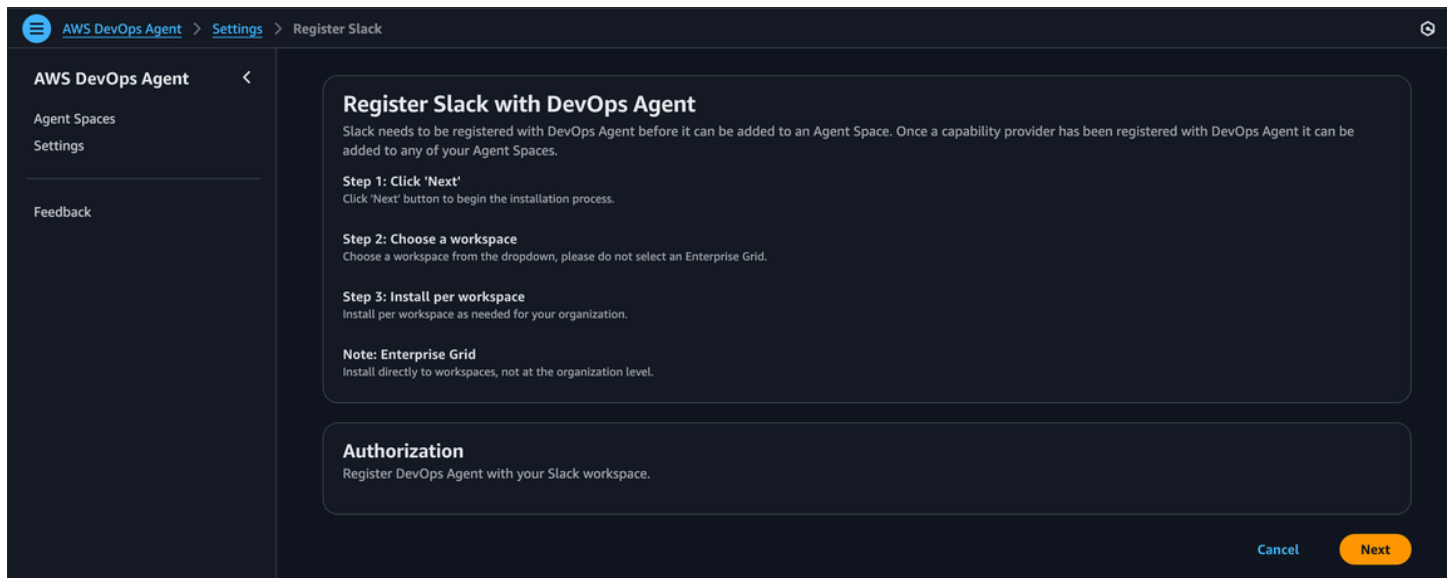
Anda dapat mengonfigurasi AWS DevOps Agen untuk memperbarui saluran Slack yang Anda pilih dengan temuan kunci investigasi respons insiden, analisis akar penyebab, dan rencana mitigasi yang dihasilkan.

Sebelum Anda mulai

Slack harus terdaftar dengan DevOps Agen sebelum dapat ditambahkan ke Ruang Agen. Untuk mengintegrasikan AWS DevOps Agen dengan Slack, Anda harus memenuhi persyaratan ini:

- Memiliki akses ke ruang kerja Slack dengan kemampuan untuk menginstal dan mengotorisasi aplikasi pihak ketiga
- Telah mengidentifikasi saluran Slack tempat Anda ingin AWS DevOps Agen mengirim pemberitahuan

Daftarkan integrasi Slack dengan Agen AWS DevOps



1. Dari halaman Penyedia Kemampuan di konsol AWS DevOps Agen, temukan Slack di bagian Penyedia yang tersedia di bawah Komunikasi dan klik Daftar.
2. Pilih tombol Register.
3. Anda akan diarahkan ke Slack untuk mengotorisasi aplikasi AWS DevOps Agen untuk ruang kerja Anda.
4. Pada halaman otorisasi Slack, instal langsung ke ruang kerja, bukan di tingkat organisasi.
5. Pilih ruang kerja dari dropdown. Jangan pilih Enterprise Grid.
6. Instal per ruang kerja sesuai kebutuhan untuk organisasi Anda.
7. Tinjau cakupan yang diminta dan klik Izinkan untuk mengotorisasi integrasi.
8. Setelah otorisasi, Anda akan kembali ke konsol AWS DevOps Agen.

Kaitkan Slack dengan Ruang DevOps Agen Anda

Setelah mendaftarkan Slack di Ruang DevOps Agen Anda, Anda dapat mengaitkannya dengan Ruang DevOps Agen Anda:

1. Dari tab Kemampuan dalam konfigurasi Anda AgentSpace, navigasikan ke Communications > Slack.
2. Pilih Tambahkan Slack
3. Masukkan ID Saluran

4. Pilih Buat untuk menyelesaikan konfigurasi Slack.

Note

Pengguna bot agen harus ditambahkan ke saluran pribadi sebelum dapat mengirim pesan.

Important

Menghapus instalasi aplikasi Slack dapat mengakibatkan aplikasi Slack tidak dapat diinstal ulang. Harap hindari menghapus aplikasi Slack.

Memanggil DevOps Agen melalui Webhook

Webhook memungkinkan sistem eksternal untuk secara otomatis memicu investigasi AWS DevOps Agen. Ini memungkinkan integrasi dengan sistem tiket, alat pemantauan, dan platform lain yang dapat mengirim permintaan HTTP ketika insiden terjadi.

Prasyarat

Sebelum mengonfigurasi akses webhook, pastikan Anda memiliki:

- Ruang Agen yang dikonfigurasi di AWS DevOps Agen
- Akses ke konsol AWS DevOps Agen
- Sistem eksternal yang akan mengirim permintaan webhook

Jenis webhook

AWS DevOps Agen mendukung jenis webhook berikut:

- Integration-specific webhooks — Dibuat secara otomatis saat Anda mengonfigurasi integrasi pihak ketiga seperti Dynatrace, Splunk, Datadog, New Relic, atau Slack. ServiceNow Webhook ini dikaitkan dengan integrasi spesifik dan menggunakan metode otentikasi yang ditentukan oleh jenis integrasi

- Webhook generik — Dapat dibuat secara manual untuk memicu penyelidikan dari sumber apa pun yang tidak tercakup oleh integrasi tertentu. Webhook generik saat ini menggunakan otentikasi HMAC (token pembawa saat ini tidak tersedia).
- Grafana alert webhooks — Grafana dapat mengirim notifikasi peringatan langsung AWS DevOps ke Agen melalui titik kontak webhook. Untuk petunjuk penyiapan termasuk templat notifikasi khusus, lihat [Menghubungkan Grafana](#).

Metode otentikasi Webhook

Metode otentikasi untuk webhook Anda bergantung pada integrasi mana yang terkait dengannya:

Otentikasi HMAC - Digunakan oleh:

- Webhook integrasi Dynatrace
- Webhook generik (tidak ditautkan ke integrasi pihak ketiga tertentu)

Otentikasi token pembawa - Digunakan oleh:

- Webhook integrasi splunk
- Webhook integrasi Datadog
- Webhook integrasi Relic baru
- ServiceNow integrasi webhooks
- Webhook integrasi slack
- Webhook integrasi Grafana

Memahami otentikasi HMAC

HMAC (Hash-based Message Authentication Code) adalah mekanisme kriptografi yang memverifikasi integritas dan keaslian permintaan webhook. Ketika Anda mengirim webhook dengan otentikasi HMAC, Anda menghasilkan tanda tangan dengan hashing timestamp permintaan dan payload bersama-sama menggunakan kunci rahasia Anda dengan algoritma. SHA-256 AWS DevOps Agen secara independen menghitung hash yang sama di sisinya dan membandingkan dua tanda tangan. Jika cocok, permintaan diterima.

Karena stempel waktu termasuk dalam tanda tangan, HMAC juga menyediakan perlindungan replay — AWS DevOps Agen dapat menolak permintaan dengan stempel waktu yang terlalu jauh di masa lalu, mencegah penyerang menangkap dan mengirim ulang permintaan yang valid.

Memilih antara token HMAC dan Bearer

Pertimbangan	HMAC	Token pembawa
Kompleksitas pengaturan	Lebih kompleks — klien Anda harus menghitung tanda tangan untuk setiap permintaan menggunakan stempel waktu dan muatan	Simpler - sertakan token statis di <code>Authorization</code> header
Integritas muatan	Terverifikasi - setiap modifikasi pada payload setelah penandatanganan membatalkan tanda tangan	Tidak diverifikasi — token mengautentikasi pengirim tetapi tidak melindungi konten muatan
Perlindungan replay	Built-in — stempel waktu di tanda tangan memungkinkan server untuk menolak permintaan basi	Tidak built-in — token yang ditangkap dapat digunakan kembali hingga diputar
Risiko eksposur rahasia	Lebih rendah — rahasia tidak pernah dikirimkan dalam permintaan; hanya tanda tangan yang dihitung yang dikirim	Lebih tinggi — token dikirim di setiap header permintaan, meningkatkan eksposur jika lalu lintas dicegat
Kapan harus digunakan	Direkomendasikan ketika Anda membutuhkan jaminan keamanan yang lebih kuat, seperti untuk webhook generik atau lingkungan dengan persyaratan kepatuhan yang ketat	Sesuai ketika kemudahan integrasi menjadi prioritas dan transportasi jaringan Anda dipercaya, seperti untuk integrasi SaaS terkelola melalui HTTPS

Catatan: Metode otentikasi ditentukan oleh jenis integrasi. Integration-specific webhook (Splunk, Datadog, New Relic,, ServiceNow Slack, Grafana) menggunakan otentikasi token pembawa. Dynatrace dan webhook generik menggunakan otentikasi HMAC. Anda tidak dapat mengubah metode otentikasi untuk webhook khusus integrasi.

Mengkonfigurasi akses webhook

Langkah 1: Arahkan ke konfigurasi webhook

1. Masuk ke Konsol AWS Manajemen dan arahkan ke konsol AWS DevOps Agen
2. Pilih Ruang Agen Anda
3. Buka tab Kemampuan
4. Di bagian Webhook, klik Konfigurasi

Langkah 2: Hasilkan kredensial webhook

Untuk webhook khusus integrasi:

Webhook dibuat secara otomatis saat Anda menyelesaikan konfigurasi integrasi pihak ketiga. URL titik akhir webhook dan kredensial disediakan di akhir proses penyiapan integrasi.

Untuk webhook generik:

1. Klik Hasilkan webhook
2. Sistem akan menghasilkan key pair HMAC
3. Simpan kunci dan rahasia yang dihasilkan dengan aman — Anda tidak akan dapat mengambilnya lagi
4. Salin URL titik akhir webhook yang disediakan

Langkah 3: Konfigurasi sistem eksternal Anda

Gunakan URL endpoint webhook dan kredensial untuk mengonfigurasi sistem eksternal Anda untuk mengirim permintaan ke Agen. AWS DevOps Langkah-langkah konfigurasi spesifik tergantung pada sistem eksternal Anda.

Mengelola kredenal webhook

Menghapus kredensial - Untuk menghapus kredensial webhook, buka bagian konfigurasi webhook dan klik Hapus. Setelah menghapus kredensial, titik akhir webhook tidak akan lagi menerima permintaan sampai Anda menghasilkan kredensial baru.

Regenerating credentials — Untuk menghasilkan kredensial baru, hapus kredensial yang ada terlebih dahulu, lalu buat key pair atau token baru.

Menggunakan webhook

Format permintaan webhook

Untuk memicu investigasi, sistem eksternal Anda harus mengirim permintaan HTTP POST ke URL endpoint webhook.

Untuk Versi 1 (otentikasi HMAC):

Header:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

Tanda tangan HMAC dihasilkan dengan menandatangani badan permintaan dengan kunci rahasia Anda menggunakan SHA-256.

Untuk Versi 2 (otentikasi token pembawa):

Header:

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Permintaan badan:

Badan permintaan harus menyertakan informasi tentang insiden tersebut:

```
json
```

```
{
  "title": "Incident title",
  "severity": "high",
  "affectedResources": ["resource-id-1", "resource-id-2"],
  "timestamp": "2025-11-23T18:00:00Z",
  "description": "Detailed incident description",
  "data": {
    "metadata": {
      "region": "us-east-1",
      "environment": "production"
    }
  }
}
```

Skema muatan:

```
{
  eventType: 'incident';
  incidentId: string;
  action: 'created' | 'updated' | 'closed' | 'resolved';
  priority: "CRITICAL" | "HIGH" | "MEDIUM" | "LOW" | "MINIMAL";
  title: string;
  description?: string;
  timestamp?: string;
  service?: string;
  // The original event generated by service is attached here.
  data?: object;
}
```

Contoh kode

Versi 1 (otentikasi HMAC) -: JavaScript

```
const crypto = require('crypto');

// Webhook configuration
const webhookUrl = 'https://your-webhook-endpoint.amazonaws.com/invoke';
const webhookSecret = 'your-webhook-secret-key';

// Incident data
const incidentData = {
  eventType: 'incident',
  incidentId: 'incident-123',
```

```
    action: 'created',
    priority: "HIGH",
    title: 'High CPU usage on production server',
    description: 'High CPU usage on production server host ABC in AWS account 1234
region us-east-1',
    timestamp: new Date().toISOString(),
    service: 'MyTestService',
    data: {
      metadata: {
        region: 'us-east-1',
        environment: 'production'
      }
    }
  };

// Convert data to JSON string
const payload = JSON.stringify(incidentData);
const timestamp = new Date().toISOString();
const hmac = crypto.createHmac("sha256", webhookSecret);
hmac.update(`${timestamp}:${payload}`, "utf8");
const signature = hmac.digest("base64");

// Send the request
fetch(webhookUrl, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'x-amzn-event-timestamp': timestamp,
    'x-amzn-event-signature': signature
  },
  body: payload
})
.then(res => {
  console.log(`Status Code: ${res.status}`);
  return res.text();
})
.then(data => {
  console.log('Response:', data);
})
.catch(error => {
  console.error('Error:', error);
});
```

Versi 1 (otentikasi HMAC) - cURL:

```
#!/bin/bash

# Configuration
WEBHOOK_URL="https://event-ai.us-east-1.api.aws/webhook/generic/YOUR_WEBHOOK_ID"
SECRET="YOUR_WEBHOOK_SECRET"

# Create payload
TIMESTAMP=$(date -u +%Y-%m-%dT%H:%M:%S.000Z)
INCIDENT_ID="test-alert-$(date +%s)"

PAYLOAD=$(cat <<EOF
{
"eventType": "incident",
"incidentId": "$INCIDENT_ID",
"action": "created",
"priority": "HIGH",
"title": "Test Alert",
"description": "Test alert description",
"service": "TestService",
"timestamp": "$TIMESTAMP"
}
EOF
)

# Generate HMAC signature
SIGNATURE=$(echo -n "${TIMESTAMP}:${PAYLOAD}" | openssl dgst -sha256 -hmac "$SECRET" -
binary | base64)

# Send webhook
curl -X POST "$WEBHOOK_URL" \
-H "Content-Type: application/json" \
-H "x-amzn-event-timestamp: $TIMESTAMP" \
-H "x-amzn-event-signature: $SIGNATURE" \
-d "$PAYLOAD"
```

Versi 2 (otentikasi token pembawa) -: JavaScript

```
function sendEventToWebhook(webhookUrl, secret) {
  const timestamp = new Date().toISOString();

  const payload = {
```

```
    eventType: 'incident',
    incidentId: 'incident-123',
    action: 'created',
    priority: "HIGH",
    title: 'Test Alert',
    description: 'Test description',
    timestamp: timestamp,
    service: 'TestService',
    data: {}
  };

  fetch(webhookUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "x-amzn-event-timestamp": timestamp,
      "Authorization": `Bearer ${secret}`, // Fixed: template literal
    },
    body: JSON.stringify(payload),
  });
}
```

Versi 2 (otentikasi token pembawa) - cURL:

```
#!/bin/bash

# Configuration
WEBHOOK_URL="https://event-ai.us-east-1.api.aws/webhook/generic/YOUR_WEBHOOK_ID"
SECRET="YOUR_WEBHOOK_SECRET"

# Create payload
TIMESTAMP=$(date -u +%Y-%m-%dT%H:%M:%S.000Z)
INCIDENT_ID="test-alert-$(date +%s)"

PAYLOAD=$(cat <<EOF
{
  "eventType": "incident",
  "incidentId": "$INCIDENT_ID",
  "action": "created",
  "priority": "HIGH",
  "title": "Test Alert",
  "description": "Test alert description",
  "service": "TestService",
```

```
"timestamp": "$TIMESTAMP"
}
EOF
)

# Send webhook
curl -X POST "$WEBHOOK_URL" \
-H "Content-Type: application/json" \
-H "x-amzn-event-timestamp: $TIMESTAMP" \
-H "Authorization: Bearer $SECRET" \
-d "$PAYLOAD"
```

Memecahkan masalah webhook

Jika Anda tidak menerima 200

Sebuah 200 dan pesan seperti webhook diterima menunjukkan otentikasi berlalu dan pesan telah antri untuk sistem untuk memverifikasi dan memproses. Jika Anda tidak mendapatkan 200 tetapi 4xx kemungkinan besar ada yang salah dengan otentikasi atau header. Coba kirim secara manual menggunakan opsi curl untuk membantu men-debug otentikasi.

Jika Anda menerima 200 tetapi penyelidikan tidak dimulai

Kemungkinan penyebabnya adalah muatan yang salah format.

1. Periksa stempel waktu dan id insiden diperbarui dan unik. Pesan duplikat di-deduplikasi.
2. Periksa pesan JSON yang valid
3. Periksa formatnya benar

Jika Anda menerima 200 dan investigasi segera dibatalkan

Kemungkinan besar Anda telah mencapai batas untuk bulan itu. Silakan berbicara dengan AWS kontak Anda untuk meminta perubahan batas tarif jika sesuai.

Topik terkait

- [the section called “Membuat Ruang Agen”](#)
- [the section called “Apa itu Aplikasi Web DevOps Agen?”](#)
- [the section called “DevOps Izin Agen IAM”](#)

Mengintegrasikan AWS DevOps Agen dengan Amazon EventBridge

Anda dapat mengintegrasikan AWS DevOps Agen dengan aplikasi berbasis peristiwa dengan menggunakan peristiwa yang terjadi selama siklus hidup investigasi dan mitigasi. AWS DevOps Agen mengirimkan acara ke Amazon EventBridge ketika keadaan investigasi atau mitigasi berubah. Anda kemudian dapat membuat EventBridge aturan yang mengambil tindakan berdasarkan peristiwa ini.

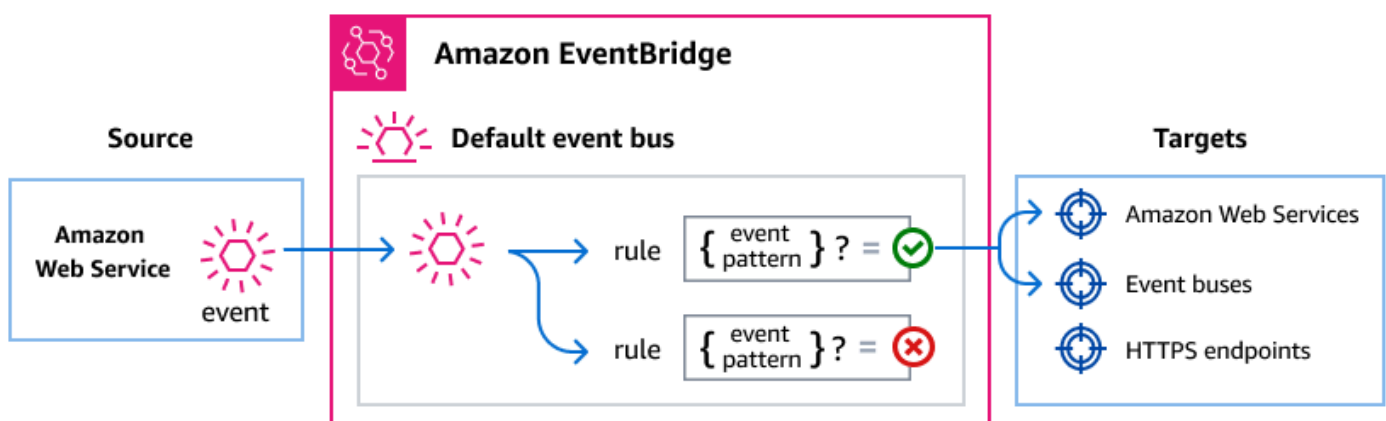
Misalnya, Anda dapat membuat aturan yang melakukan tindakan berikut:

- Memanggil fungsi AWS Lambda untuk memproses hasil investigasi ketika penyelidikan selesai.
- Kirim notifikasi Amazon SNS saat investigasi gagal atau habis waktu.
- Perbarui sistem tiket saat penyelidikan baru dibuat.
- Mulai alur kerja AWS Step Functions saat tindakan mitigasi selesai.

Bagaimana EventBridge rute AWS DevOps Acara agen

AWS DevOps Agen mengirimkan acara ke bus acara EventBridge default. EventBridge kemudian mengevaluasi peristiwa terhadap aturan yang Anda buat. Saat acara cocok dengan pola acara aturan, EventBridge kirimkan acara ke target yang ditentukan.

Diagram berikut menunjukkan bagaimana EventBridge rute peristiwa AWS DevOps Agen.



1. AWS DevOps Agen mengirimkan peristiwa ke bus peristiwa EventBridge default ketika status siklus hidup investigasi atau mitigasi berubah.

2. EventBridge mengevaluasi acara terhadap aturan yang Anda buat.
3. Jika acara cocok dengan pola acara aturan, EventBridge kirimkan acara ke target yang ditentukan dalam aturan.

AWS DevOps Acara agen

AWS DevOps Agen mengirimkan acara berikut ke EventBridge. Semua acara menggunakan sumbernya `aws.aidevops`.

Acara investigasi yang didukung

jenis-detail	Deskripsi
Investigation Created	Investigasi dibuat di ruang agen.
Investigation Priority Updated	Prioritas investigasi diubah.
Investigation In Progress	Investigasi memulai analisis aktif.
Investigation Completed	Investigasi berhasil diselesaikan dengan temuan.
Investigation Failed	Investigasi mengalami kesalahan dan tidak dapat diselesaikan.
Investigation Timed Out	Investigasi melebihi durasi maksimum yang diizinkan.
Investigation Cancelled	Investigasi dibatalkan sebelum selesai.
Investigation Pending Triage	Investigasi sedang menunggu triase sebelum analisis aktif dimulai.
Investigation Linked	Investigasi terkait dengan insiden atau tiket terkait.
Investigation Skipped	Investigasi dilewati karena cocok dengan kriteria lewati yang ditentukan dalam keterampilan.

Acara mitigasi yang didukung

jenis-detail	Deskripsi
Mitigation In Progress	Tindakan mitigasi dimulai.
Mitigation Completed	Tindakan mitigasi selesai dengan sukses.
Mitigation Failed	Tindakan mitigasi mengalami kesalahan dan tidak dapat diselesaikan.
Mitigation Timed Out	Tindakan mitigasi melebihi durasi maksimum yang diizinkan.
Mitigation Cancelled	Tindakan mitigasi dibatalkan sebelum selesai.

Untuk deskripsi bidang terperinci dan contoh peristiwa, lihat [the section called “AWS DevOps Referensi detail acara agen”](#).

Membuat pola acara yang cocok AWS DevOps Acara agen

EventBridge aturan menggunakan pola acara untuk memilih peristiwa dan merutekan mereka ke target. Pola acara cocok dengan struktur peristiwa yang ditanganinya. Anda membuat pola acara untuk memfilter peristiwa AWS DevOps Agen berdasarkan bidang acara.

Contoh berikut menunjukkan pola peristiwa untuk kasus penggunaan umum.

Cocokkan semua acara AWS DevOps Agent

Pola acara berikut cocok dengan semua acara dari AWS DevOps Agen.

```
{
  "source": ["aws.aidevops"]
}
```

Cocokkan saja acara investigasi

Pola peristiwa berikut menggunakan kecocokan awalan untuk memilih hanya peristiwa siklus hidup investigasi.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Cocokkan hanya acara penyelesaian dan kegagalan

Pola peristiwa berikut cocok dengan peristiwa untuk investigasi dan mitigasi yang diselesaikan atau gagal.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

Pertandingan acara untuk ruang agen tertentu

Pola acara berikut cocok dengan peristiwa dari ruang agen tertentu.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Untuk informasi selengkapnya tentang pola peristiwa, lihat [pola EventBridge peristiwa Amazon](#) di Panduan EventBridge Pengguna Amazon.

EventBridge Izin Amazon

AWS DevOps Agen tidak memerlukan izin tambahan untuk mengirimkan acara ke EventBridge. Acara dikirim ke bus acara default secara otomatis.

Bergantung pada target yang Anda konfigurasi untuk EventBridge aturan, Anda mungkin perlu menambahkan izin tertentu. Untuk informasi selengkapnya tentang izin yang diperlukan untuk target, lihat [Menggunakan kebijakan berbasis sumber daya untuk Amazon EventBridge di Panduan Pengguna Amazon](#). EventBridge

EventBridge Sumber daya tambahan

Untuk informasi selengkapnya tentang EventBridge konsep dan konfigurasi, lihat topik berikut di Panduan EventBridge Pengguna Amazon:

- [EventBridge bus acara](#)
- [EventBridge acara](#)
- [EventBridge pola acara](#)
- [EventBridge aturan](#)
- [EventBridge target](#)

AWS DevOps Referensi detail acara agen

Peristiwa dari AWS layanan memiliki bidang metadata umum, termasuk `source`, `detail-type`, `account`, `region`, dan `time`. Peristiwa ini juga berisi detail bidang dengan data khusus untuk layanan. Untuk acara AWS DevOps Agen, `source` selalu `aws.aidevops` dan `detail-type` mengidentifikasi peristiwa tertentu.

Peristiwa investigasi

`detail-type` Nilai-nilai berikut mengidentifikasi peristiwa investigasi:

- Investigation Created
- Investigation Priority Updated
- Investigation In Progress
- Investigation Completed
- Investigation Failed
- Investigation Timed Out
- Investigation Cancelled
- Investigation Pending Triage
- Investigation Linked

- Investigation Skipped

detail-typeBidang source dan disertakan di bawah ini karena berisi nilai spesifik untuk peristiwa AWS DevOps Agen. Untuk definisi bidang metadata lain yang disertakan dalam semua peristiwa, lihat [Struktur peristiwa di Referensi EventBridge](#) Acara Amazon.

Berikut ini adalah struktur JSON untuk peristiwa investigasi.

```
{
  . . . ,
  "detail-type" : "string",
  "source" : "aws.aidevops",
  . . . ,
  "detail" : {
    "version" : "string",
    "metadata" : {
      "agent_space_id" : "string",
      "task_id" : "string",
      "execution_id" : "string"
    },
    "data" : {
      "task_type" : "string",
      "priority" : "string",
      "status" : "string",
      "created_at" : "string",
      "updated_at" : "string",
      "summary_record_id" : "string"
    }
  }
}
```

detail-typeMengidentifikasi jenis acara. Untuk acara investigasi, ini adalah salah satu nama acara yang tercantum sebelumnya.

sourceMengidentifikasi layanan yang menghasilkan acara. Untuk acara AWS DevOps Agen, nilai ini adalah `aws.aidevops`.

detailObjek JSON yang berisi data khusus peristiwa. `detail`Objek mencakup bidang-bidang berikut:

- `version(string)` - Versi skema dari detail acara. Saat ini `1.0.0`.

- `metadata.agent_space_id(string)` — Pengenal unik dari ruang agen tempat peristiwa itu berasal.
- `metadata.task_id(string)` — Pengidentifikasi unik tugas.
- `metadata.execution_id(string)` - Pengidentifikasi unik dari eksekusi dijalankan. Hadir ketika eksekusi telah ditugaskan untuk penyelidikan.
- `data.task_type(string)` — Jenis tugas. Nilai:INVESTIGATION.
- `data.priority(string)` - Tingkat prioritas. Nilai:CRITICAL,HIGH,MEDIUM,LOW,MINIMAL.
- `data.status(string)` - Status saat ini.
Nilai:PENDING_START,IN_PROGRESS,COMPLETED,FAILED,TIMED_OUT,CANCELLED,PENDING_TRIAGE,I
- `data.created_at(string)` — Stempel waktu ISO 8601 saat tugas dibuat.
- `data.updated_at(string)` - Stempel waktu ISO 8601 saat tugas terakhir diperbarui.
- `data.summary_record_id(string)` — Pengidentifikasi catatan ringkasan yang berisi temuan investigasi. Termasuk ketika ringkasan dibuat untuk penyelidikan yang telah selesai. Anda dapat mengambil konten ringkasan melalui AWS DevOps Agent API dengan menggunakan pengenal ini untuk mencari catatan jurnal dengan jenis rekaman. `investigation_summary_md`

Contoh: Investigasi Selesai acara

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789015",
  "detail-type": "Investigation Completed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:10:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "b2c3d4e5-6789-01ab-cdef-example22222"
    }
  },
  "data": {
```

```
    "task_type": "INVESTIGATION",
    "priority": "CRITICAL",
    "status": "COMPLETED",
    "created_at": "2026-03-12T18:00:00Z",
    "updated_at": "2026-03-12T18:10:00Z",
    "summary_record_id": "d4e5f6g7-6789-01ab-cdef-example44444"
  }
}
```

Contoh: Acara Investigasi Gagal

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789016",
  "detail-type": "Investigation Failed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:10:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "b2c3d4e5-6789-01ab-cdef-example22222"
    }
  },
  "data": {
    "task_type": "INVESTIGATION",
    "priority": "CRITICAL",
    "status": "FAILED",
    "created_at": "2026-03-12T18:00:00Z",
    "updated_at": "2026-03-12T18:10:00Z"
  }
}
```

Acara mitigasi

detail-typeNilai-nilai berikut mengidentifikasi peristiwa mitigasi:

- Mitigation In Progress
- Mitigation Completed
- Mitigation Failed
- Mitigation Timed Out
- Mitigation Cancelled

detail-typeBidang source dan disertakan di bawah ini karena berisi nilai spesifik untuk peristiwa AWS DevOps Agen. Untuk definisi bidang metadata lain yang disertakan dalam semua peristiwa, lihat [Struktur peristiwa di Referensi EventBridge](#) Acara Amazon.

Berikut ini adalah struktur JSON untuk peristiwa mitigasi.

```
{
  . . . ,
  "detail-type" : "string",
  "source" : "aws.aidevops",
  . . . ,
  "detail" : {
    "version" : "string",
    "metadata" : {
      "agent_space_id" : "string",
      "task_id" : "string",
      "execution_id" : "string"
    },
    "data" : {
      "task_type" : "string",
      "priority" : "string",
      "status" : "string",
      "created_at" : "string",
      "updated_at" : "string",
      "summary_record_id" : "string"
    }
  }
}
```

detail-type Mengidentifikasi jenis acara. Untuk acara mitigasi, ini adalah salah satu nama acara yang tercantum sebelumnya.

source Mengidentifikasi layanan yang menghasilkan acara. Untuk acara AWS DevOps Agen, nilai ini adalah `aws.aidevops`.

detail Objek JSON yang berisi data khusus peristiwa. `detail` objek mencakup bidang-bidang berikut:

- `version(string)` - Versi skema dari detail acara. Saat ini `1.0.0`.
- `metadata.agent_space_id(string)` — Pengenal unik dari ruang agen tempat peristiwa itu berasal.
- `metadata.task_id(string)` — Pengidentifikasi unik tugas.
- `metadata.execution_id(string)` - Pengidentifikasi unik dari eksekusi dijalankan. Hadir ketika eksekusi telah ditugaskan untuk mitigasi.
- `data.task_type(string)` — Jenis tugas. Nilai: `INVESTIGATION`.
- `data.priority(string)` - Tingkat prioritas. Nilai: `CRITICAL,HIGH,MEDIUM,LOW,MINIMAL`.
- `data.status(string)` - Status saat ini.
Nilai: `IN_PROGRESS,COMPLETED,FAILED,TIMED_OUT,CANCELLED`.
- `data.created_at(string)` — Stempel waktu ISO 8601 saat tugas dibuat.
- `data.updated_at(string)` - Stempel waktu ISO 8601 saat tugas terakhir diperbarui.
- `data.summary_record_id(string)` — Pengidentifikasi catatan ringkasan yang berisi temuan mitigasi. Termasuk saat ringkasan dibuat untuk mitigasi yang telah selesai. Anda dapat mengambil konten ringkasan melalui AWS DevOps Agent API dengan menggunakan pengenal ini untuk mencari catatan jurnal dengan jenis rekaman. `mitigation_summary_md`

Contoh: Mitigasi Selesai acara

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-12345678901c",
  "detail-type": "Mitigation Completed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:20:00Z",
  "region": "us-east-1",
  "resources": [
```

```

    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "c3d4e5f6-7890-12ab-cdef-example33333"
    },
    "data": {
      "task_type": "INVESTIGATION",
      "priority": "CRITICAL",
      "status": "COMPLETED",
      "created_at": "2026-03-12T18:00:00Z",
      "updated_at": "2026-03-12T18:20:00Z",
      "summary_record_id": "e5f6g7h8-7890-12ab-cdef-example55555"
    }
  }
}

```

Contoh: Acara Mitigasi Gagal

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-12345678901d",
  "detail-type": "Mitigation Failed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:20:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "c3d4e5f6-7890-12ab-cdef-example33333"
    },
    "data": {

```

```
    "task_type": "INVESTIGATION",
    "priority": "CRITICAL",
    "status": "FAILED",
    "created_at": "2026-03-12T18:00:00Z",
    "updated_at": "2026-03-12T18:20:00Z"
  }
}
```

Log dan Metrik Terjual

Anda dapat memantau ruang agen dan operasi layanan Anda dengan menggunakan CloudWatch metrik dan log Amazon yang dijual. Topik ini menjelaskan CloudWatch metrik yang secara otomatis diterbitkan AWS DevOps Agen ke akun Anda dan log vendid yang dapat Anda konfigurasi untuk pengiriman ke tujuan pilihan Anda.

Metrik yang dijual CloudWatch

AWS DevOps Agen secara otomatis menerbitkan metrik ke Amazon CloudWatch di akun Anda. Metrik ini tersedia tanpa konfigurasi apa pun. Anda dapat menggunakannya untuk memantau penggunaan, melacak aktivitas operasional, dan membuat alarm.

peran tertaut layanan

Agar CloudWatch metrik Amazon dipublikasikan di akun Anda untuk layanan ini, AWS DevOps Agen akan secara otomatis membuat [peran terkait layanan AWSServiceRoleForAIDevOps Peran Tertaut](#) Layanan untuk Anda. Jika peran IAM yang menjalankan API tidak memiliki izin yang sesuai, pembuatan sumber daya akan gagal dengan file. `InvalidParameterException`

Important

Pelanggan yang membuat AgentSpace sebelum 13 Maret 2026 harus membuat Peran Tertaut Layanan `AWSServiceRoleForAIDevOps` secara manual agar CloudWatch metrik AWS DevOps Agen dipublikasikan di akun mereka.

Buat Peran Tertaut Layanan Secara Manual (Untuk pelanggan yang sudah ada)

Lakukan salah satu tindakan berikut:

- Di konsol IAM, buat peran AWSServiceRoleForAIDevOps di bawah layanan AWS DevOps Agen.
- Dari AWS CLI, jalankan perintah berikut:

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Semua metrik diterbitkan di bawah AWS/AIDevOps namespace.

Dimensi

Semua metrik mencakup dimensi berikut.

Dimensi	Deskripsi
AgentSpaceUUID	Pengidentifikasi unik dari ruang agen. Untuk menggabungkan metrik di semua ruang agen di akun Anda, gunakan ekspresi CloudWatch matematika atau hilangkan filter dimensi.

Referensi metrik

Nama metrik	Deskripsi	Unit	Frekuensi penerbitan	Statistik yang berguna
ConsumedChatRequests	Jumlah permintaan obrolan yang dikonsumsi di ruang agen. Untuk mendapatkan jumlah total akun Anda, gunakan SUM statistik di semua	Hitungan	Setiap 5 menit	Jumlah, Rata-rata

Nama metrik	Deskripsi	Unit	Frekuensi penerbitan	Statistik yang berguna
	AgentSpaceUUID dimensi.			
ConsumedInvestigationTime	Waktu yang dihabiskan untuk menjalankan penyelidikan di ruang agen.	Detik	Setiap 5 menit	Jumlah, Rata-rata, Maksimum
ConsumedEvaluationTime	Waktu yang dihabiskan untuk menjalankan evaluasi di ruang agen.	Detik	Setiap 5 menit	Jumlah, Rata-rata, Maksimum
TopologyCompletionCount	Jumlah penyelesaian pemrosesan topologi. AWS DevOps Agen memancarkan metrik ini saat topologi selesai diproses, baik dari pembuatan awal selama orientasi, pembaruan manual, atau penyegaran harian terjadwal.	Hitungan	Event-driven (dipancarkan pada setiap penyelesaian)	Jumlah, SampleCount

Melihat metrik di konsol CloudWatch

1. Buka [konsol CloudWatch](#).
2. Pada panel navigasi, silakan pilih Metrik, dan kemudian pilih Semua metrik.
3. Pilih namespace AWS/AIDevOps.
4. Pilih By AgentSpace untuk melihat metrik ruang agen Anda.

Note

Anda dapat membuat CloudWatch alarm pada metrik ini untuk menerima notifikasi saat penggunaan melebihi ambang batas. Misalnya, buat alarm `ConsumedChatRequests` untuk memantau konsumsi permintaan obrolan.

Prasyarat

Sebelum Anda mengonfigurasi pengiriman log, pastikan Anda memiliki yang berikut:

- AWS Akun aktif dengan akses ke konsol AWS DevOps Agen
- Prinsipal IAM dengan izin untuk CloudWatch pengiriman Log APIs
- (Opsional) Bucket Amazon S3 atau aliran pengiriman Amazon Data Firehose, jika Anda berencana menggunakannya sebagai tujuan log

Log yang dipasok

AWS DevOps Agen mendukung log vended yang memberikan visibilitas ke peristiwa yang diproses oleh ruang agen dan pendaftaran layanan Anda. Log vended menggunakan infrastruktur Amazon CloudWatch Logs untuk mengirimkan log ke tujuan pilihan Anda.

Untuk menggunakan log vended, Anda harus mengonfigurasi tujuan pengiriman. Destinasi berikut didukung:

- Amazon CloudWatch Logs - Grup log di akun Anda
- Amazon S3 - Bucket S3 di akun Anda
- Amazon Data Firehose - Aliran pengiriman Firehose di akun Anda

Jenis log yang didukung

Jenis log tunggal didukung: APPLICATION_LOGS. Jenis log ini mencakup semua peristiwa operasional yang dipancarkan layanan.

Jenis peristiwa log

Tabel berikut merangkum peristiwa yang dicatat AWS DevOps Agen.

Peristiwa	Deskripsi	Tingkat log
Agen inbound event diterima	Agen dipicu oleh sumber terintegrasi dan menerima peristiwa masuk (misalnya, peristiwa PagerDuty insiden).	INFO
Acara masuk agen dijatuhkan	Peristiwa inbound dijatuhkan sebelum agen memprosesnya. Log mencakup alasannya (misalnya, data yang salah bentuk).	Akan Ditentukan Kemudian
Kegagalan komunikasi keluar agen	Komunikasi keluar ke integrasi pihak ketiga gagal. Log termasuk ID tugas dan pengidentifikasi tujuan (misalnya, kesalahan otentikasi).	Akan Ditentukan Kemudian
Pembuatan topologi antri	Pekerjaan pembuatan topologi diantrian untuk diproses.	INFO
Pembuatan topologi dimulai	Pekerjaan penciptaan topologi mulai diproses.	INFO
Pembuatan topologi selesai	Pekerjaan pembuatan topologi menyelesaikan pemrosesan. Acara ini berlaku untuk	INFO

Peristiwa	Deskripsi	Tingkat log
	pembuatan awal, pembaruan, dan penyegaran harian.	
Penemuan sumber daya gagal	Penemuan sumber daya selama pembuatan topologi mengalami kegagalan.	ERROR
Registrasi layanan gagal	Registrasi layanan mengalami kegagalan yang tidak dapat dipulihkan	ERROR
Validasi Webhook gagal	Ketika webhook diterima oleh agen Devops tidak cocok dengan skema yang diharapkan	ERROR
Pembaruan status Validasi Asosiasi	Ketika asosiasi ruang Agen (primary/secondary akun tipikal), status validasi berubah dari valid menjadi tidak valid dan sebaliknya (misalnya , disebabkan oleh peran yang salah, yang tidak dapat diasumsikan oleh layanan).	KESALAHAN/INFO

Izin

AWS DevOps Agen menggunakan [log CloudWatch vendid \(izin V2\)](#) untuk mengirimkan log. Untuk mengatur pengiriman log, peran IAM yang mengonfigurasi pengiriman harus memiliki izin berikut:

- `aidevops:AllowVendedLogDeliveryForResource`— Diperlukan untuk memungkinkan pengiriman log untuk sumber daya ruang agen.
- Izin untuk pengiriman CloudWatch Log APIs (`logs:PutDeliverySource`, `logs:PutDeliveryDestination`, `logs:CreateDelivery`, dan operasi terkait).
- Izin khusus untuk tujuan pengiriman yang Anda pilih.

Untuk kebijakan IAM lengkap yang diperlukan untuk setiap jenis tujuan, lihat topik berikut di Panduan Pengguna CloudWatch Log Amazon:

- [Log dikirim ke CloudWatch Log](#)
- [Log dikirim ke Amazon S3](#)
- [Log dikirim ke Firehose](#)

Konfigurasi pengiriman log (konsol)

AWS DevOps Agen menyediakan dua lokasi di AWS Management Console untuk mengonfigurasi pengiriman log:

- Halaman pengaturan Pendaftaran Layanan - Konfigurasi pengiriman log untuk acara tingkat layanan. Log ini menggunakan layanan ARN (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) sebagai sumber daya.
- Halaman Ruang Agen - Konfigurasi pengiriman log untuk acara yang khusus untuk ruang agen individu. Log ini menggunakan ruang agen ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) sebagai sumber daya.

Untuk mengonfigurasi pengiriman log untuk pendaftaran layanan

1. Buka konsol AWS DevOps Agen di Konsol AWS Manajemen.
2. Pada panel navigasi, silakan pilih Pengaturan.
3. Di tab Penyedia Kemampuan > Log, pilih Konfigurasi.
4. Untuk tipe Tujuan, pilih salah satu dari berikut ini:
5. CloudWatch Log — Pilih atau buat grup log.
6. Amazon S3 — Masukkan ember S3 ARN.
7. Amazon Data Firehose — Pilih atau buat aliran pengiriman Firehose.
8. Untuk Pengaturan tambahan - opsional, Anda dapat menentukan opsi berikut:
 - a. Untuk pemilihan Bidang, pilih nama bidang log yang ingin Anda kirimkan ke tujuan Anda. Anda dapat memilih [bidang log akses](#) dan subset [bidang log akses real-time](#).
 - b. (Hanya Amazon S3) Untuk Partisi, tentukan jalur untuk mempartisi data file log Anda.

- c. (Hanya Amazon S3) Untuk format file yang kompatibel dengan HIVE, Anda dapat memilih kotak centang untuk menggunakan jalur S3 yang kompatibel dengan HIVE. Ini membantu menyederhanakan pemuatan data baru ke alat yang kompatibel dengan HIVE Anda.
 - d. Untuk format Output, tentukan format pilihan Anda.
 - e. Untuk pembatas bidang, tentukan cara memisahkan bidang log.
9. Pilih Simpan.
10. Verifikasi bahwa status pengiriman menunjukkan Aktif.

Untuk mengonfigurasi pengiriman log untuk ruang agen

1. Buka konsol AWS DevOps Agen di Konsol AWS Manajemen.
2. Pilih ruang agen yang ingin Anda konfigurasi.
3. Di tab Konfigurasi, pilih Konfigurasi.
4. Untuk [tipe Tujuan](#), pilih salah satu dari berikut ini:
5. CloudWatch Log — Pilih atau buat grup log.
6. Amazon S3 — Masukkan ember S3 ARN.
7. Amazon Data Firehose — Pilih atau buat aliran pengiriman Firehose.
8. Untuk Pengaturan tambahan — * opsional *, Anda dapat menentukan opsi berikut:
 - a. Untuk pemilihan Bidang, pilih nama bidang log yang ingin Anda kirimkan ke tujuan Anda. Anda dapat memilih [bidang log akses](#) dan subset [bidang log akses real-time](#).
 - b. (Hanya Amazon S3) Untuk Partisi, tentukan jalur untuk mempartisi data file log Anda.
 - c. (Hanya Amazon S3) Untuk format file yang kompatibel dengan HIVE, Anda dapat memilih kotak centang untuk menggunakan jalur S3 yang kompatibel dengan HIVE. Ini membantu menyederhanakan pemuatan data baru ke alat yang kompatibel dengan HIVE Anda.
 - d. Untuk format Output, tentukan format pilihan Anda.
 - e. Untuk pembatas bidang, tentukan cara memisahkan bidang log.
9. Pilih Simpan.
10. Verifikasi bahwa status pengiriman menunjukkan Aktif.

Konfigurasi pengiriman log (CloudWatch API)

Anda juga dapat menggunakan CloudWatch Logs API untuk mengonfigurasi pengiriman log secara terprogram. Pengiriman log kerja terdiri dari tiga elemen:

- A `DeliverySource`- Merupakan sumber daya ruang AWS DevOps Agen yang menghasilkan log.
- A `DeliveryDestination`- Merupakan tujuan di mana log ditulis.
- Pengiriman — Menghubungkan sumber pengiriman ke tujuan pengiriman.

Langkah 1: Buat sumber pengiriman

Gunakan [PutDeliverySource](#) operasi untuk membuat sumber pengiriman. Lewati ARN sumber daya ruang AWS DevOps Agen Anda dan tentukan `APPLICATION_LOGS` sebagai jenis log.

Contoh berikut membuat sumber pengiriman untuk ruang agen:

```
{
  "name": "my-agent-space-delivery-source",
  "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-id",
  "logType": "APPLICATION_LOGS"
}
```

Contoh berikut membuat sumber pengiriman untuk layanan:

```
{
  "name": "my-service-delivery-source",
  "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
  "logType": "APPLICATION_LOGS"
}
```

Langkah 2: Buat tujuan pengiriman

Gunakan [PutDeliveryDestination](#) operasi untuk mengonfigurasi tempat log disimpan. Anda dapat memilih Amazon CloudWatch Log, Amazon S3, atau Amazon Data Firehose.

Contoh berikut membuat tujuan CloudWatch Log:

```
{
  "name": "my-cwl-destination",
  "deliveryDestinationConfiguration": {
    "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/aidevops/my-agent-space"
  },
  "outputFormat": "json"
}
```

Contoh berikut membuat tujuan Amazon S3:

```
{
  "name": "my-s3-destination",
  "deliveryDestinationConfiguration": {
    "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
  },
  "outputFormat": "json"
}
```

Contoh berikut membuat tujuan Amazon Data Firehose:

```
{
  "name": "my-firehose-destination",
  "deliveryDestinationConfiguration": {
    "destinationResourceArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/my-aidevops-log-stream"
  },
  "outputFormat": "json"
}
```

Note

Jika Anda mengirimkan log lintas akun, Anda harus menggunakan [PutDeliveryDestinationPolicy](#) di akun tujuan untuk mengotorisasi pengiriman.

Jika Anda ingin menggunakan CloudFormation, Anda dapat menggunakan yang berikut ini:

- [Pengiriman](#)
- [DeliveryDestination](#)
- [DeliverySource](#)

ResourceArn ini adalah AgentSpaceArn, dan LogType harus APPLICATION_LOGS sebagai jenis log yang didukung.

Langkah 3: Buat pengiriman

Gunakan [CreateDelivery](#) operasi untuk menautkan sumber pengiriman ke tujuan pengiriman.

```
{
  "deliverySourceName": "my-agent-space-delivery-source",
  "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-destination:my-cwl-destination"
}
```

AWS CloudFormation

Anda juga dapat mengonfigurasi pengiriman log AWS CloudFormation dengan menggunakan sumber daya berikut:

- [AWS: :Log:: DeliverySource](#)
- [AWS: :Log:: DeliveryDestination](#)
- [AWS: :Log: :Pengiriman](#)

Setel ResourceArn ke ruang AWS DevOps agen Agen atau layanan ARN, dan atur LogType ke. APPLICATION_LOGS

Referensi skema log

AWS DevOps Agen menggunakan skema log bersama di semua jenis acara. Tidak setiap peristiwa log menggunakan setiap bidang.

Tabel berikut menjelaskan bidang dalam skema log.

Bidang	Tipe	Deskripsi
event_timestamp	Panjang	Stempel waktu Unix saat peristiwa terjadi
resource_arn	String	ARN dari sumber daya yang menghasilkan acara
optional_account_id	String	AWS ID akun yang terkait dengan log.
opsional_level	String	Tingkat log:INFO,WARN, ERROR

Bidang	Tipe	Deskripsi
optional_agent_space_id	String	Pengidentifikasi ruang agen.
optional_association_id	String	Pengidentifikasi asosiasi untuk log.
optional_status	String	Status operasi topologi.
optional_webhook_id	String	Pengidentifikasi Webhook.
optional_mcp_endpoint_url	String	URL titik akhir server MCP
optional_service_type	String	Jenis Layanan:DYNATRACE ,, DATADOGGITHUB,SLACK,SERVICENOW .
optional_service_endpoint_url	String	URL titik akhir untuk integrasi pihak ketiga.
optional_service_id	String	Pengidentifikasi sumber.
request_id	String	Minta pengenalan untuk berkorelasi dengan AWS CloudTrail atau tiket dukungan.
optional_operation	String	Nama operasi yang dilakukan.
optional_task_type	String	Jenis tugas backlog agen: INVESTIGATION atau EVALUATION
optional_task_id	String	Pengidentifikasi tugas backlog Tugas IDAgent Backlog Agen.
optional_reference	String	Referensi dari tugas agen (misalnya, tiket Jira).
optional_error_type	String	Jenis kesalahan

Bidang	Tipe	Deskripsi
optional_error_message	String	Deskripsi kesalahan saat operasi gagal.
opsional_details	Tali (JSON)	Payload peristiwa khusus layanan yang berisi parameter dan hasil operasi.

Mengelola dan menonaktifkan pengiriman log

Anda dapat mengubah atau menghapus pengiriman log kapan saja dari konsol AWS DevOps Agen di Konsol AWS Manajemen atau dengan menggunakan API CloudWatch Log.

Mengelola pengiriman log (konsol)

1. Buka konsol AWS DevOps Agen di Konsol AWS Manajemen.
2. Arahkan ke halaman Pengaturan (untuk log tingkat layanan) atau halaman Ruang Agen tertentu (untuk log tingkat Ruang Agen).
3. Di tab Konfigurasi (untuk log tingkat Ruang Agen) atau tab Penyedia Kemampuan > Log (untuk log tingkat layanan), pilih pengiriman yang akan dimodifikasi.
4. Perbarui konfigurasi sesuai kebutuhan dan pilih Simpan.

Catatan: Anda tidak dapat mengubah jenis tujuan pengiriman yang ada. Untuk mengubah jenis tujuan, hapus pengiriman saat ini dan buat yang baru.

Nonaktifkan pengiriman log (konsol)

1. Buka konsol AWS DevOps Agen di Konsol AWS Manajemen.
2. Arahkan ke halaman Pengaturan (untuk log tingkat layanan) atau halaman Ruang Agen tertentu (untuk log tingkat Ruang Agen).
3. Di tab Konfigurasi (untuk log tingkat Ruang Agen) atau tab Penyedia Kemampuan > Log (untuk log tingkat layanan), pilih pengiriman yang akan dihapus.
4. Pilih Hapus dan konfirmasi.

Nonaktifkan pengiriman log (API)

Untuk menghapus pengiriman log menggunakan API, hapus sumber daya dalam urutan berikut:

1. Hapus pengiriman dengan menggunakan [DeleteDelivery](#).
2. Hapus sumber pengiriman dengan menggunakan [DeleteDeliverySource](#).
3. (Opsional) Jika tujuan pengiriman tidak lagi diperlukan, hapus dengan menggunakan [DeleteDeliveryDestination](#).

Important

Anda bertanggung jawab untuk menghapus sumber daya pengiriman log setelah menghapus sumber daya ruang agen yang menghasilkan log (misalnya, setelah Anda menghapus ruang agen). Jika Anda tidak menghapus sumber daya ini, konfigurasi pengiriman yatim piatu mungkin tetap ada.

Harga

AWS DevOps Agen tidak mengenakan biaya untuk mengaktifkan log vended. Namun, Anda dapat dikenakan biaya untuk pengiriman, konsumsi, penyimpanan, atau akses, tergantung pada tujuan pengiriman log yang Anda pilih. Untuk detail harga, lihat Log Terjual di tab Log di [CloudWatch Harga Amazon](#).

Untuk harga khusus tujuan, lihat berikut ini:

- [Harga Amazon CloudWatch Log](#)
- [Harga Amazon S3](#)
- [Harga Amazon Data Firehose](#)

Menghubungkan ke alat yang dihosting secara pribadi

Ikhtisar koneksi pribadi

AWS DevOps Agen dapat diperluas dengan alat Model Context Protocol (MCP) kustom dan integrasi lain yang memberikan agen akses ke sistem internal seperti registrasi paket pribadi, platform

observabilitas yang dihosting sendiri, API dokumentasi internal, dan instance kontrol sumber (lihat:). [Mengkonfigurasi kemampuan untuk AWS DevOps Agen](#) Layanan ini sering berjalan di dalam [Amazon Virtual Private Cloud \(Amazon VPC\)](#) dengan akses internet publik terbatas atau tidak ada, yang berarti AWS DevOps Agen tidak dapat menjangkau mereka secara default.

Koneksi pribadi untuk AWS DevOps Agen memungkinkan Anda menghubungkan Ruang Agen Anda dengan aman ke layanan yang berjalan di VPC Anda tanpa memaparkannya ke internet publik. Koneksi pribadi bekerja dengan integrasi apa pun yang perlu mencapai titik akhir pribadi, termasuk server MCP, instans Grafana atau Splunk yang dihosting sendiri, dan sistem kontrol sumber seperti Server Perusahaan dan. GitHub GitLab Self-Managed

Note

Jika alat yang dihosting secara pribadi Anda membuat permintaan keluar ke AWS DevOps Agen dari dalam VPC Anda, lalu lintas ini juga dapat diamankan dengan menggunakan Titik Akhir VPC sehingga tetap berada di dalam jaringan. AWS Misalnya, ini dapat digunakan dengan alat yang memicu DevOps Agen melalui peristiwa webhook (lihat: [the section called “Memanggil DevOps Agen melalui Webhook”](#)). Untuk informasi selengkapnya, lihat [the section called “VPC Endpoint \(AWS PrivateLink\)”](#).

Cara kerja koneksi pribadi

Koneksi pribadi menciptakan jalur jaringan yang aman antara AWS DevOps Agen dan sumber daya target di VPC Anda. Di bawah tenda, AWS DevOps Agen menggunakan Amazon [VPC Lattice](#) untuk membuat jalur konektivitas pribadi yang aman ini. VPC Lattice adalah layanan jaringan aplikasi yang memungkinkan Anda terhubung, mengamankan, dan memantau komunikasi antar aplikasi di seluruh VPC, akun, dan jenis komputasi, tanpa mengelola infrastruktur jaringan yang mendasarinya.

Saat Anda membuat koneksi pribadi, hal berikut terjadi:

- Anda menyediakan VPC, subnet, dan grup keamanan (opsional) yang memiliki konektivitas jaringan ke layanan target Anda.
- AWS DevOps Agen membuat [gateway sumber daya](#) yang dikelola layanan dan menyediakan antarmuka jaringan elastis (ENI) di subnet yang Anda tentukan.
- Agen menggunakan gateway sumber daya untuk mengarahkan lalu lintas ke alamat IP atau nama DNS layanan target Anda melalui jalur jaringan pribadi.

Gateway sumber daya sepenuhnya dikelola oleh AWS DevOps Agen dan muncul sebagai sumber daya hanya-baca di akun Anda (bernama `aidevops-
{your-private-connection-name}`). Anda tidak perlu mengkonfigurasi atau memeliharanya. Satu-satunya sumber daya yang dibuat di VPC Anda adalah ENI di subnet yang Anda tentukan. ENI ini berfungsi sebagai titik masuk untuk lalu lintas pribadi dan dikelola sepenuhnya oleh layanan. Mereka tidak menerima koneksi masuk dari internet, dan Anda mempertahankan kontrol penuh atas lalu lintas mereka melalui grup keamanan Anda sendiri.

Keamanan

Koneksi pribadi dirancang dengan beberapa lapisan keamanan:

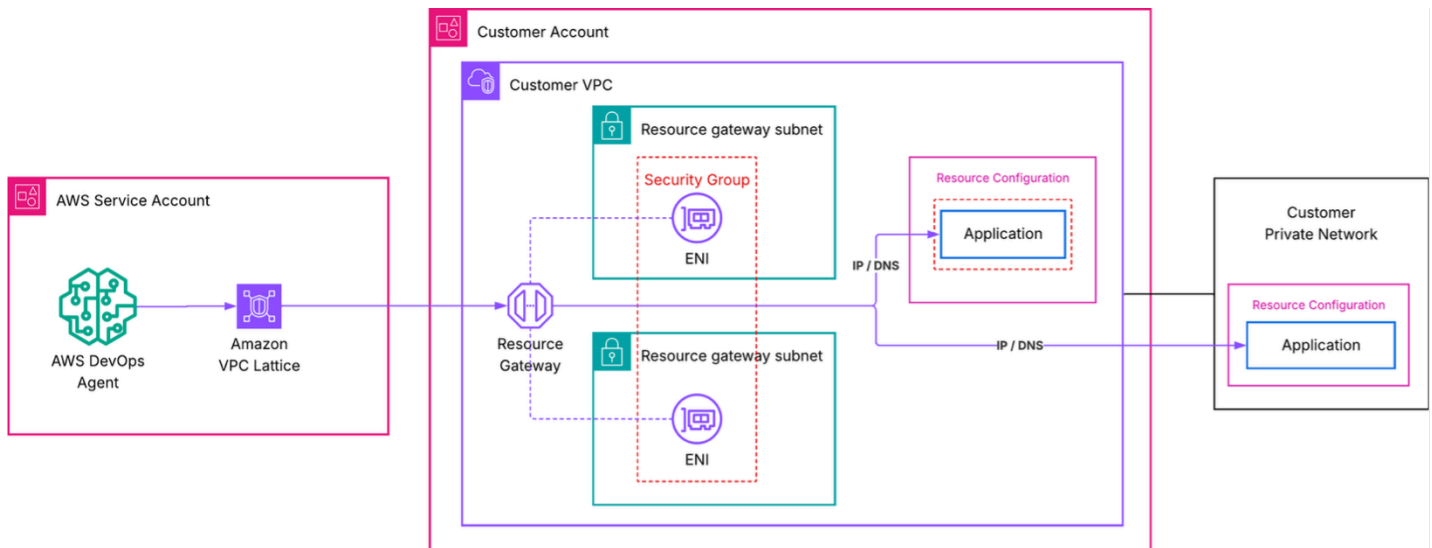
- Tidak ada eksposur internet publik — Semua lalu lintas antara AWS DevOps Agen dan layanan target Anda tetap berada di AWS jaringan. Layanan Anda tidak pernah membutuhkan alamat IP publik atau gateway internet.
- Service-controlled gateway sumber daya — Gateway sumber daya yang dikelola layanan hanya dapat dibaca di akun Anda. Ini hanya dapat digunakan oleh AWS DevOps Agen, dan tidak ada layanan atau prinsipal lain yang dapat mengarahkan lalu lintas melaluinya. Anda dapat memverifikasi ini di [AWS CloudTrail](#) log, yang merekam semua panggilan VPC Lattice API.
- Grup keamanan Anda, aturan Anda — Anda mengontrol lalu lintas masuk dan keluar ke ENI melalui grup keamanan yang Anda miliki dan kelola. Jika Anda tidak menentukan grup keamanan, AWS DevOps Agen akan membuat grup keamanan default yang tercakup ke port yang Anda tentukan.
- Service-linked peran dengan hak istimewa paling sedikit — AWS DevOps Agen menggunakan [peran terkait layanan](#) untuk hanya membuat sumber daya VPC Lattice dan Amazon EC2 yang diperlukan. Peran ini tercakup pada sumber daya yang ditandai `AWSAIDevOpsManaged` dan tidak dapat mengakses sumber daya lain di akun Anda.

Note

Jika organisasi Anda memiliki [kebijakan kontrol layanan \(SCP\) yang membatasi](#) tindakan VPC Lattice API, gateway sumber daya yang dikelola layanan dibuat melalui peran terkait layanan. Pastikan SCP Anda mengizinkan tindakan yang diperlukan untuk peran terkait layanan.

Arsitektur

Diagram berikut menunjukkan jalur jaringan untuk koneksi pribadi.



Dalam arsitektur ini:

- AWS DevOps Agen memulai permintaan ke layanan target Anda.
- Amazon VPC Lattice merutekan permintaan melalui gateway sumber daya yang dikelola layanan di VPC Anda. Untuk pengaturan lanjutan yang menggunakan sumber daya VPC Lattice Anda sendiri, lihat [Penyiapan lanjutan menggunakan sumber daya VPC Lattice yang ada](#).
- ENI di VPC Anda menerima lalu lintas dan meneruskannya ke alamat IP atau nama DNS layanan target Anda.
- Grup keamanan Anda mengatur lalu lintas mana yang diizinkan melalui ENI.
- Dari perspektif layanan target Anda, permintaan berasal dari alamat IP pribadi ENI dalam VPC Anda.

Buat koneksi pribadi

Anda dapat membuat koneksi pribadi menggunakan Konsol AWS Manajemen atau AWS CLI.

Note

Availability Zone berikut tidak didukung oleh VPC Lattice: use1-az3,, usw1-az2,, apne1-az3, apne2-az2 euc1-az2, euw1-az4. cac1-az3 ilc1-az2

Prasyarat

Sebelum membuat koneksi pribadi, verifikasi bahwa Anda memiliki yang berikut:

- Ruang Agen aktif — Anda memerlukan Ruang Agen yang ada di akun Anda. Jika Anda tidak memilikinya, lihat [Memulai dengan AWS DevOps Agen](#).
- Layanan target yang dapat dijangkau secara pribadi — Server MCP Anda, platform observabilitas, atau layanan lainnya harus dapat dijangkau di alamat IP pribadi atau nama DNS yang diketahui dari VPC tempat gateway sumber daya digunakan. Layanan dapat berjalan di VPC yang sama, VPC peered, atau lokal, asalkan dapat dirutekan dari subnet gateway sumber daya. Layanan harus melayani lalu lintas HTTPS dengan versi TLS minimum 1.2 pada port yang Anda tentukan saat membuat koneksi.
- Subnet di VPC Anda — Identifikasi 1-20 subnet tempat ENI akan dibuat. Sebaiknya pilih subnet di beberapa Availability Zone untuk ketersediaan tinggi. Subnet ini harus memiliki konektivitas jaringan ke layanan target Anda. Satu subnet per Availability Zone dapat digunakan oleh VPC Lattice.
- (Opsional) Grup keamanan — Jika Anda ingin mengontrol lalu lintas dengan aturan tertentu, siapkan hingga lima ID grup keamanan untuk dilampirkan ke ENI. Jika Anda menghilangkan grup keamanan, AWS DevOps Agen akan membuat grup keamanan default.

Koneksi pribadi adalah sumber daya tingkat akun. Setelah membuat koneksi pribadi, Anda dapat menggunakannya kembali di beberapa integrasi dan Ruang Agen yang perlu menjangkau host yang sama.

Buat koneksi pribadi menggunakan konsol

1. Buka konsol AWS DevOps Agen.
2. Di panel navigasi, pilih Penyedia kemampuan, lalu pilih Koneksi pribadi.
3. Pilih Buat koneksi baru.
4. Untuk Nama, masukkan nama deskriptif untuk koneksi, seperti `my-mcp-tool-connection`.
5. Untuk VPC, pilih VPC tempat ENI gateway sumber daya akan digunakan.
6. Untuk Subnet, pilih satu atau lebih subnet (hingga 20). Sebaiknya pilih subnet di setidaknya dua Availability Zone.
7. Untuk jenis alamat IP, pilih jenis alamat IP dari layanan target Anda (IPv4, IPv6, atau DualStack).

8. (Opsional) Untuk Jumlah alamat IPv4, jika Anda memilih IPv4 atau Dualstack untuk jenis alamat IP, Anda dapat memasukkan jumlah alamat IPv4 per ENI untuk gateway sumber daya Anda. Defaultnya adalah 16 alamat IPv4 per ENI.
9. (Opsional) Untuk grup Keamanan, pilih grup keamanan yang ada (hingga 5) untuk membatasi lalu lintas apa yang diizinkan untuk mencapai layanan target Anda. Jika Anda tidak memilih salah satu, grup keamanan default akan dibuat.
10. (Opsional) Untuk rentang Port, tentukan port TCP yang didengarkan aplikasi target Anda (misalnya, 443 atau 8080-8090). Anda dapat menentukan hingga 11 rentang port.
11. Untuk alamat Host, masukkan alamat IP atau nama DNS dari layanan target Anda (misalnya, `mcp.internal.example.com` atau `10.0.1.50`). Layanan harus dapat dijangkau dari VPC yang dipilih. Jika Anda memilih nama DNS, itu harus dapat diselesaikan secara publik.
12. (Opsional) Untuk kunci publik Sertifikat, jika alamat host yang Anda tentukan menggunakan sertifikat TLS yang dikeluarkan oleh otoritas sertifikat pribadi, masukkan kunci PEM-encoded publik sertifikat. Hal ini memungkinkan AWS DevOps Agen untuk mempercayai koneksi TLS ke layanan target Anda.
13. Pilih Buat koneksi.

Status koneksi berubah menjadi `Create in progress`. Proses ini bisa memakan waktu hingga 10 menit. Ketika status berubah menjadi `Aktif`, jalur jaringan siap.

Jika status berubah menjadi `Create gagal`, verifikasi hal berikut:

- Subnet yang Anda tentukan memiliki alamat IP yang tersedia.
- Akun Anda belum mencapai kuota layanan VPC Lattice.
- Tidak ada kebijakan IAM yang membatasi yang mencegah peran terkait layanan menciptakan sumber daya.

Note

Langkah-langkah ini juga dapat dilakukan dengan memilih `Create a new private connection` selama pendaftaran penyedia kemampuan. Untuk informasi selengkapnya, lihat [Menggunakan koneksi pribadi dengan penyedia kemampuan](#).

Buat koneksi pribadi menggunakan AWS CLI

Jalankan perintah berikut untuk membuat koneksi pribadi. Ganti nilai placeholder dengan milik Anda sendiri.

```
aws devops-agent create-private-connection \  
  --name my-mcp-tool-connection \  
  --mode '{  
    "serviceManaged": {  
      "hostAddress": "mcp.internal.example.com",  
      "vpcId": "vpc-0123456789abcdef0",  
      "subnetIds": [  
        "subnet-0123456789abcdef0",  
        "subnet-0123456789abcdef1"  
      ],  
      "securityGroupIds": [  
        "sg-0123456789abcdef0"  
      ],  
      "portRanges": ["443"]  
    }  
  }'
```

Tanggapan tersebut mencakup nama koneksi dan status `CREATE_IN_PROGRESS`:

```
{  
  "name": "my-mcp-tool-connection",  
  "status": "CREATE_IN_PROGRESS",  
  "resourceGatewayId": "rgw-0123456789abcdef0",  
  "hostAddress": "mcp.internal.example.com",  
  "vpcId": "vpc-0123456789abcdef0"  
}
```

Untuk memeriksa status koneksi, gunakan `describe-private-connection` perintah:

```
aws devops-agent describe-private-connection \  
  --name my-mcp-tool-connection
```

Ketika statusnya `ACTIVE`, koneksi pribadi Anda siap digunakan.

Gunakan koneksi pribadi dengan penyedia kemampuan

Untuk menggunakan koneksi pribadi, Anda dapat menautkannya selama pendaftaran penyedia kemampuan. Kemampuan yang didukung yang dapat digunakan dengan koneksi pribadi meliputi: GitHub, GitLab, MCP Server, dan Grafana. Anda dapat melakukan langkah ini menggunakan AWS Management Console atau AWS CLI.

Note

Saat mendaftarkan penyedia kemampuan, AWS DevOps Agen memvalidasi bahwa titik akhir dapat dijangkau dan merespons. Pastikan layanan target Anda berjalan dan menerima koneksi sebelum menyelesaikan pendaftaran.

Gunakan koneksi pribadi dengan penyedia kemampuan menggunakan konsol

Di konsol AWS DevOps Agen, koneksi pribadi dapat ditautkan ke kemampuan saat pendaftaran dengan memilih opsi “Connect to endpoint using a private connection”.

MCP server details

Only MCP servers that implement the Streamable HTTP transport protocol are supported.

Name

The name of the MCP server

Endpoint URL

The MCP server endpoint URL will be displayed in AWS CloudTrail logs in your account.

Description - optional

Enable Dynamic Client Registration

Allow DevOps Agent to automatically register with your MCP's authorization server.

Connect to endpoint using a private connection

If not checked, the connection will be made over the public internet.

Use an existing private connection

Select from your existing private connections

Create a new private connection

Create a new VPC connection using Amazon VPC Lattice.

1. Buka konsol AWS DevOps Agen dan arahkan ke Ruang Agen Anda.
2. Di bagian Penyedia Kemampuan, pilih Registrasi.
3. Pilih Daftar untuk jenis kemampuan yang ingin Anda gunakan dengan koneksi pribadi.
4. Pada tampilan detail pendaftaran, masukkan URL Titik Akhir yang ingin Anda sambungkan menggunakan koneksi pribadi (misalnya, `https://mcp.internal.example.com`).
5. Pilih Connect to endpoint menggunakan koneksi pribadi.

6. Pilih koneksi pribadi yang ada yang sesuai dengan URL Endpoint yang ingin Anda sambungkan, atau pilih Buat koneksi pribadi baru untuk membuatnya.
7. Selesaikan proses pendaftaran untuk penyedia kapabilitas.

Note

Saat Anda memilih koneksi pribadi untuk penyedia kemampuan yang menggunakan otentikasi OAuth (Client Credentials atau 3LO), koneksi pribadi berlaku untuk titik akhir penyedia kemampuan dan titik akhir pertukaran token. Pastikan koneksi pribadi dikonfigurasi dengan alamat host yang dapat merutekan lalu lintas ke kedua titik akhir.

Gunakan koneksi pribadi dengan penyedia kemampuan menggunakan AWS CLI

Anda dapat mendaftarkan kemampuan dengan koneksi pribadi dengan memasukkan `private-connection-name` argumen. Di bawah ini adalah contoh mendaftarkan Server MCP dengan otorisasi API Key menggunakan koneksi `my-mcp-tool-connection` pribadi. Ganti nilai placeholder dengan milik Anda sendiri.

```
aws devops-agent register-service \  
  --service mcpserver \  
  --private-connection-name my-mcp-tool-connection \  
  --service-details '{  
    "mcpserver": {  
      "name": "my-mcp-tool",  
      "endpoint": "https://mcp.internal.example.com",  
      "authorizationConfig": {  
        "apiKey": {  
          "apiKeyName": "api-key",  
          "apiKeyValue": "secret-value",  
          "apiKeyHeader": "x-api-key"  
        }  
      }  
    }  
  }' \  
  --region us-east-1
```

Verifikasi koneksi pribadi

Setelah koneksi pribadi mencapai status Aktif dan telah digunakan oleh penyedia kemampuan, verifikasi bahwa AWS DevOps Agen dapat mencapai layanan target Anda:

1. Buka konsol AWS DevOps Agen dan arahkan ke Ruang Agen Anda.
2. Mulai sesi obrolan baru.
3. Memanggil perintah yang menggunakan integrasi yang didukung oleh koneksi pribadi Anda. Misalnya, jika alat MCP Anda menyediakan akses ke basis pengetahuan internal, ajukan pertanyaan kepada agen yang memerlukan basis pengetahuan tersebut.
4. Konfirmasikan bahwa agen mengembalikan hasil dari layanan pribadi.

Jika koneksi gagal, periksa hal berikut:

- [Batas Kisi VPC - Verifikasi bahwa Anda belum mencapai gateway sumber daya atau batas kuota VPC Lattice lainnya](#)
- Aturan grup keamanan — Verifikasi bahwa grup keamanan yang dilampirkan ke ENI memungkinkan lalu lintas keluar pada port yang didengarkan layanan Anda. Juga verifikasi bahwa grup keamanan layanan Anda memungkinkan lalu lintas masuk pada port target. Lalu lintas tiba dari IP pesawat data VPC Lattice dalam jangkauan CIDR VPC Anda. Anda dapat menggunakan referensi grup keamanan (mengizinkan grup keamanan ENI sebagai sumber) atau mengizinkan masuk dari CIDR VPC.
- Konektivitas subnet — Verifikasi bahwa subnet yang Anda pilih dapat merutekan lalu lintas ke layanan Anda. Jika layanan berjalan di subnet yang berbeda, konfirmasikan bahwa tabel rute memungkinkan lalu lintas di antara mereka.
- Ketersediaan layanan — Konfirmasikan bahwa layanan Anda berjalan dan menerima koneksi pada port yang diharapkan.
- Zona Ketersediaan Tidak Didukung - Verifikasi subnet Anda berada di Availability Zone yang didukung. Jalankan `aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` dan periksa Availability Zone yang tidak didukung yang tercantum di atas.

Hapus koneksi pribadi

Anda dapat menghapus koneksi pribadi yang tidak digunakan menggunakan Konsol AWS Manajemen atau AWS CLI.

Hapus koneksi pribadi menggunakan konsol

1. Buka konsol AWS DevOps Agen.
2. Di panel navigasi, pilih Penyedia kemampuan, lalu pilih Koneksi pribadi.
3. Pilih menu Tindakan untuk koneksi pribadi yang ingin Anda hapus, dan pilih Hapus.

Koneksi pribadi akan ditampilkan dengan status “Menghapus koneksi” sementara AWS DevOps Agen menghapus gateway sumber daya terkelola dan ENI dari VPC Anda. Setelah penghapusan selesai, koneksi tidak lagi muncul di daftar koneksi pribadi Anda.

Hapus koneksi pribadi menggunakan AWS CLI

```
aws devops-agent delete-private-connection \  
  --name my-mcp-tool-connection
```

Respons mengembalikan statusDELETE_IN_PROGRESS. AWS DevOps Agen menghapus gateway sumber daya terkelola dan ENI dari VPC Anda. Setelah penghapusan selesai, koneksi tidak lagi muncul di daftar koneksi pribadi Anda.

Pengaturan lanjutan menggunakan sumber daya VPC Lattice yang ada

Jika organisasi Anda sudah menggunakan Amazon VPC Lattice dan mengelola konfigurasi sumber daya Anda sendiri, Anda dapat membuat sambungan pribadi dalam mode yang dikelola sendiri. Alih-alih meminta AWS DevOps Agen membuat gateway sumber daya untuk Anda, Anda memberikan Nama Sumber Daya Amazon (ARN) dari konfigurasi sumber daya yang ada yang mengarah ke layanan target Anda.

Pendekatan ini berguna ketika Anda:

- Ingin kontrol penuh atas gateway sumber daya dan siklus hidup konfigurasi sumber daya.
- Perlu berbagi konfigurasi sumber daya di beberapa AWS akun atau layanan.
- Memerlukan log akses VPC Lattice untuk pemantauan lalu lintas terperinci.

- Jalankan arsitektur jaringan hub-and-spoke.

Untuk membuat koneksi pribadi yang dikelola sendiri dengan AWS CLI:

```
aws devops-agent create-private-connection \  
  --name my-advanced-connection \  
  --mode '{  
    "selfManaged": {  
      "resourceConfigurationId": "arn:aws:vpc-lattice:us-  
east-1:123456789012:resourceconfiguration/rcfg-0123456789abcdef0"  
    }  
  }'
```

Untuk detail selengkapnya tentang menyiapkan gateway sumber daya VPC Lattice dan konfigurasi sumber daya, lihat Panduan Pengguna [Amazon VPC Lattice](#).

Topik terkait

- [the section called “VPC Endpoint \(AWS PrivateLink\)”](#)
- [the section called “Menghubungkan Server MCP”](#)
- [Mengkonfigurasi kemampuan untuk AWS DevOps Agen](#)
- [AWS DevOps Agen Keamanan](#)
- [the section called “DevOps Izin Agen IAM”](#)

AWS DevOps Agen Keamanan

Dokumen ini memberikan informasi tentang pertimbangan keamanan, perlindungan data, kontrol akses, dan kemampuan kepatuhan untuk AWS DevOps Agen. Gunakan informasi ini untuk memahami bagaimana AWS DevOps Agen dirancang untuk memenuhi persyaratan keamanan dan kepatuhan Anda.

Multi-layered keamanan

AWS DevOps Agen mengimplementasikan keamanan di beberapa lapisan. Bahkan jika izin yang lebih luas diberikan untuk peran IAM agen, agen memberlakukan kontrol akses internalnya sendiri untuk membatasi ruang lingkup tindakannya.

Sebaiknya ikuti prinsip hak istimewa paling sedikit saat mengonfigurasi izin IAM untuk AWS DevOps Agen, dan menerapkan keamanan di beberapa lapisan. Pertahanan secara mendalam memastikan bahwa tidak ada kesalahan konfigurasi tunggal yang dapat membahayakan keamanan lingkungan Anda.

Agen Spaces

Agen Spaces berfungsi sebagai batas keamanan utama di AWS DevOps Agen. Setiap Ruang Agen:

- Beroperasi secara independen dengan konfigurasi dan izinnya sendiri
- Menentukan AWS akun dan sumber daya mana yang dapat diakses agen
- Menetapkan koneksi ke platform pihak ketiga

Agen Spaces menjaga isolasi ketat untuk memastikan keamanan dan mencegah akses yang tidak diinginkan di berbagai lingkungan atau tim.

Pemrosesan regional dan aliran data

AWS DevOps Agen beroperasi secara global dengan kemampuan pemrosesan regional. Agen mengambil data operasional dari AWS wilayah di semua AWS akun yang diberikan akses dalam Ruang Agen yang dikonfigurasi. Pengumpulan data lintas akun multi-wilayah ini memastikan analisis insiden yang komprehensif sambil menghormati batas geografis untuk pemrosesan inferensi.

Penggunaan Amazon Bedrock dan inferensi lintas wilayah

AWS DevOps Agen akan secara otomatis memilih wilayah optimal dalam geografi Anda untuk memproses permintaan inferensi Anda. Ini memaksimalkan sumber daya komputasi yang tersedia, ketersediaan model, dan memberikan pengalaman pelanggan terbaik. Data Anda akan tetap disimpan hanya di wilayah tempat Ruang Agen Anda dibuat, namun, permintaan input dan hasil keluaran dapat diproses di luar wilayah tersebut seperti yang dijelaskan dalam daftar berikut. Semua data akan dikirimkan dienkripsi melalui jaringan aman Amazon.

AWS DevOps Agen akan dengan aman merutekan permintaan inferensi Anda ke sumber daya komputasi yang tersedia dalam wilayah geografis tempat permintaan tersebut berasal, sebagai berikut:

- Permintaan inferensi yang berasal dari Uni Eropa akan diproses di dalam Uni Eropa.
- Permintaan inferensi yang berasal dari Amerika Serikat akan diproses di Amerika Serikat.
- Permintaan inferensi yang berasal dari Australia akan diproses di Australia.
- Permintaan inferensi yang berasal dari Jepang akan diproses di Jepang.
- Jika permintaan inferensi berasal dari area yang tidak terdaftar, permintaan tersebut akan diproses secara default di Amerika Serikat.
- DevOps Agen dan Bedrock tidak terpengaruh oleh kebijakan pelanggan dalam Kebijakan Kontrol Layanan (SCP) atau Control Tower yang membatasi konten pelanggan ke wilayah tertentu
- Bedrock dapat menggunakan wilayah selain wilayah asal dalam geografi Anda untuk melakukan inferensi stateless guna mengoptimalkan kinerja dan ketersediaan

Manajemen identitas dan akses

Metode autentikasi

AWS DevOps Agen menyediakan dua metode otentikasi untuk masuk ke aplikasi web AWS DevOps Agent Space:

- AWS Integrasi Pusat Identitas — Metode otentikasi utama menggunakan OAuth 2.0 dengan otentikasi berbasis sesi menggunakan cookie. HTTP-only AWS Identity Center dapat berfederasi dengan penyedia identitas eksternal melalui protokol OIDC dan SAMP standar, termasuk penyedia seperti Okta, Ping Identity, dan Microsoft Entra ID. Metode ini mendukung otentikasi multi-faktor

melalui penyedia identitas Anda. AWS Identity Center default untuk durasi sesi hingga 12 jam dan dapat dikonfigurasi ke durasi yang diinginkan.

- Tautan autentikasi IAM — Metode alternatif menyediakan akses langsung ke aplikasi web dari Konsol AWS Manajemen menggunakan JWT-based token yang berasal dari sesi Konsol AWS Manajemen yang ada. Opsi ini berguna untuk mengevaluasi AWS DevOps Agen sebelum menerapkan integrasi Pusat Identitas penuh serta mendapatkan akses administratif jika aplikasi web AWS DevOps Agen menjadi tidak dapat diakses melalui otentikasi berbasis Identity Center. Sesi dibatasi hingga 10 menit.

Peran IAM

AWS DevOps Agen menggunakan peran IAM untuk menentukan izin akses:

- Peran akun utama — Memberikan agen akses ke sumber daya di AWS akun tempat Anda membuat Ruang Agen serta akses ke peran akun sekunder.
- Peran akun sekunder — Memberikan agen akses ke sumber daya di AWS akun tambahan yang terhubung ke Ruang Agen.
- Peran aplikasi web — Memberi pengguna akses ke data investigasi AWS DevOps Agen dan temuan di aplikasi web.

Peran ini harus dikonfigurasi mengikuti prinsip hak istimewa paling sedikit, hanya memberikan izin baca saja yang diperlukan untuk penyelidikan.

Perlindungan data

Enkripsi data

AWS DevOps Agen mengenkripsi semua data pelanggan:

- Enkripsi saat istirahat - Semua data dienkripsi dengan kunci AWS-managed.
- Enkripsi dalam perjalanan — Semua log yang diambil, metrik, item pengetahuan, metadata tiket, dan data lainnya dienkripsi dalam perjalanan di dalam jaringan pribadi agen dan ke jaringan luar.

Penyimpanan dan retensi data

Data disimpan di wilayah tempat Ruang Agen Anda dibuat, sementara pemrosesan inferensi dapat terjadi dalam geografi Anda seperti yang dijelaskan di bagian penggunaan Amazon Bedrock di atas.

Informasi identitas pribadi (PII)

AWS DevOps Agen tidak memfilter informasi PII saat merangkum data yang dikumpulkan selama investigasi, evaluasi rekomendasi, atau tanggapan obrolan. Disarankan agar data PII disunting sebelum disimpan di log observabilitas.

Jurnal agen dan pencatatan audit

Jurnal agen

Baik kemampuan Investigasi Insiden dan Pencegahan memelihara jurnal terperinci yang:

- Catat setiap langkah penalaran dan tindakan yang diambil
- Ciptakan transparansi lengkap ke dalam proses pengambilan keputusan agen
- Tidak dapat dimodifikasi oleh agen setelah direkam, meminimalkan serangan seperti injeksi cepat dari menyembunyikan tindakan penting
- Sertakan semua pesan obrolan dari halaman Investigasi

AWS CloudTrail integrasi

Semua panggilan AWS DevOps Agent API secara otomatis ditangkap oleh AWS CloudTrail dalam AWS akun hosting. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan:

- Permintaan yang dibuat untuk agen
- Alamat IP dari mana permintaan itu dibuat
- Siapa yang membuat permintaan
- Ketika itu dibuat

Perlindungan injeksi yang cepat

Serangan injeksi cepat terjadi ketika penyerang menyematkan instruksi berbahaya ke dalam data eksternal, seperti halaman web atau dokumen, yang nantinya akan diproses oleh sistem AI generatif. AWS DevOps Agen secara native mengkonsumsi banyak sumber data sebagai bagian dari operasi normalnya, termasuk log, tag sumber daya, dan data operasional lainnya. AWS DevOps Agen melindungi terhadap serangan injeksi cepat melalui pengamanan di bawah ini, tetapi penting untuk memastikan semua sumber data yang terhubung dan akses pengguna ke sumber data tersebut dipercaya. Lihat bagian [Model tanggung jawab bersama](#) untuk informasi selengkapnya.

Pengamanan injeksi yang cepat:

- Kemampuan menulis terbatas — Alat yang tersedia untuk agen tidak dapat mengubah sumber daya, kecuali membuka tiket dan kasus dukungan. Ini mencegah instruksi berbahaya memodifikasi infrastruktur atau aplikasi Anda.
- Penegakan batas akun — AWS DevOps Agen hanya beroperasi dalam batas yang diizinkan oleh peran yang diberikan kepada agen di akun sekunder primer dan terhubung. AWS Agen tidak dapat mengakses atau memodifikasi sumber daya di luar cakupan yang dikonfigurasi.
- Perlindungan keamanan AI - AWS DevOps Agen menggunakan model dengan perlindungan AI Safety Level 3 (ASL-3). Perlindungan ini termasuk pengklasifikasi yang mendeteksi dan mencegah serangan injeksi cepat sebelum mereka dapat mempengaruhi perilaku agen.
- Jejak audit abadi — Jurnal agen mencatat setiap langkah penalaran dan tindakan yang diambil. Entri jurnal tidak dapat dimodifikasi oleh agen setelah direkam, mencegah serangan injeksi cepat menyembunyikan tindakan jahat.

Sementara AWS DevOps Agen menyediakan beberapa lapisan perlindungan terhadap serangan injeksi yang cepat, konfigurasi tertentu dapat meningkatkan risiko:

- Alat server MCP khusus - Fitur MCP yang dibawa sendiri memungkinkan Anda memperkenalkan alat khusus ke agen, yang dapat menghadirkan peluang tambahan untuk injeksi cepat. Alat khusus mungkin tidak memiliki kontrol keamanan yang sama dengan alat AWS DevOps Agen asli, dan instruksi berbahaya berpotensi memanfaatkan alat ini dengan cara yang tidak diinginkan. Lihat bagian [Model tanggung jawab bersama](#) untuk informasi selengkapnya.
- Serangan pengguna resmi — Pengguna yang berwenang untuk beroperasi dalam batas AWS akun atau alat yang terhubung memiliki peluang lebih tinggi untuk mencoba serangan terhadap agen. Pengguna ini mungkin memiliki kemampuan untuk memodifikasi sumber data yang

dikonsumsi agen, seperti log atau tag sumber daya, sehingga lebih mudah untuk menyematkan instruksi berbahaya yang akan diproses agen.

Untuk mengurangi risiko ini:

1. Tinjau dan uji server MCP khusus dengan cermat sebelum menerapkannya di Ruang Agen.
 - a. Pastikan mereka hanya diizinkan untuk melakukan tindakan hanya-baca
 - b. Verifikasi bahwa pengguna alat eksternal yang diakses oleh server MCP adalah entitas tepercaya, karena AWS DevOps Agen yang berinteraksi dengan MCP bergantung pada hubungan kepercayaan implisit yang dibuat antara pengguna alat ini dan Agen AWS DevOps
2. Menerapkan prinsip hak istimewa paling sedikit saat memberikan pengguna akses ke sistem yang menyediakan data kepada agen
3. Secara teratur mengaudit server MCP mana yang terhubung ke Ruang Agen Anda
4. Karena konten apa pun yang diambil dari URL yang diizinkan dapat mencoba memanipulasi perilaku agen, hanya sertakan sumber tepercaya dalam daftar izin Anda.

Keamanan integrasi

AWS DevOps Agen mendukung beberapa jenis integrasi, masing-masing dengan model keamanannya sendiri:

- Integrasi dua arah asli — Built-in integrasi yang dapat mengirim data ke agen dan menerima pembaruan dari agen. Ini menggunakan metode otentikasi vendor
- Server MCP — Server Protokol Konteks Model Jarak Jauh yang memanfaatkan alur otentikasi OAuth 2.0 dan kunci API untuk berkomunikasi secara aman dengan sistem eksternal.
- Pemicu Webhook — Investigasi memicu dari layanan jarak jauh seperti tiket atau sistem observabilitas. Webhook menggunakan Hash-based Message Authentication Code (HMAC) untuk keamanan.
- Komunikasi keluar — Integrasi seperti Slack dan sistem tiket menerima pembaruan dari agen tetapi belum mendukung komunikasi dua arah.

Penyedia pendaftaran

Beberapa alat eksternal diautentikasi di tingkat akun dan dibagikan di antara semua Ruang Agen di akun. Saat Anda mendaftarkan alat ini, Anda mengautentikasi sekali di tingkat akun, dan kemudian setiap Ruang Agen dapat terhubung ke sumber daya tertentu dalam koneksi terdaftar tersebut.

Alat-alat berikut menggunakan pendaftaran tingkat akun:

- **GitHub**— Menggunakan aliran OAuth untuk otentikasi. Setelah mendaftar GitHub di tingkat akun, setiap Ruang Agen dapat terhubung ke repositori tertentu dalam organisasi Anda GitHub .
- **Dynatrace** — Menggunakan otentikasi token OAuth. Setelah mendaftarkan Dynatrace di level akun, setiap Ruang Agen dapat terhubung ke lingkungan Dynatrace tertentu atau konfigurasi pemantauan.
- **Slack** — Menggunakan otentikasi token OAuth. Setelah mendaftarkan Slack di tingkat akun, setiap Ruang Agen dapat terhubung ke saluran Slack tertentu.
- **Datadog** — Menggunakan MCP dengan aliran OAuth untuk otentikasi. Setelah mendaftarkan Datadog di tingkat akun, setiap Ruang Agen dapat terhubung ke sumber daya pemantauan Datadog tertentu.
- **New Relic** — Menggunakan otentikasi kunci API. Setelah mendaftarkan New Relic di level akun, setiap Ruang Agen dapat terhubung ke konfigurasi pemantauan New Relic tertentu.
- **Splunk** — Menggunakan otentikasi token pembawa. Setelah mendaftarkan Splunk di tingkat akun, setiap Ruang Agen dapat terhubung ke sumber data Splunk tertentu.
- **GitLab**— Menggunakan otentikasi token akses. Setelah mendaftar GitLab di tingkat akun, setiap Ruang Agen dapat terhubung ke GitLab repositori tertentu.
- **ServiceNow**— Menggunakan otentikasi klien key/token OAuth. Setelah mendaftar ServiceNow di level akun, setiap Ruang Agen dapat terhubung ke ServiceNow instance atau antrian tiket tertentu.
- **Server MCP jarak jauh yang dapat diakses publik umum** — Gunakan alur OAuth untuk otentikasi. Setelah mendaftarkan server MCP jarak jauh di tingkat akun, setiap Ruang Agen dapat terhubung ke sumber daya tertentu yang diekspos oleh server tersebut.

Konektivitas jaringan

AWS DevOps Agen terhubung ke sistem pihak ketiga Anda dan server MCP jarak jauh untuk melakukan penyelidikan dan operasi lainnya.

Lalu lintas masuk dari AWS DevOps Agen untuk sistem Anda

AWS DevOps Agen memulai koneksi keluar ke sistem pihak ketiga Anda dan server MCP jarak jauh, yang tiba sebagai lalu lintas masuk ke infrastruktur Anda. Cara Anda mengamankan lalu lintas ini tergantung pada bagaimana alat Anda di-host:

- Alat yang dihosting secara pribadi - Jika alat Anda dapat dijangkau dari dalam AWS VPC, Anda dapat menggunakan koneksi pribadi AWS DevOps Agen untuk menjaga lalu lintas terisolasi ke AWS jaringan, dan di luar internet publik. Untuk informasi selengkapnya, lihat [the section called “Menghubungkan ke alat yang dihosting secara pribadi”](#).
- Alat yang dihosting secara publik - Jika alat Anda dapat dijangkau melalui internet publik dan menggunakan aturan IP yang diizinkan atau firewall, Anda harus mengizinkan lalu lintas masuk dari alamat IP sumber Agen berikut: AWS DevOps
 - Asia Pacific (Sydney) (ap-southeast-2)
 - 13.237.95.197
 - 13.238.84.102
 - 52.64.174.242
 - 13.211.249.13
 - 15.134.235.54
 - 3.107.145.226
 - Asia Pacific (Tokyo) (ap-northeast-1)
 - 13.192.12.233
 - 35.74.181.230
 - 57.183.50.158
 - 13.114.228.89
 - 54.150.140.28
 - 46.51.224.121
 - Eropa (Frankfurt) (eu-central-1)
 - 18.158.110.140
 - 52.57.96.160
 - 52.59.55.56
 - 63.183.67.111
 - 63.184.95.132

- 63.184.36.38
- Eropa (Irlandia) (eu-west-1)
 - 34.251.85.24
 - 52.30.157.157
 - 52.51.192.222
 - 99.81.41.52
 - 54.246.170.103
 - 52.212.224.65
- US East (N. Virginia) (us-east-1)
 - 34.228.181.128
 - 44.219.176.187
 - 54.226.244.221
 - 100.56.22.59
 - 3.234.39.4
 - 44.215.92.10
- US West (Oregon) (us-west-2)
 - 34.212.16.133
 - 52.89.67.212
 - 54.187.135.61
 - 34.209.115.89
 - 44.224.219.86
 - 54.201.89.243

Lalu lintas keluar dari VPC Anda ke AWS DevOps Agen

Untuk lalu lintas keluar dari AWS VPC Anda AWS DevOps ke Agen (misalnya, [the section called “Memanggil DevOps Agen melalui Webhook”](#) menggunakan), Anda dapat menggunakan Titik Akhir VPC untuk menjaga lalu lintas jaringan ini terisolasi ke jaringan. AWS Untuk informasi selengkapnya, lihat [the section called “VPC Endpoint \(AWS PrivateLink\)”](#).

Model tanggung jawab bersama

AWS tanggung jawab

AWS bertanggung jawab untuk:

- Menjaga keamanan data yang diambil oleh agen
- Mengamankan alat asli yang tersedia untuk digunakan oleh agen
- Melindungi infrastruktur yang menjalankan AWS DevOps Agen

Tanggung jawab pelanggan

Pelanggan bertanggung jawab untuk:

- Mengelola akses pengguna ke ruang agen
- Membatasi akses ke pengguna tepercaya sistem eksternal yang memberikan masukan kepada agen, seperti layanan dan sumber daya yang menghasilkan log, CloudTrail peristiwa, tiket, dan banyak lagi - yang dapat digunakan untuk mencoba injeksi cepat berbahaya.
- Pastikan semua sumber data yang terhubung memiliki data tepercaya yang tidak mungkin digunakan untuk mencoba serangan injeksi yang cepat
- Memastikan integrasi server MCP yang dibawa sendiri beroperasi dengan aman
- Memastikan peran IAM yang ditugaskan ke agen dicakup dengan benar
- Menyunting data PII sebelum disimpan di log observabilitas dan sumber data agen lainnya
- Mengikuti praktik yang disarankan untuk hanya memberikan izin baca-saja ke sumber data yang terhubung, termasuk membawa server MCP Anda sendiri

Penggunaan data

AWS tidak menggunakan data agen, pesan obrolan, atau data dari sumber data terintegrasi untuk melatih model atau meningkatkan produk. Ruang AWS DevOps Agen menggunakan umpan balik dalam produk pelanggan untuk meningkatkan tanggapan dan investigasi agen, tetapi AWS tidak menggunakannya untuk meningkatkan layanan itu sendiri.

DevOps Izin Agen IAM

AWS DevOps Agen menggunakan tindakan AWS Identity and Access Management (IAM) khusus layanan untuk mengontrol akses ke fitur dan kemampuannya. Tindakan ini menentukan apa yang dapat dilakukan pengguna dalam konsol AWS DevOps Agen dan Aplikasi Web Operator. Ini terpisah dari izin API AWS layanan yang digunakan agen itu sendiri untuk menyelidiki sumber daya Anda.

Untuk informasi selengkapnya tentang membatasi akses agen, lihat [Membatasi Akses Agen di AWS Akun](#).

Tindakan manajemen Ruang Agen

Tindakan ini mengontrol akses ke konfigurasi dan manajemen Ruang Agen:

- `aidevops: GetAgentSpace` — Memungkinkan pengguna untuk melihat detail tentang Ruang Agen, termasuk konfigurasi, status, dan akun terkaitnya. Pengguna memerlukan izin ini untuk mengakses Ruang Agen di Konsol AWS Manajemen.
- `aidevops: GetAssociation` — Memungkinkan pengguna untuk melihat detail tentang asosiasi akun tertentu, termasuk konfigurasi peran IAM dan status koneksi.
- `aidevops: ListAssociations` — Memungkinkan pengguna untuk mencantumkan semua asosiasi AWS akun yang dikonfigurasi untuk Ruang Agen, termasuk akun primer dan sekunder.

Investigasi dan tindakan eksekusi

Tindakan ini mengontrol akses ke fitur investigasi insiden:

- `aidevops: ListExecutions` — Memungkinkan pengguna untuk melihat metadata eksekusi— termasuk ID, status, dan lainnya—untuk investigasi, mitigasi, evaluasi, dan percakapan obrolan yang terkait dengan tugas.
- `aidevops: ListJournalRecords` — Memungkinkan pengguna untuk mengakses log terperinci yang menunjukkan langkah-langkah penalaran agen, tindakan yang diambil, dan sumber data yang dikonsultasikan selama penyelidikan, mitigasi, evaluasi, dan percakapan obrolan. Ini berguna untuk memahami bagaimana agen mencapai kesimpulannya.

Tindakan manajemen obrolan

Obrolan memerlukan izin IAM berikut agar berfungsi:

- `aidevops: ListChats` — Memungkinkan pengguna untuk membuat daftar dan mengakses riwayat percakapan obrolan.
- `aidevops: CreateChat` — Memungkinkan pengguna untuk membuat percakapan obrolan baru.
- `aidevops: SendMessage` — Memungkinkan pengguna untuk mengirimkan pertanyaan dan menerima tanggapan streaming.

Topologi dan tindakan penemuan

Tindakan ini mengontrol akses ke fitur pemetaan sumber daya aplikasi:

- `aidevops: DiscoverTopology` — Memungkinkan pengguna untuk memicu penemuan topologi dan pemetaan untuk Ruang Agen. Tindakan ini memulai proses pemindaian AWS akun dan membangun topologi sumber daya aplikasi.

Tindakan pencegahan dan rekomendasi

Tindakan ini mengontrol akses ke fitur Pencegahan:

- `aidevops: ListGoals` - Memungkinkan pengguna untuk melihat tujuan dan sasaran pencegahan yang sedang dikerjakan agen berdasarkan pola insiden baru-baru ini.
- `aidevops: ListRecommendations` — Memungkinkan pengguna untuk melihat semua rekomendasi yang dihasilkan oleh fitur Pencegahan, termasuk prioritas dan kategori mereka.
- `aidevops: GetRecommendation` — Memungkinkan pengguna untuk melihat informasi terperinci tentang rekomendasi tertentu, termasuk insiden yang akan dicegah dan panduan implementasi.

Tindakan manajemen tugas backlog

Tindakan ini mengontrol kemampuan untuk mengelola rekomendasi sebagai tugas backlog:

- `aidevops: CreateBacklogTask` - Memungkinkan pengguna untuk membuat investigasi insiden atau tugas evaluasi pencegahan.
- `aidevops: UpdateBacklogTask` — Memungkinkan pengguna untuk menyetujui rencana mitigasi atau membatalkan penyelidikan atau evaluasi aktif.
- `aidevops: GetBacklogTask` — Memungkinkan pengguna untuk mengambil detail tentang tugas tertentu.

- `aidevops: ListBacklogTasks` — Memungkinkan pengguna untuk membuat daftar tugas untuk Ruang Agen, difilter berdasarkan jenis tugas, status, prioritas, atau waktu pembuatan.

Tindakan manajemen pengetahuan

Tindakan ini mengontrol kemampuan untuk menambah dan mengelola pengetahuan khusus yang dapat digunakan agen selama investigasi:

- `aidevops: CreateKnowledgeItem` — Memungkinkan pengguna untuk menambahkan item pengetahuan khusus, seperti keterampilan, panduan pemecahan masalah, atau informasi khusus aplikasi yang harus dirujuk agen.
- `aidevops: ListKnowledgeItems` — Memungkinkan pengguna untuk melihat semua item pengetahuan yang dikonfigurasi untuk Ruang Agen.
- `aidevops: GetKnowledgeItem` — Memungkinkan pengguna untuk mengambil detail item pengetahuan tertentu.
- `aidevops: UpdateKnowledgeItem` — Memungkinkan pengguna untuk memodifikasi item pengetahuan yang ada untuk menjaga informasi tetap terkini.
- `aidevops: DeleteKnowledgeItem` — Memungkinkan pengguna untuk menghapus item pengetahuan yang tidak lagi relevan.

AWS Support tindakan integrasi

Tindakan ini mengontrol integrasi dengan kasus AWS Support:

- `aidevops: InitiateChatForCase` — Memungkinkan pengguna untuk memulai sesi obrolan dengan AWS Support langsung dari penyelidikan, secara otomatis memberikan konteks tentang insiden tersebut.
- `aidevops: EndChatForCase` — Memungkinkan pengguna untuk mengakhiri sesi obrolan kasus AWS Support yang aktif.
- `aidevops: DescribeSupportLevel` — Memungkinkan pengguna untuk memeriksa level paket AWS Support untuk akun untuk menentukan opsi dukungan yang tersedia.

Tindakan penggunaan dan pemantauan

Tindakan ini mengontrol akses ke informasi penggunaan:

- `aidevops: GetAccountUsage` — Memungkinkan pengguna untuk melihat kuota bulanan AWS DevOps Agen untuk jam investigasi, jam evaluasi pencegahan, dan permintaan obrolan, serta penggunaan bulan berjalan.

Contoh kebijakan IAM umum

Kebijakan administrator

Kebijakan ini memberikan akses penuh ke semua fitur AWS DevOps Agen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aidevops:*",
      "Resource": "*"
    }
  ]
}
```

Kebijakan operator

Kebijakan ini memberikan akses ke fitur investigasi dan pencegahan tanpa kemampuan administratif:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:InvokeAgent",
        "aidevops:ListExecutions",
        "aidevops:ListJournalRecords",
        "aidevops:ListAssociations",
        "aidevops:GetAssociation",
        "aidevops:DiscoverTopology",
        "aidevops:ListRecommendations",
        "aidevops:GetRecommendation",
        "aidevops>CreateBacklogTask",
      ]
    }
  ]
}
```

```

    "aidevops:UpdateBacklogTask",
    "aidevops:GetBacklogTask",
    "aidevops>ListBacklogTasks",
    "aidevops>ListKnowledgeItems",
    "aidevops:GetKnowledgeItem",
    "aidevops:InitiateChatForCase",
    "aidevops:EndChatForCase",
    "aidevops>ListChats",
    "aidevops>CreateChat",
    "aidevops:SendMessage",
    "aidevops>ListGoals",
    "aidevops>CreateKnowledgeItem",
    "aidevops:UpdateKnowledgeItem",
    "aidevops:DescribeSupportLevel",
    "aidevops>ListPendingMessages"
  ],
  "Resource": "*"
}
]
}

```

Read-only kebijakan

Kebijakan ini memberikan akses hanya lihat ke investigasi dan rekomendasi:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops>ListAssociations",
        "aidevops:GetAssociation",
        "aidevops>ListExecutions",
        "aidevops>ListJournalRecords",
        "aidevops>ListRecommendations",
        "aidevops:GetRecommendation",
        "aidevops>ListBacklogTasks",
        "aidevops:GetBacklogTask",
        "aidevops>ListKnowledgeItems",
        "aidevops:GetKnowledgeItem",
        "aidevops:GetAccountUsage"
      ]
    }
  ]
}

```

```
    ],
    "Resource": "*"
  }
]
```

Menggunakan peran terkait layanan untuk AWS DevOps Agen

AWS DevOps [Agen menggunakan peran AWS terkait layanan Identity and Access Management \(IAM\)](#). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Agen. AWS DevOps Service-linked peran telah ditentukan sebelumnya oleh AWS DevOps Agen dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Service-linked izin peran

Peran terkait layanan `AWSServiceRoleForAIDevOps` mempercayai layanan utama `aidevops.amazonaws.com` untuk memegang peran tersebut.

Peran menggunakan kebijakan terkelola `AWSServiceRoleForAIDevOpsPolicy` dengan izin berikut:

- `cloudwatch:PutMetricData`— Publikasikan metrik penggunaan ke `AWS/AIDevOps` CloudWatch namespace. Dicapuk oleh `cloudwatch:namespace` kondisi untuk hanya mengizinkan namespace. `AWS/AIDevOps`
- `vpc-lattice:CreateResourceGateway`— Buat gateway sumber daya VPC Lattice untuk koneksi pribadi. Dicapuk oleh `aws:RequestTag/AWSAIDevOpsManaged` kondisi sehingga layanan hanya dapat membuat gateway sumber daya yang membawa tag. `AWSAIDevOpsManaged`
- `vpc-lattice:TagResource`— Tag gateway sumber daya Kisi VPC. Tercakup oleh suatu `aws:RequestTag/AWSAIDevOpsManaged` kondisi.
- `vpc-lattice>DeleteResourceGateway`— Hapus gateway sumber daya VPC Lattice. Dicapuk oleh suatu `aws:ResourceTag/AWSAIDevOpsManaged` kondisi sehingga layanan hanya dapat menghapus gateway sumber daya yang dibuatnya.
- `vpc-lattice:GetResourceGateway`— Ambil informasi tentang gateway sumber daya VPC Lattice. Dicapuk oleh suatu `aws:ResourceTag/AWSAIDevOpsManaged` kondisi sehingga layanan hanya dapat membaca gateway sumber daya yang dibuatnya.
- `ec2:DescribeVpcs,ec2:DescribeSubnets,ec2:DescribeSecurityGroups` — Ambil informasi tentang sumber daya jaringan VPC yang diperlukan untuk mengonfigurasi gateway

sumber daya. Tindakan hanya-baca ini berlaku untuk semua sumber daya VPC karena EC2 API tidak mendukung izin tingkat sumber daya untuk panggilan Deskripsikan.

- `iam:CreateServiceLinkedRole`— Buat peran terkait layanan VPC Lattice yang diperlukan untuk operasi gateway sumber daya. Izin ini hanya terbatas pada prinsipal `vpc-lattice.amazonaws.com` layanan dan tidak dapat digunakan untuk membuat peran terkait layanan untuk layanan lainnya.

Membuat peran terkait layanan

Anda tidak perlu membuat peran tertaut layanan `AWSServiceRoleForAIDevOps` secara manual. Saat Anda mulai menggunakan AWS DevOps Agen, layanan akan menciptakan peran terkait layanan untuk Anda.

Untuk memungkinkan layanan membuat peran atas nama Anda, Anda harus memiliki `iam:CreateServiceLinkedRole` izin. Kami merekomendasikan pelingkupan izin ini dengan `iam:AWSServiceName` syarat `aidevops.amazonaws.com` untuk mengikuti prinsip hak istimewa paling sedikit. Untuk informasi selengkapnya, lihat [izin Service-linked peran](#).

Mengedit peran terkait layanan

Anda tidak dapat mengedit peran yang terhubung dengan layanan `AWSServiceRoleForAIDevOps`. Setelah peran dibuat, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran berdasarkan nama. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#).

Menghapus peran tertaut layanan

Jika Anda tidak perlu lagi menggunakan AWS DevOps Agen, kami sarankan Anda menghapus peran `AWSServiceRoleForAIDevOps` terkait layanan. Sebelum Anda dapat menghapus peran, Anda harus terlebih dahulu menghapus koneksi pribadi yang dikonfigurasi di Ruang Agen Anda. Menghapus peran terkait layanan tidak secara otomatis menghapus gateway sumber daya VPC Lattice yang ditandai dengan yang sebelumnya dibuat oleh layanan. `AWSAIDevOpsManaged` Anda harus menghapus gateway sumber daya ini secara manual jika tidak lagi diperlukan. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan](#).

AWS Kebijakan terkelola untuk AWS DevOps Agen

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh. AWS Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk

kasus penggunaan umum sehingga Anda dapat menghindari keharusan menyelidiki izin apa yang diperlukan. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola](#) di `_Panduan Pengguna IAM_`.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk AWS DevOps Agen.

`AIDevOpsAgentReadOnlyAccess`

Menyediakan akses baca saja ke DevOps Agen Amazon melalui Konsol AWS Manajemen

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AIDevOpsAgentReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "aidevops:Get*",
        "aidevops:List*",
        "aidevops:SearchServiceAccessibleResource"
      ],
      "Resource": "*"
    }
  ]
}
```

`AIDevOpsAgentFullAccess`

Menyediakan akses penuh ke Amazon DevOps Agent melalui AWS Management Console

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AIDevOpsAgentSpaceAccess",
      "Effect": "Allow",
      "Action": [
        "aidevops:CreateAgentSpace",
        "aidevops>DeleteAgentSpace",
        "aidevops:GetAgentSpace",
        "aidevops:ListAgentSpaces",
        "aidevops:UpdateAgentSpace"
      ]
    }
  ]
}
```

```
],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsServiceAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:DeregisterService",
    "aidevops:GetService",
    "aidevops:ListServices",
    "aidevops:RegisterService",
    "aidevops:SearchServiceAccessibleResource"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsAssociationAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:AssociateService",
    "aidevops:DisassociateService",
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:UpdateAssociation",
    "aidevops:ValidateAwsAssociations"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsWebhookAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:ListWebhooks"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsOperatorAppAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:DisableOperatorApp",
    "aidevops:EnableOperatorApp",
    "aidevops:GetOperatorApp",
    "aidevops:UpdateOperatorAppIdpConfig"
  ]
}
```

```
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsKnowledgeAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:CreateKnowledgeItem",
    "aidevops>DeleteKnowledgeItem",
    "aidevops:GetKnowledgeItem",
    "aidevops:ListKnowledgeItems",
    "aidevops:ListKnowledgeItemVersions",
    "aidevops:UpdateKnowledgeItem"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsBacklogAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:CreateBacklogTask",
    "aidevops:GetBacklogTask",
    "aidevops:ListBacklogTasks",
    "aidevops:ListGoals",
    "aidevops:UpdateBacklogTask",
    "aidevops:UpdateGoal"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsRecommendationAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:GetRecommendation",
    "aidevops:ListRecommendations",
    "aidevops:UpdateRecommendation"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsAgentChatAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:CreateChat",
```

```
"aidevops:ListChats",
"aidevops:ListPendingMessages",
"aidevops:SendMessage"
],
"Resource": "*"
},
{
  "Sid": "AIDevOpsJournalAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:ListExecutions",
    "aidevops:ListJournalRecords"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsTopologyAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:DiscoverTopology"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsSupportAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:DescribeServices",
    "aidevops:DescribeSupportLevel",
    "aidevops:EndChatForCase",
    "aidevops:InitiateChatForCase"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsUsageAccess",
  "Effect": "Allow",
  "Action": [
    "aidevops:GetAccountUsage"
  ],
  "Resource": "*"
},
{
  "Sid": "AIDevOpsTaggingAccess",
```

```
"Effect": "Allow",
"Action": [
  "aidevops:ListTagsForResource",
  "aidevops:TagResource",
  "aidevops:UntagResource"
],
"Resource": "*"
},
{
  "Sid": "AIDevOpsVendedLogs",
  "Effect": "Allow",
  "Action": [
    "aidevops:AllowVendedLogDeliveryForResource"
  ],
  "Resource": "*"
}
]
```

AIDevOpsOperatorAppAccessPolicy

Menyediakan akses untuk menggunakan aplikasi web AWS DevOps operator untuk Ruang Agen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOperatorAgentSpaceActions",
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:GetAssociation",
        "aidevops:ListAssociations",
        "aidevops:CreateBacklogTask",
        "aidevops:GetBacklogTask",
        "aidevops:UpdateBacklogTask",
        "aidevops:ListBacklogTasks",
        "aidevops:ListJournalRecords",
        "aidevops:DiscoverTopology",
        "aidevops:ListGoals",
        "aidevops:ListRecommendations",
        "aidevops:ListExecutions",
        "aidevops:GetRecommendation",

```

```

    "aidevops:UpdateRecommendation",
    "aidevops:CreateKnowledgeItem",
    "aidevops:ListKnowledgeItems",
    "aidevops:ListKnowledgeItemVersions",
    "aidevops:GetKnowledgeItem",
    "aidevops:UpdateKnowledgeItem",
    "aidevops>DeleteKnowledgeItem",
    "aidevops:ListPendingMessages",
    "aidevops:InitiateChatForCase",
    "aidevops:EndChatForCase",
    "aidevops:DescribeSupportLevel",
    "aidevops:ListChats",
    "aidevops:CreateChat",
    "aidevops:SendMessage",
    "aidevops:DescribeServices"
  ],
  "Resource": "arn:aws:aidevops:*:*:agentspace/${aws:PrincipalTag/AgentSpaceId}",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "AllowOperatorAccountActions",
  "Effect": "Allow",
  "Action": [
    "aidevops:GetAccountUsage"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
},
{
  "Sid": "AllowSupportOperatorActions",
  "Effect": "Allow",
  "Action": [
    "support:DescribeCases",
    "support:DescribeServices",
    "support:InitiateChatForCase",
    "support:DescribeSupportLevel"
  ]
}

```

```

],
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:ResourceAccount": "${aws:PrincipalAccount}"
  }
}
},
{
  "Sid": "AllowSecretsManagerOperatorActions",
  "Effect": "Allow",
  "Action": [
    "secretsmanager:CreateSecret",
    "secretsmanager:ListSecrets"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
  }
}
]
}

```

AIDevOpsAgentAccessPolicy

Memberikan izin yang diperlukan oleh AWS DevOps Agen untuk melakukan investigasi dan melakukan analisis pada sumber daya pelanggan AWS .

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AIOPSServiceAccess",
      "Effect": "Allow",
      "Action": [
        "access-analyzer:GetAnalyzer",
        "access-analyzer:List*",
        "acm-pca:Describe*",
        "acm-pca:GetCertificate",
        "acm-pca:GetCertificateAuthorityCertificate",
        "acm-pca:GetCertificateAuthorityCsr",

```

```
"acm-pca:List*",
"acm:DescribeCertificate",
"acm:GetAccountConfiguration",
"aidevops:GetKnowledgeItem",
"aidevops:ListKnowledgeItems",
"airflow:List*",
"amplify:GetApp",
"amplify:GetBranch",
"amplify:GetDomainAssociation",
"amplify:List*",
"aoss:BatchGetCollection",
"aoss:BatchGetLifecyclePolicy",
"aoss:BatchGetVpcEndpoint",
"aoss:GetAccessPolicy",
"aoss:GetSecurityConfig",
"aoss:GetSecurityPolicy",
"aoss:List*",
"appconfig:GetApplication",
"appconfig:GetConfigurationProfile",
"appconfig:GetEnvironment",
"appconfig:GetHostedConfigurationVersion",
"appconfig:List*",
"appflow:Describe*",
"appflow:List*",
"application-autoscaling:Describe*",
"application-signals:BatchGetServiceLevelObjectiveBudgetReport",
"application-signals:GetService",
"application-signals:GetServiceLevelObjective",
"application-signals:List*",
"applicationinsights:Describe*",
"applicationinsights:List*",
"apprunner:Describe*",
"apprunner:List*",
"appstream:Describe*",
"appstream:List*",
"appsync:GetApiAssociation",
"appsync:GetDataSource",
"appsync:GetDomainName",
"appsync:GetFunction",
"appsync:GetGraphQLApi",
"appsync:GetGraphQLApiEnvironmentVariables",
"appsync:GetIntrospectionSchema",
"appsync:GetResolver",
"appsync:GetSourceApiAssociation",
```

```
"appsync:List*",
"aps:Describe*",
"aps:List*",
"arc-zonal-shift:GetManagedResource",
"arc-zonal-shift:List*",
"athena:GetCapacityAssignmentConfiguration",
"athena:GetCapacityReservation",
"athena:GetDataCatalog",
"athena:GetNamedQuery",
"athena:GetPreparedStatement",
"athena:GetWorkGroup",
"athena:List*",
"auditmanager:GetAssessment",
"auditmanager:List*",
"autoscaling:Describe*",
"backup-gateway:GetHypervisor",
"backup-gateway:List*",
"backup:Describe*",
"backup:GetBackupPlan",
"backup:GetBackupSelection",
"backup:GetBackupVaultAccessPolicy",
"backup:GetBackupVaultNotifications",
"backup:GetRestoreTestingPlan",
"backup:GetRestoreTestingSelection",
"backup:List*",
"batch:DescribeComputeEnvironments",
"batch:DescribeJobQueues",
"batch:DescribeSchedulingPolicies",
"batch:List*",
"bedrock:GetAgent",
"bedrock:GetAgentActionGroup",
"bedrock:GetAgentAlias",
"bedrock:GetAgentKnowledgeBase",
"bedrock:GetDataSource",
"bedrock:GetGuardrail",
"bedrock:GetKnowledgeBase",
"bedrock:List*",
"budgets:Describe*",
"budgets:List*",
"ce:Describe*",
"ce:Get*",
"ce:List*",
"chatbot:Describe*",
"chatbot:GetMicrosoftTeamsChannelConfiguration",
```

```
"chatbot:List*",
"cleanrooms-ml:GetTrainingDataset",
"cleanrooms-ml:List*",
"cleanrooms:GetAnalysisTemplate",
"cleanrooms:GetCollaboration",
"cleanrooms:GetConfiguredTable",
"cleanrooms:GetConfiguredTableAnalysisRule",
"cleanrooms:GetConfiguredTableAssociation",
"cleanrooms:GetMembership",
"cleanrooms:List*",
"cloudformation:Describe*",
"cloudformation:GetResource",
"cloudformation:GetStackPolicy",
"cloudformation:GetTemplate",
"cloudformation:List*",
"cloudfront:Describe*",
"cloudfront:GetCachePolicy",
"cloudfront:GetCloudFrontOriginAccessIdentity",
"cloudfront:GetContinuousDeploymentPolicy",
"cloudfront:GetDistribution",
"cloudfront:GetDistributionConfig",
"cloudfront:GetFunction",
"cloudfront:GetKeyGroup",
"cloudfront:GetMonitoringSubscription",
"cloudfront:GetOriginAccessControl",
"cloudfront:GetOriginRequestPolicy",
"cloudfront:GetPublicKey",
"cloudfront:GetRealtimeLogConfig",
"cloudfront:GetResponseHeadersPolicy",
"cloudfront:List*",
"cloudtrail:Describe*",
"cloudtrail:GetChannel",
"cloudtrail:GetEventConfiguration",
"cloudtrail:GetEventDataStore",
"cloudtrail:GetEventSelectors",
"cloudtrail:GetInsightSelectors",
"cloudtrail:GetQueryResults",
"cloudtrail:GetResourcePolicy",
"cloudtrail:GetTrail",
"cloudtrail:GetTrailStatus",
"cloudtrail:List*",
"cloudtrail:LookupEvents",
"cloudtrail:StartQuery",
"cloudwatch:Describe*",
```

```
"cloudwatch:GenerateQuery",
"cloudwatch:GetDashboard",
"cloudwatch:GetInsightRuleReport",
"cloudwatch:GetMetricData",
"cloudwatch:GetMetricStatistics",
"cloudwatch:GetMetricStream",
"cloudwatch:GetService",
"cloudwatch:GetServiceLevelObjective",
"cloudwatch:List*",
"codeartifact:Describe*",
"codeartifact:GetDomainPermissionsPolicy",
"codeartifact:GetRepositoryPermissionsPolicy",
"codeartifact:List*",
"codebuild:BatchGetFleets",
"codebuild:List*",
"codecommit:GetRepository",
"codecommit:GetRepositoryTriggers",
"codedeploy:BatchGetDeployments",
"codedeploy:BatchGetDeploymentTargets",
"codedeploy:GetApplication",
"codedeploy:GetDeploymentConfig",
"codedeploy:GetDeploymentTarget",
"codedeploy:List*",
"codeguru-profiler:Describe*",
"codeguru-profiler:GetNotificationConfiguration",
"codeguru-profiler:GetPolicy",
"codeguru-profiler:List*",
"codeguru-reviewer:Describe*",
"codeguru-reviewer:List*",
"codepipeline:GetPipeline",
"codepipeline:GetPipelineState",
"codepipeline:List*",
"codestar-connections:GetConnection",
"codestar-connections:GetRepositoryLink",
"codestar-connections:GetSyncConfiguration",
"codestar-connections:List*",
"codestar-notifications:Describe*",
"codestar-notifications:List*",
"cognito-identity:DescribeIdentityPool",
"cognito-identity:GetIdentityPoolRoles",
"cognito-identity:ListIdentityPools",
"cognito-identity:ListTagsForResource",
"cognito-idp:AdminListGroupsForUser",
"cognito-idp:DescribeIdentityProvider",
```

```
"cognito-idp:DescribeResourceServer",
"cognito-idp:DescribeRiskConfiguration",
"cognito-idp:DescribeUserImportJob",
"cognito-idp:DescribeUserPool",
"cognito-idp:DescribeUserPoolDomain",
"cognito-idp:GetGroup",
"cognito-idp:GetLogDeliveryConfiguration",
"cognito-idp:GetUICustomization",
"cognito-idp:GetUserPoolMfaConfig",
"cognito-idp:GetWebACLForResource",
"cognito-idp:ListGroup",
"cognito-idp:ListIdentityProviders",
"cognito-idp:ListResourceServers",
"cognito-idp:ListUserPoolClients",
"cognito-idp:ListUserPools",
"cognito-idp:ListTagsForResource",
"comprehend:Describe*",
"comprehend:List*",
"config:Describe*",
"config:GetStoredQuery",
"config:List*",
"connect:Describe*",
"connect:GetTaskTemplate",
"connect:List*",
"databrew:Describe*",
"databrew:List*",
"datapipeline:Describe*",
"datapipeline:GetPipelineDefinition",
"datapipeline:List*",
"datasync:Describe*",
"datasync:List*",
"deadline:GetFarm",
"deadline:GetFleet",
"deadline:GetLicenseEndpoint",
"deadline:GetMonitor",
"deadline:GetQueue",
"deadline:GetQueueEnvironment",
"deadline:GetQueueFleetAssociation",
"deadline:GetStorageProfile",
"deadline:List*",
"detective:GetMembers",
"detective:List*",
"devicefarm:GetDevicePool",
"devicefarm:GetInstanceProfile",
```

```
"devicefarm:GetNetworkProfile",
"devicefarm:GetProject",
"devicefarm:GetTestGridProject",
"devicefarm:GetVPCEConfiguration",
"devicefarm:List*",
"devops-guru:Describe*",
"devops-guru:GetResourceCollection",
"devops-guru:List*",
"dms:Describe*",
"dms:List*",
"ds:Describe*",
"dynamodb:Describe*",
"dynamodb:GetResourcePolicy",
"dynamodb:List*",
"ec2:Describe*",
"ec2:GetAssociatedEnclaveCertificateIamRoles",
"ec2:GetIpamPoolAllocations",
"ec2:GetIpamPoolCidrs",
"ec2:GetManagedPrefixListEntries",
"ec2:GetNetworkInsightsAccessScopeContent",
"ec2:GetSnapshotBlockPublicAccessState",
"ec2:GetTransitGatewayMulticastDomainAssociations",
"ec2:GetTransitGatewayRouteTableAssociations",
"ec2:GetTransitGatewayRouteTablePropagations",
"ec2:GetVerifiedAccessEndpointPolicy",
"ec2:GetVerifiedAccessGroupPolicy",
"ec2:GetVerifiedAccessInstanceWebAcl",
"ec2:SearchLocalGatewayRoutes",
"ec2:SearchTransitGatewayRoutes",
"ecr:Describe*",
"ecr:GetLifecyclePolicy",
"ecr:GetRegistryPolicy",
"ecr:GetRepositoryPolicy",
"ecr:List*",
"ecs:Describe*",
"ecs:List*",
"eks:AccessKubernetesApi",
"eks:Describe*",
"eks:List*",
"elasticache:Describe*",
"elasticache:List*",
"elasticbeanstalk:Describe*",
"elasticbeanstalk:List*",
"elasticfilesystem:Describe*",
```

```
"elasticloadbalancing:GetResourcePolicy",
"elasticloadbalancing:GetTrustStoreCaCertificatesBundle",
"elasticloadbalancing:GetTrustStoreRevocationContent",
"elasticloadbalancing:Describe*",
"elasticmapreduce:Describe*",
"elasticmapreduce:List*",
"emr-containers:Describe*",
"emr-containers:List*",
"emr-serverless:GetApplication",
"emr-serverless:List*",
"es:Describe*",
"es:List*",
"events:Describe*",
"events:List*",
"evidently:GetExperiment",
"evidently:GetFeature",
"evidently:GetLaunch",
"evidently:GetProject",
"evidently:GetSegment",
"evidently:List*",
"firehose:Describe*",
"firehose:List*",
"fis:GetExperimentTemplate",
"fis:GetTargetAccountConfiguration",
"fis:List*",
"fms:GetNotificationChannel",
"fms:GetPolicy",
"fms:List*",
"forecast:Describe*",
"forecast:List*",
"frauddetector:BatchGetVariable",
"frauddetector:Describe*",
"frauddetector:GetDetectors",
"frauddetector:GetDetectorVersion",
"frauddetector:GetEntityType",
"frauddetector:GetEventTypes",
"frauddetector:GetExternalModels",
"frauddetector:GetLabels",
"frauddetector:GetListElements",
"frauddetector:GetListsMetadata",
"frauddetector:GetModelVersion",
"frauddetector:GetOutcomes",
"frauddetector:GetRules",
"frauddetector:GetVariables",
```

```
"frauddetector:List*",
"fsx:Describe*",
"gamelift:Describe*",
"gamelift:List*",
"globalaccelerator:Describe*",
"globalaccelerator:List*",
"glue:GetDatabase",
"glue:GetDatabases",
"glue:GetJob",
"glue:GetRegistry",
"glue:GetSchema",
"glue:GetSchemaVersion",
"glue:GetTable",
"glue:GetTags",
"glue:GetTrigger",
"glue:List*",
"glue:querySchemaVersionMetadata",
"grafana:Describe*",
"grafana:List*",
"greengrass:Describe*",
"greengrass:GetDeployment",
"greengrass:List*",
"groundstation:GetConfig",
"groundstation:GetDataflowEndpointGroup",
"groundstation:GetMissionProfile",
"groundstation:List*",
"guardduty:GetDetector",
"guardduty:GetFilter",
"guardduty:GetIPSet",
"guardduty:GetMalwareProtectionPlan",
"guardduty:GetMasterAccount",
"guardduty:GetMembers",
"guardduty:GetThreatIntelSet",
"guardduty:List*",
"health:DescribeEvents",
"health:DescribeEventDetails",
"healthlake:Describe*",
"healthlake:List*",
"iam:GetGroup",
"iam:GetGroupPolicy",
"iam:GetInstanceProfile",
"iam:GetLoginProfile",
"iam:GetOpenIDConnectProvider",
"iam:GetPolicy",
```

```
"iam:GetPolicyVersion",
"iam:GetRole",
"iam:GetRolePolicy",
"iam:GetSAMLProvider",
"iam:GetServerCertificate",
"iam:GetServiceLinkedRoleDeletionStatus",
"iam:GetUser",
"iam:GetUserPolicy",
"iam:ListAttachedRolePolicies",
"iam:ListOpenIDConnectProviders",
"iam:ListRolePolicies",
"iam:ListRoles",
"iam:ListServerCertificates",
"iam:ListVirtualMFADevices",
"identitystore:DescribeGroup",
"identitystore:DescribeGroupMembership",
"identitystore:ListGroupMemberships",
"identitystore:ListGroups",
"imagebuilder:GetComponent",
"imagebuilder:GetContainerRecipe",
"imagebuilder:GetDistributionConfiguration",
"imagebuilder:GetImage",
"imagebuilder:GetImagePipeline",
"imagebuilder:GetImageRecipe",
"imagebuilder:GetInfrastructureConfiguration",
"imagebuilder:GetLifecyclePolicy",
"imagebuilder:GetWorkflow",
"imagebuilder:List*",
"inspector2:List*",
"inspector:Describe*",
"inspector:List*",
"internetmonitor:GetMonitor",
"internetmonitor:List*",
"iot:Describe*",
"iot:GetPackage",
"iot:GetPackageVersion",
"iot:GetPolicy",
"iot:GetThingShadow",
"iot:GetTopicRule",
"iot:GetTopicRuleDestination",
"iot:GetV2LoggingOptions",
"iot:List*",
"iotanalytics:Describe*",
"iotanalytics:List*",
```

```
"iotevents:Describe*",
"iotevents:List*",
"iotsitewise:Describe*",
"iotsitewise:List*",
"iotwireless:GetDestination",
"iotwireless:GetDeviceProfile",
"iotwireless:GetFuotaTask",
"iotwireless:GetMulticastGroup",
"iotwireless:GetNetworkAnalyzerConfiguration",
"iotwireless:GetServiceProfile",
"iotwireless:GetWirelessDevice",
"iotwireless:GetWirelessGateway",
"iotwireless:GetWirelessGatewayTaskDefinition",
"iotwireless:List*",
"ivs:GetChannel",
"ivs:GetEncoderConfiguration",
"ivs:GetPlaybackRestrictionPolicy",
"ivs:GetRecordingConfiguration",
"ivs:GetStage",
"ivs:List*",
"ivschat:GetLoggingConfiguration",
"ivschat:GetRoom",
"ivschat:List*",
"kafka:Describe*",
"kafka:GetClusterPolicy",
"kafka:List*",
"kafkaconnect:Describe*",
"kafkaconnect:List*",
"kendra:Describe*",
"kendra:List*",
"kinesis:Describe*",
"kinesis:GetResourcePolicy",
"kinesis:List*",
"kinesisanalytics:Describe*",
"kinesisanalytics:List*",
"kinesisvideo:Describe*",
"kms:DescribeKey",
"kms:ListResourceTags",
"kms:ListKeys",
"kms:GetKeyPolicy",
"kms:GetKeyRotationStatus",
"kms:ListAliases",
"kms:ListKeyRotations",
"lakeformation:Describe*",
```

```
"lakeformation:GetLFTag",
"lakeformation:GetResourceLFTags",
"lakeformation:List*",
"lambda:GetAlias",
"lambda:GetCodeSigningConfig",
"lambda:GetEventSourceMapping",
"lambda:GetFunctionCodeSigningConfig",
"lambda:GetFunctionConfiguration",
"lambda:GetFunctionEventInvokeConfig",
"lambda:GetFunctionRecursionConfig",
"lambda:GetFunctionUrlConfig",
"lambda:GetLayerVersion",
"lambda:GetLayerVersionPolicy",
"lambda:GetPolicy",
"lambda:GetProvisionedConcurrencyConfig",
"lambda:GetRuntimeManagementConfig",
"lambda:List*",
"launchwizard:GetDeployment",
"launchwizard:List*",
"license-manager:GetLicense",
"license-manager:List*",
"lightsail:GetAlarms",
"lightsail:GetBuckets",
"lightsail:GetCertificates",
"lightsail:GetContainerServices",
"lightsail:GetDisk",
"lightsail:GetDisks",
"lightsail:GetInstance",
"lightsail:GetInstances",
"lightsail:GetLoadBalancer",
"lightsail:GetLoadBalancers",
"lightsail:GetLoadBalancerTlsCertificates",
"lightsail:GetStaticIp",
"lightsail:GetStaticIps",
"logs:Describe*",
"logs:FilterLogEvents",
"logs:GetDataProtectionPolicy",
"logs:GetDelivery",
"logs:GetDeliveryDestination",
"logs:GetDeliveryDestinationPolicy",
"logs:GetDeliverySource",
"logs:GetLogAnomalyDetector",
"logs:GetLogDelivery",
"logs:GetLogGroupFields",
```

```
"logs:GetQueryResults",
"logs:List*",
"logs:StartQuery",
"logs:StopLiveTail",
"logs:StopQuery",
"logs:TestMetricFilter",
"m2:GetApplication",
"m2:GetEnvironment",
"m2:List*",
"macie2:GetAllowList",
"macie2:GetCustomDataIdentifier",
"macie2:GetFindingsFilter",
"macie2:GetMacieSession",
"macie2:List*",
"mediaconnect:Describe*",
"mediaconnect:List*",
"medialive:Describe*",
"medialive:GetCloudWatchAlarmTemplate",
"medialive:GetCloudWatchAlarmTemplateGroup",
"medialive:GetEventBridgeRuleTemplate",
"medialive:GetEventBridgeRuleTemplateGroup",
"medialive:GetSignalMap",
"medialive:List*",
"mediapackage-vod:Describe*",
"mediapackage-vod:List*",
"mediapackage:Describe*",
"mediapackage:List*",
"mediapackagev2:GetChannel",
"mediapackagev2:GetChannelGroup",
"mediapackagev2:GetChannelPolicy",
"mediapackagev2:GetOriginEndpoint",
"mediapackagev2:GetOriginEndpointPolicy",
"mediapackagev2:List*",
"memorydb:Describe*",
"memorydb:List*",
"mobiletargeting:GetInAppTemplate",
"mobiletargeting:List*",
"mq:Describe*",
"mq:List*",
"network-firewall:Describe*",
"network-firewall:List*",
"networkmanager:Describe*",
"networkmanager:GetConnectAttachment",
"networkmanager:GetConnectPeer",
```

```
"networkmanager:GetCoreNetwork",
"networkmanager:GetCoreNetworkPolicy",
"networkmanager:GetCustomerGatewayAssociations",
"networkmanager:GetDevices",
"networkmanager:GetLinkAssociations",
"networkmanager:GetLinks",
"networkmanager:GetSites",
"networkmanager:GetSiteToSiteVpnAttachment",
"networkmanager:GetTransitGatewayPeering",
"networkmanager:GetTransitGatewayRegistrations",
"networkmanager:GetTransitGatewayRouteTableAttachment",
"networkmanager:GetVpcAttachment",
"networkmanager:List*",
"oam:GetLink",
"oam:GetSink",
"oam:GetSinkPolicy",
"oam:List*",
"omics:GetAnnotationStore",
"omics:GetReferenceStore",
"omics:GetRunGroup",
"omics:GetSequenceStore",
"omics:GetVariantStore",
"omics:GetWorkflow",
"omics:List*",
"organizations:Describe*",
"organizations:List*",
"osis:GetPipeline",
"osis:List*",
"payment-cryptography:GetAlias",
"payment-cryptography:GetKey",
"payment-cryptography:List*",
"pca-connector-ad:GetConnector",
"pca-connector-ad:GetDirectoryRegistration",
"pca-connector-ad:GetServicePrincipalName",
"pca-connector-ad:GetTemplate",
"pca-connector-ad:GetTemplateGroupAccessControlEntry",
"pca-connector-ad:List*",
"pca-connector-scep:GetChallengeMetadata",
"pca-connector-scep:GetConnector",
"pca-connector-scep:List*",
"personalize:Describe*",
"personalize:List*",
"pi:Describe*",
"pi:Get*",
```

```
"pi:List*",
"pipes:Describe*",
"pipes:List*",
"proton:GetEnvironmentTemplate",
"proton:GetServiceTemplate",
"proton:List*",
"qbusiness:GetApplication",
"qbusiness:GetDataSource",
"qbusiness:GetIndex",
"qbusiness:GetPlugin",
"qbusiness:GetRetriever",
"qbusiness:GetWebExperience",
"qbusiness:List*",
"ram:GetPermission",
"ram:GetResourceShares",
"ram:List*",
"rds:Describe*",
"rds:List*",
"redshift-serverless:GetNamespace",
"redshift-serverless:GetWorkgroup",
"redshift-serverless:List*",
"redshift:Describe*",
"refactor-spaces:GetApplication",
"refactor-spaces:GetEnvironment",
"refactor-spaces:GetRoute",
"refactor-spaces:List*",
"rekognition:Describe*",
"rekognition:List*",
"resiliencehub:Describe*",
"resiliencehub:List*",
"resource-explorer-2:GetDefaultView",
"resource-explorer-2:GetIndex",
"resource-explorer-2:GetView",
"resource-explorer-2:List*",
"resource-explorer-2:Search",
"resource-groups:GetGroup",
"resource-groups:GetGroupConfiguration",
"resource-groups:GetGroupQuery",
"resource-groups:GetTags",
"resource-groups:List*",
"route53-recovery-control-config:Describe*",
"route53-recovery-control-config:List*",
"route53-recovery-readiness:GetCell",
"route53-recovery-readiness:GetReadinessCheck",
```

```
"route53-recovery-readiness:GetRecoveryGroup",
"route53-recovery-readiness:GetResourceSet",
"route53-recovery-readiness:List*",
"route53:GetDNSSEC",
"route53:GetHealthCheck",
"route53:GetHealthCheckStatus",
"route53:GetHostedZone",
"route53:List*",
"route53profiles:GetProfile",
"route53profiles:GetProfileAssociation",
"route53profiles:GetProfileResourceAssociation",
"route53profiles:List*",
"route53resolver:GetFirewallDomainList",
"route53resolver:GetFirewallRuleGroup",
"route53resolver:GetFirewallRuleGroupAssociation",
"route53resolver:GetOutpostResolver",
"route53resolver:GetResolverConfig",
"route53resolver:GetResolverQueryLogConfig",
"route53resolver:GetResolverQueryLogConfigAssociation",
"route53resolver:GetResolverRule",
"route53resolver:GetResolverRuleAssociation",
"route53resolver:List*",
"rum:GetAppMonitor",
"rum:List*",
"s3-outposts:ListEndpoints",
"s3-outposts:ListOutpostsWithS3",
"s3:GetAccessGrant",
"s3:GetAccessGrantsInstance",
"s3:GetAccessGrantsLocation",
"s3:GetAccessPoint",
"s3:GetAccessPointConfigurationForObjectLambda",
"s3:GetAccessPointForObjectLambda",
"s3:GetAccessPointPolicy",
"s3:GetAccessPointPolicyForObjectLambda",
"s3:GetAccessPointPolicyStatusForObjectLambda",
"s3:GetBucketAbac",
"s3:GetBucketAcl",
"s3:GetBucketCORS",
"s3:GetBucketLocation",
"s3:GetBucketLogging",
"s3:GetBucketMetadataTableConfiguration",
"s3:GetBucketNotification",
"s3:GetBucketObjectLockConfiguration",
"s3:GetBucketOwnershipControls",
```

```
"s3:GetBucketPolicy",
"s3:GetBucketPublicAccessBlock",
"s3:GetBucketTagging",
"s3:GetBucketVersioning",
"s3:GetEncryptionConfiguration",
"s3:GetLifecycleConfiguration",
"s3:GetMultiRegionAccessPoint",
"s3:GetMultiRegionAccessPointPolicy",
"s3:GetMultiRegionAccessPointPolicyStatus",
"s3:GetReplicationConfiguration",
"s3:GetStorageLensConfiguration",
"s3:GetStorageLensConfigurationTagging",
"s3:GetStorageLensGroup",
"s3:ListAllMyBuckets",
"sagemaker:Describe*",
"sagemaker:List*",
"scheduler:GetSchedule",
"scheduler:GetScheduleGroup",
"scheduler:List*",
"schemas:Describe*",
"schemas:GetResourcePolicy",
"schemas:List*",
"secretsmanager:Describe*",
"secretsmanager:GetResourcePolicy",
"secretsmanager:List*",
"securityhub:BatchGetAutomationRules",
"securityhub:BatchGetSecurityControls",
"securityhub:Describe*",
"securityhub:GetConfigurationPolicy",
"securityhub:GetConfigurationPolicyAssociation",
"securityhub:GetEnabledStandards",
"securityhub:GetFindingAggregator",
"securityhub:GetInsights",
"securityhub:List*",
"securitylake:GetSubscriber",
"securitylake:List*",
"servicecatalog:Describe*",
"servicecatalog:GetApplication",
"servicecatalog:GetAttributeGroup",
"servicecatalog:List*",
"servicequotas:GetServiceQuota",
"servicequotas:ListServiceQuotas",
"ses:Describe*",
"ses:GetAccount",
```

```
"ses:GetAddonInstance",
"ses:GetAddonSubscription",
"ses:GetArchive",
"ses:GetConfigurationSet",
"ses:GetConfigurationSetEventDestinations",
"ses:GetContactList",
"ses:GetDedicatedIpPool",
"ses:GetDedicatedIps",
"ses:GetEmailIdentity",
"ses:GetEmailTemplate",
"ses:GetIngressPoint",
"ses:GetRelay",
"ses:GetRuleSet",
"ses:GetTemplate",
"ses:GetTrafficPolicy",
"ses:List*",
"shield:Describe*",
"shield:List*",
"signer:GetSigningProfile",
"signer:List*",
"sns:GetDataProtectionPolicy",
"sns:GetSubscriptionAttributes",
"sns:GetTopicAttributes",
"sns:List*",
"sqs:GetQueueAttributes",
"sqs:GetQueueUrl",
"sqs:List*",
"ssm-contacts:GetContact",
"ssm-contacts:GetContactChannel",
"ssm-contacts:List*",
"ssm-incidents:GetReplicationSet",
"ssm-incidents:GetResponsePlan",
"ssm-incidents:List*",
"ssm-sap:GetApplication",
"ssm-sap:List*",
"ssm:Describe*",
"ssm:GetDefaultPatchBaseline",
"ssm:GetDocument",
"ssm:GetParameters",
"ssm:GetPatchBaseline",
"ssm:GetResourcePolicies",
"ssm:List*",
"sso:GetInlinePolicyForPermissionSet",
"sso:GetManagedApplicationInstance",
```

```
"sso:GetPermissionsBoundaryForPermissionSet",
"sso:GetSharedSsoConfiguration",
"sso:ListAccountAssignments",
"sso:ListApplicationAssignments",
"sso:ListApplications",
"sso:ListCustomerManagedPolicyReferencesInPermissionSet",
"sso:ListInstances",
"sso:ListManagedPoliciesInPermissionSet",
"sso:ListTagsForResource",
"states:GetExecutionHistory",
"states:Describe*",
"states:List*",
"support:CreateCase",
"support:DescribeCases",
"synthetics:Describe*",
"synthetics:GetCanary",
"synthetics:GetCanaryRuns",
"synthetics:GetGroup",
"synthetics:List*",
"tag:GetResources",
"timestream:Describe*",
"timestream:List*",
"transfer:Describe*",
"transfer:List*",
"verifiedpermissions:GetIdentitySource",
"verifiedpermissions:GetPolicy",
"verifiedpermissions:GetPolicyStore",
"verifiedpermissions:GetPolicyTemplate",
"verifiedpermissions:GetSchema",
"verifiedpermissions:List*",
"vpc-lattice:GetAccessLogSubscription",
"vpc-lattice:GetAuthPolicy",
"vpc-lattice:GetListener",
"vpc-lattice:GetResourcePolicy",
"vpc-lattice:GetRule",
"vpc-lattice:GetService",
"vpc-lattice:GetServiceNetwork",
"vpc-lattice:GetServiceNetworkServiceAssociation",
"vpc-lattice:GetServiceNetworkVpcAssociation",
"vpc-lattice:GetTargetGroup",
"vpc-lattice:List*",
"wafv2:GetIPSet",
"wafv2:GetLoggingConfiguration",
"wafv2:GetRegexPatternSet",
```

```

        "wafv2:GetRuleGroup",
        "wafv2:GetWebACL",
        "wafv2:GetWebACLForResource",
        "wafv2:List*",
        "workspaces-web:GetBrowserSettings",
        "workspaces-web:GetIdentityProvider",
        "workspaces-web:GetNetworkSettings",
        "workspaces-web:GetPortal",
        "workspaces-web:GetPortalServiceProviderMetadata",
        "workspaces-web:GetTrustStore",
        "workspaces-web:GetUserAccessLoggingSettings",
        "workspaces-web:GetUserSettings",
        "workspaces-web:List*",
        "workspaces:Describe*",
        "xray:BatchGetTraces",
        "xray:GetGroup",
        "xray:GetGroups",
        "xray:GetSamplingRules",
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:List*"
    ],
    "Resource": "*"
},
{
    "Sid": "AIOPSAPIGatewayAccess",
    "Effect": "Allow",
    "Action": [
        "apigateway:GET"
    ],
    "Resource": [
        "arn:aws:apigateway:*::/restapis",
        "arn:aws:apigateway:*::/restapis/*",
        "arn:aws:apigateway:*::/restapis/*/deployments",
        "arn:aws:apigateway:*::/restapis/*/deployments/*",
        "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations",
        "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations/
    *",
        "arn:aws:apigateway:*::/restapis/*/stages",
        "arn:aws:apigateway:*::/restapis/*/stages/*",
        "arn:aws:apigateway:*::/apis",
        "arn:aws:apigateway:*::/apis/*",
        "arn:aws:apigateway:*::/apis/*/deployments",
        "arn:aws:apigateway:*::/apis/*/deployments/*",

```

```
        "arn:aws:apigateway:*::/apis/*/integrations",
        "arn:aws:apigateway:*::/apis/*/integrations/*",
        "arn:aws:apigateway:*::/apis/*/stages",
        "arn:aws:apigateway:*::/apis/*/stages/*",
        "arn:aws:apigateway:*::/domainnames/*"
    ]
}
]
```

Membatasi Akses Agen dalam AWS Akun

AWS DevOps Agen menggunakan peran IAM untuk menemukan dan mendeskripsikan AWS sumber daya selama investigasi insiden dan evaluasi pencegahan. Anda dapat mengontrol tingkat akses yang dimiliki agen dengan mengonfigurasi kebijakan IAM yang dilampirkan pada peran ini. Topologi aplikasi tidak menunjukkan semua yang dapat diakses agen — kebijakan IAM adalah satu-satunya cara untuk benar-benar membatasi API AWS layanan dan sumber daya apa yang dapat diakses agen.

Memahami peran IAM untuk AWS DevOps Agen

AWS DevOps Agen menggunakan peran IAM untuk mengakses sumber daya dalam dua jenis akun:

- Peran akun utama — Memberikan agen akses ke sumber daya di AWS akun tempat Anda membuat Ruang Agen.
- Peran akun sekunder — Memberikan agen akses ke sumber daya di AWS akun tambahan yang Anda sambungkan ke Ruang Agen.

Untuk kedua jenis akun, Anda dapat membatasi AWS layanan mana yang dapat diakses agen, membatasi akses ke sumber daya tertentu dalam layanan tersebut, dan mengontrol wilayah mana agen dapat beroperasi.

Memahami pagar pembatas izin

AWS DevOps Agen menerapkan pagar pembatas izin untuk setiap sesi yang dibuatnya saat mengakses sumber daya Anda. AWS Pagar pembatas ini bertindak sebagai plafon — ini menentukan set maksimum izin yang dapat digunakan agen, terlepas dari izin apa yang Anda berikan pada peran IAM.

Cara kerjanya

Ketika agen mengasumsikan peran IAM Anda, ia melewati [kebijakan sesi](#) yang membatasi izin efektif untuk sesi tersebut. Izin yang efektif adalah persimpangan dari:

1. Kebijakan peran IAM Anda — Kebijakan terkelola dan kebijakan inline apa pun yang Anda lampirkan ke peran tersebut.
2. Pagar pembatas izin — Kebijakan sesi yang diterapkan oleh AWS DevOps Agen pada waktu peran tertentu.

Izin harus ada di kedua lapisan untuk diterapkan. Jika Anda menambahkan izin ke peran Anda yang tidak termasuk dalam pagar pembatas, agen tidak dapat menggunakannya.

Izin default

Kebijakan `AIDevOpsAgentAccessPolicy` terkelola menyediakan set default izin hanya-baca yang digunakan agen untuk investigasi. Izin ini termasuk dalam pagar pembatas, sehingga berfungsi tanpa konfigurasi tambahan.

Memperluas izin di luar default

AWS DevOps Agen mendukung serangkaian izin tambahan yang dikurasi di luar kebijakan terkelola default. Izin ini disertakan dalam pagar pembatas tetapi tidak diaktifkan secara default. Untuk menggunakannya, tambahkan izin khusus ke peran Anda sebagai kebijakan sebaris.

Misalnya, untuk mengizinkan agen membaca objek dari bucket S3 Anda selama penyelidikan, tambahkan kebijakan inline ke peran Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-application-bucket",
        "arn:aws:s3:::my-application-bucket/*"
      ]
    }
  ]
}
```

```

    ]
  }
]
}
```

Karena `s3:GetObject` dan `s3:ListBucket` termasuk dalam pagar pembatas, kebijakan inline ini berlaku. Anda dapat memasukkan bucket Resource ke spesifik untuk mengikuti prinsip hak istimewa paling sedikit.

Izin tambahan yang didukung

Izin berikut disertakan dalam pagar pembatas dan dapat diaktifkan dengan menambahkannya ke peran Anda sebagai kebijakan sebaris. Ini tidak diberikan secara default - Anda harus secara eksplisit ikut serta.

Layanan	Tindakan	Kasus penggunaan
Amazon S3	<code>s3:GetObject</code> , <code>s3:ListBucket</code>	Baca data aplikasi, log, atau konfigurasi yang disimpan di S3
AWS Connect Langsung	<code>directconnect:DescribeConnections</code> , <code>directconnect:DescribeDirectConnectGatewayAssociations</code> , <code>directconnect:DescribeDirectConnectGateways</code> , <code>directconnect:DescribeLags</code> , <code>directconnect:DescribeVirtualInterfaces</code>	Selidiki masalah konektivitas jaringan

Catatan: Daftar ini dapat berkembang seiring waktu karena kemampuan baru ditambahkan ke AWS DevOps Agen. Izin yang tidak tercantum di sini atau dalam kebijakan `AIDevOpsAgentAccessPolicy` terkelola diblokir oleh pagar pembatas.

Izin diblokir oleh pagar pembatas

Jika Anda menambahkan izin ke peran Anda yang tidak ada di pagar pembatas, agen tidak dapat menggunakannya. Ini dirancang — pagar pembatas mencegah agen melakukan tindakan di luar ruang lingkup yang dimaksudkan, bahkan jika peran tersebut memungkinkan mereka.

Misalnya, menulis operasi seperti `s3:PutObject`, `ec2:TerminateInstances`, atau `dynamodb>DeleteItem` tidak termasuk dalam pagar pembatas. Bahkan jika peran Anda memberikan izin ini, agen tidak dapat melakukan tindakan ini.

Ringkasan

Lapisan	Siapa yang mengendalikannya	Tujuan
Kebijakan peran IAM	Anda	Tentukan apa yang Anda ingin agen untuk dapat lakukan
Izin pagar pembatas	AWS DevOps Agen	Mendefinisikan maksimum yang bisa dilakukan agen
Izin yang efektif	Persimpangan keduanya	Apa yang sebenarnya bisa dilakukan agen

Model ini memastikan bahwa agen beroperasi dalam batas keamanan yang terdefinisi dengan baik sambil memberi Anda fleksibilitas untuk memperluas kemampuannya untuk kasus penggunaan spesifik Anda.

Memilih batas sumber daya Anda

Saat membatasi akses sumber daya, Anda perlu menyertakan izin yang cukup agar agen berhasil menyelidiki insiden aplikasi. Hal ini mencakup:

- Semua sumber daya untuk aplikasi dalam ruang lingkup yang harus dipantau dan diselidiki oleh agen
- Semua infrastruktur pendukung yang bergantung pada aplikasi tersebut

Infrastruktur pendukung dapat mencakup:

- Komponen jaringan (VPC, subnet, penyeimbang beban, gateway API)
- Penyimpanan data (database, cache, penyimpanan objek)
- Menghitung sumber daya (instans EC2, fungsi Lambda, kontainer)
- Layanan pemantauan dan pencatatan (CloudWatch, CloudTrail)
- Sumber daya identitas dan manajemen akses yang diperlukan untuk memahami izin

Jika Anda membatasi akses terlalu sempit, agen mungkin tidak dapat mengidentifikasi akar penyebab yang berasal dari infrastruktur pendukung di luar batas yang Anda tetapkan.

Membatasi akses layanan

Anda dapat membatasi AWS layanan mana yang dapat diakses agen dengan memodifikasi kebijakan IAM yang melekat pada peran agen. Saat membuat kebijakan khusus, ikuti praktik terbaik berikut:

- Berikan izin hanya-baca saja — Agen perlu membaca konfigurasi sumber daya, metrik, dan log selama penyelidikan. Hindari pemberian izin yang memungkinkan agen untuk memodifikasi atau menghapus sumber daya.
- Batasi layanan yang diperlukan — Sertakan hanya AWS layanan yang berisi sumber daya yang relevan dengan aplikasi Anda. Misalnya, jika aplikasi Anda tidak menggunakan Amazon RDS, jangan sertakan izin RDS dalam kebijakan.
- Gunakan tindakan spesifik alih-alih wildcard — Alih-alih memberikan `service:*` izin, tentukan tindakan individual seperti `atau. cloudwatch: GetMetricData ec2: DescribeInstances`

Contoh kebijakan yang membatasi layanan tertentu:

```
json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch: GetMetricData",
        "cloudwatch: GetMetricStatistics",
        "cloudwatch: DescribeAlarms",
```

```
    "logs:GetLogEvents",
    "logs:FilterLogEvents",
    "ec2:DescribeInstances",
    "lambda:GetFunction",
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": "*"
}
]
```

Membatasi akses sumber daya

Untuk membatasi agen ke sumber daya tertentu dalam layanan, gunakan izin tingkat sumber daya dalam kebijakan IAM Anda. Ini memungkinkan Anda memberikan akses hanya ke sumber daya yang cocok dengan pola tertentu.

Menggunakan pola ARN sumber daya:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

Contoh ini membatasi agen untuk hanya mengakses fungsi Lambda dengan nama yang dimulai dengan “produksi-”.

Menggunakan pembatasan berbasis tag:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeInstanceStatus"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/Environment": "production"
    }
  }
}
```

Contoh ini membatasi agen untuk hanya mengakses instans EC2 yang ditandai. Environment=production

Membatasi akses regional

Untuk membatasi AWS wilayah mana yang dapat diakses agen, gunakan kunci `aws:RequestedRegion` kondisi dalam kebijakan IAM Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "lambda:Get*",
        "cloudwatch:Get*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-west-2"
          ]
        }
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Contoh ini membatasi agen untuk mengakses sumber daya hanya di wilayah us-east-1 dan us-west-2.

Membuat kebijakan IAM khusus

Saat membuat Ruang Agen atau menambahkan akun sekunder, Anda memiliki opsi untuk membuat peran IAM kustom menggunakan templat kebijakan. Ini memungkinkan Anda untuk menerapkan prinsip hak istimewa paling sedikit.

Saat membuat Ruang Agen

Dari konsol DevOps Agen di Konsol AWS Manajemen...

- Pilih Buat peran DevOps Agen baru menggunakan dokumen kebijakan dan ikuti petunjuknya

Saat mengedit Ruang Agen

Dari konsol DevOps Agen di Konsol AWS Manajemen...

- Pilih tab Kemampuan
- Pilih akun sekunder yang ingin Anda edit dari bagian Cloud dan klik Edit
- Pilih Buat kebijakan DevOps Agen baru menggunakan templat dan ikuti petunjuknya

Praktik terbaik kebijakan kustom

- Berikan izin hanya-baca saja - Hindari izin yang memungkinkan modifikasi atau penghapusan sumber daya
- Gunakan izin tingkat sumber daya bila memungkinkan — Batasi akses ke sumber daya tertentu menggunakan pola atau tag ARN
- Tinjau dan audit izin secara berkala - Tinjau kebijakan IAM agen secara berkala untuk memastikan kebijakan tersebut tetap selaras dengan persyaratan keamanan Anda

Menyiapkan Autentikasi Pusat Identitas IAM

Autentikasi IAM Identity Center menyediakan cara terpusat untuk mengelola akses pengguna ke aplikasi web AWS DevOps Agent Space. Panduan ini menjelaskan cara mengonfigurasi autentikasi IAM Identity Center dan mengelola pengguna.

Prasyarat

Sebelum menyiapkan autentikasi IAM Identity Center, pastikan Anda memiliki:

- Pusat Identitas IAM diaktifkan di organisasi atau akun
- Izin administrator di Agen AWS DevOps
- Ruang Agen yang dikonfigurasi atau siap dibuat

Opsi otentikasi

AWS DevOps Agen menawarkan dua metode otentikasi untuk mengakses aplikasi web Agent Space:

Autentikasi IAM Identity Center — Direkomendasikan untuk lingkungan produksi. Menyediakan manajemen pengguna terpusat, integrasi dengan penyedia identitas eksternal, dan sesi hingga 12 jam.

Akses admin (otentikasi IAM) - Menyediakan akses cepat untuk administrator selama pengaturan dan konfigurasi awal. Sesi dibatasi hingga 30 menit.

Mengkonfigurasi Pusat Identitas IAM selama pembuatan Ruang Agen

Saat Anda membuat Ruang Agen, Anda dapat mengonfigurasi autentikasi Pusat Identitas IAM pada tab Access:

Langkah 1: Arahkan ke konfigurasi aplikasi Web

1. Setelah mengonfigurasi detail Ruang Agen dan akses AWS akun Anda, lanjutkan ke tab Akses
2. Anda akan melihat dua bagian: “Connect IAM Identity Center” dan “Admin access”

Langkah 2: Konfigurasikan integrasi Pusat Identitas IAM

Di bagian Connect [Agent Space] ke IAM Identity Center:

1. Verifikasi instans Pusat Identitas IAM — Konsol menampilkan instans Pusat Identitas mana yang akan mengelola akses pengguna Aplikasi Web (misalnya, `sso:ins-7223a9580931edbe`). Instance IAM Identity Center terdekat Anda akan secara otomatis diisi sebelumnya.
2. Pilih opsi Nama Peran Aplikasi Pusat Identitas IAM - Pilih salah satu dari tiga opsi:

Buat peran DevOps Agen baru secara otomatis (disarankan):

- Sistem secara otomatis membuat peran layanan baru dengan izin yang sesuai
- Ini adalah opsi paling sederhana dan berfungsi untuk sebagian besar kasus penggunaan

Tetapkan peran yang ada:

- Gunakan peran IAM yang sudah ada yang telah Anda buat
- Sistem akan memverifikasi peran memiliki izin yang diperlukan
- Pilih opsi ini jika organisasi Anda memiliki peran yang telah dibuat sebelumnya untuk Agen AWS DevOps

Membuat peran DevOps Agen baru menggunakan templat kebijakan:

- Gunakan detail kebijakan yang disediakan untuk membuat peran kustom Anda sendiri di Konsol IAM
- Pilih opsi ini jika Anda perlu menyesuaikan izin peran

Setelah mengklik Connect, sistem secara otomatis:

- Membuat atau mengonfigurasi peran IAM yang ditentukan
- Menyiapkan aplikasi Pusat Identitas IAM untuk Ruang Agen Anda
- Membangun hubungan kepercayaan antara IAM Identity Center dan aplikasi web Agent Space
- Mengkonfigurasi alur otentikasi OAuth 2.0 untuk akses pengguna yang aman

Alternatif: Menggunakan akses admin

Jika Anda ingin segera mengakses aplikasi web Agent Space tanpa menyiapkan Pusat Identitas IAM:

1. Di bagian akses Admin, perhatikan ARN Peran IAM yang menyediakan akses administrator (misalnya,) `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`
2. Klik tombol akses Admin biru untuk meluncurkan aplikasi web Agent Space dengan autentikasi IAM
3. Sesi menggunakan metode ini dibatasi hingga 30 menit

Note

Akses admin ditujukan untuk pengaturan dan konfigurasi awal. Untuk penggunaan produksi dan operasi yang sedang berlangsung, konfigurasi autentikasi IAM Identity Center.

Menambahkan Pengguna dan Grup

Setelah mengonfigurasi autentikasi Pusat Identitas IAM, Anda perlu memberi pengguna dan grup tertentu akses ke aplikasi web Agent Space:

Langkah 1: Akses manajemen pengguna

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Access
3. Di bawah Akses Pengguna, klik Kelola Pengguna dan Grup

Langkah 2: Tambahkan pengguna atau grup

1. Pilih Tambah Pengguna atau Grup
2. Cari pengguna atau grup di direktori Pusat Identitas IAM
3. Pilih kotak centang di samping pengguna atau grup yang ingin Anda tambahkan
4. Klik Tambah untuk memberi mereka akses

Pengguna yang dipilih sekarang dapat mengakses aplikasi web Agent Space menggunakan kredensi IAM Identity Center mereka.

Bekerja dengan penyedia identitas eksternal

Jika Anda menggunakan penyedia identitas eksternal (seperti Okta, Microsoft Entra ID, atau Ping Identity) dengan IAM Identity Center:

- Pengguna dan grup disinkronkan dari penyedia identitas eksternal Anda ke Pusat Identitas IAM
- Saat menambahkan pengguna dan grup ke aplikasi web Agent Space, Anda memilih dari direktori yang disinkronkan
- Atribut pengguna dan keanggotaan grup dikelola oleh penyedia identitas eksternal Anda
- Perubahan pada penyedia identitas Anda secara otomatis tercermin di Pusat Identitas IAM setelah sinkronisasi

Cara pengguna mengakses aplikasi web Agent Space

Setelah menambahkan pengguna ke Ruang Agen Anda:

1. Bagikan URL aplikasi web Agent Space dengan pengguna yang berwenang
2. Saat pengguna menavigasi ke URL, mereka diarahkan ke halaman login IAM Identity Center
3. Setelah memasukkan kredensialnya (dan menyelesaikan MFA jika dikonfigurasi), mereka dialihkan kembali ke aplikasi web Agent Space
4. Sesi mereka berlaku selama 8 jam secara default (dapat dikonfigurasi oleh administrator Pusat Identitas)

Mengelola akses pengguna

Anda dapat memperbarui akses pengguna kapan saja:

Menambahkan lebih banyak pengguna atau grup:

- Ikuti langkah yang sama yang dijelaskan di atas untuk menambahkan pengguna atau grup tambahan

Menghapus akses:

1. Di bagian Akses Pengguna, temukan pengguna atau grup yang akan dihapus
2. Klik tombol Hapus di samping nama mereka

3. Konfirmasikan penghapusan

Pengguna yang dihapus akan segera kehilangan akses, tetapi sesi aktif dapat berlanjut hingga kedaluwarsa.

Manajemen sesi

Sesi IAM Identity Center untuk aplikasi web Agent Space memiliki karakteristik sebagai berikut:

- Durasi sesi default - 8 jam
- Keamanan sesi - Cookie khusus HTTP untuk perlindungan yang ditingkatkan
- Otentikasi multi-faktor - Didukung saat dikonfigurasi di Pusat Identitas IAM
- Kredensial API — Kredensi Sigv4 berdurasi pendek (15 menit) dikeluarkan untuk panggilan API dan diperbarui secara otomatis

Untuk mengonfigurasi durasi sesi:

1. Arahkan ke konsol Pusat Identitas IAM
2. Buka Pengaturan > Otentikasi
3. Di bawah Durasi sesi, konfigurasi durasi pilihan Anda (dari 1 jam hingga 12 jam)
4. Pilih Save changes (Simpan perubahan)

Memutuskan Pusat Identitas

1. Di konsol Agent Space Anda, klik Tindakan di kanan atas dan pilih Putuskan sambungan dari Pusat Identitas IAM
2. Konfirmasikan dalam dialog konfirmasi

Menyiapkan Otentikasi Penyedia Identitas Eksternal (IDP)

Autentikasi penyedia identitas eksternal (iDP) memungkinkan organisasi Anda menggunakan penyedia identitas yang kompatibel dengan OIDC yang ada, seperti Okta atau Microsoft Entra ID, untuk mengelola akses pengguna ke aplikasi web Agent Space. AWS DevOps Pengguna masuk dengan kredensi perusahaan mereka langsung melalui IDP Anda, tanpa memerlukan AWS Pusat Identitas IAM.

Prasyarat

Sebelum menyiapkan otentikasi iDP eksternal, pastikan Anda memiliki:

- Penyedia identitas yang kompatibel dengan OIDC (Okta atau Microsoft Entra ID)
- Akses administrator ke penyedia identitas Anda
- Izin administrator untuk mengakses konsol AWS DevOps Agen
- Ruang Agen yang dikonfigurasi atau siap dibuat

Cara kerjanya

Saat Anda mengonfigurasi otentikasi iDP eksternal:

- Pengguna menavigasi ke URL aplikasi web Agent Space
- Mereka diarahkan ke halaman login penyedia identitas Anda
- Setelah mengautentikasi dengan kredensi perusahaan mereka, mereka diarahkan kembali ke aplikasi web
- Aplikasi web menukar token otentikasi dengan AWS kredensial berumur pendek yang dicakup ke Ruang Agen

Sesi berlaku hingga 8 jam. Kredensial secara otomatis disegarkan menggunakan token penyegaran OIDC tanpa mengharuskan pengguna untuk mengautentikasi ulang.

Mengkonfigurasi otentikasi iDP eksternal


Langkah 1: Daftarkan aplikasi di penyedia identitas Anda

Pilih penyedia identitas Anda dan ikuti petunjuk persiapan yang sesuai.

Opsi A: Okta


1. Di Konsol Admin Okta, navigasikan ke Applications > Applications dan pilih Create App Integration
2. Pilih OIDC - OpenID Connect sebagai metode login dan Aplikasi Web sebagai jenis aplikasi. Pilih Berikutnya
3. Tetapkan nama deskriptif untuk aplikasi (misalnya, AWS DevOps Agent)
4. Di bawah jenis Hibah, pastikan hal-hal berikut diperiksa:

- Kode Otorisasi (default)
- Refresh Token — Ini diperlukan untuk penyegaran sesi. Jika tidak diaktifkan, pengguna tidak akan dapat mempertahankan sesi.


 Note

Okta tidak mengaktifkan jenis hibah Refresh Token secara default. Anda harus mengaktifkannya secara eksplisit.

1. Biarkan pengalihan Masuk URIs sebagai nilai default untuk saat ini - Anda akan memperbaruinya setelah mengonfigurasi Ruang Agen
2. Di bawah Penugasan, tetapkan pengguna atau grup yang seharusnya memiliki akses
3. Pilih Simpan.
4. Pada tab Umum aplikasi, perhatikan nilai-nilai berikut:
 - ID Klien
 - Rahasia klien - Pilih Salin untuk menyimpan nilai ini dengan aman
5. Perhatikan domain Okta Anda — ini adalah URL Penerbit Anda (misalnya, `https://dev-12345678.okta.com`).

 Note

Pada tab Masuk, verifikasi Penerbit diatur ke URL Okta (bukan Dinamis). Ini memastikan URL penerbit yang stabil.

 Note

Jangan menambahkan klaim grup ke token ID di tab Klaim server otorisasi Anda. AWS DevOps Agen tidak menggunakan keanggotaan grup dari IDP Anda.

Opsi B: Microsoft Entra ID

1. Di portal Azure, navigasikan ke Microsoft Entra ID > Pendaftaran aplikasi > Pendaftaran baru

2. Tetapkan nama deskriptif (misalnya, AWS DevOps Agent)
3. Di bawah Jenis akun yang didukung, pilih opsi yang sesuai untuk organisasi Anda (biasanya hanya Akun di direktori organisasi ini)
4. Biarkan URI Redirect kosong untuk saat ini. Pilih Daftar
5. Pada halaman Ikhtisar aplikasi, perhatikan nilai-nilai berikut:
 - ID Aplikasi (klien) — digunakan sebagai ID Klien saat mengonfigurasi Ruang Agen
 - ID Direktori (penyewa) - digunakan untuk membangun URL Penerbit
6. Arahkan ke Sertifikat & rahasia > Rahasia klien baru
 - Tetapkan deskripsi dan periode kedaluwarsa
 - Pilih Tambah dan salin Nilai rahasia segera - itu tidak akan ditampilkan lagi
7. URL Penerbit untuk ID Entra mengikuti format ini. Ganti {tenant-id} dengan ID Direktori (penyewa) Anda dari langkah 5:
 - `https://login.microsoftonline.com/{tenant-id}/v2.0`

Note

Jangan aktifkan klaim opsional grup dalam konfigurasi Token. AWS DevOps Agen tidak menggunakan keanggotaan grup dari IDP Anda.

Langkah 2: Aktifkan Aplikasi Operator dengan otentikasi IDP

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Access
3. Di bawah Akses pengguna, pilih Penyedia identitas eksternal
4. Dalam formulir konfigurasi, konfigurasikan yang berikut ini:
 - Penyedia Identitas — Pilih penyedia identitas Anda (Okta atau Microsoft Entra ID)
 - URL Penerbit — URL penerbit OIDC dari penyedia identitas Anda
 - ID Klien — ID klien dari aplikasi OIDC yang Anda buat
 - Rahasia Klien - Rahasia klien dari aplikasi OIDC Anda
5. Di bawah Nama Peran Aplikasi Penyedia Identitas, pilih salah satu dari tiga opsi:
 - Buat peran DevOps Agen baru secara otomatis (disarankan) - Membuat peran layanan baru dengan izin yang sesuai

- Menetapkan peran yang ada — Gunakan peran IAM yang sudah ada yang telah Anda buat
 - Membuat peran DevOps Agen baru menggunakan templat kebijakan — Gunakan detail yang disediakan untuk membuat peran Anda sendiri di Konsol IAM
6. Tinjau peringatan peringatan URL Callback yang ditampilkan di bagian bawah formulir. Salin URL ini — Anda harus menambahkannya ke pengalihan yang diizinkan penyedia identitas Anda URIs sebelum pengguna dapat masuk.
7. Pilih Connect

Setelah memilih Connect, konsol menampilkan Konfigurasi Penyedia Identitas Eksternal dengan detail berikut:

- Penyedia — Penyedia identitas yang Anda pilih
- URL Penerbit - URL penerbit OIDC yang dikonfigurasi
- ID Klien - ID klien yang dikonfigurasi
- IAM Role ARN — Peran IAM yang digunakan untuk akses pengguna
- URL Callback - Konfigurasi URL ini di penyedia identitas Anda sebagai URI pengalihan yang diizinkan
- Login URL — Gunakan URL ini untuk mengakses aplikasi web melalui penyedia identitas Anda

Langkah 3: Tambahkan URL callback ke penyedia identitas Anda

Okta

1. Di Konsol Admin Okta, arahkan ke tab Umum aplikasi Anda
2. Di bawah Login, pilih Edit
3. Tambahkan URL callback sebagai URI pengalihan Masuk:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opsional) Atur URI Inisiate login untuk mengaktifkan login yang dimulai IDP dari dasbor Okta:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Disarankan) Tambahkan URI pengalihan Keluar untuk mengarahkan pengguna kembali ke aplikasi web setelah logout. Tanpa ini, pengguna mungkin melihat halaman kesalahan saat keluar:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Pilih Simpan.

ID Microsoft Entra

1. Di portal Azure, navigasikan ke halaman Otentikasi aplikasi Anda
2. Di bawah Konfigurasi Platform, pilih Tambahkan platform > Web
3. Masukkan URL callback sebagai URI Redirect:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opsional) Tambahkan URI pengalihan keluar untuk mengarahkan pengguna kembali ke aplikasi web setelah logout:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. Pilih Konfigurasi

Langkah 4: Verifikasi konfigurasi

1. Arahkan ke URL Login yang ditampilkan di konsol:
 - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Anda harus diarahkan ke halaman login penyedia identitas Anda
3. Masuk dengan kredensi perusahaan Anda
4. Setelah otentikasi berhasil, Anda diarahkan kembali ke aplikasi web Agent Space

Memperbarui konfigurasi iDP

Anda dapat memutar rahasia klien tanpa memutuskan sambungan:

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Access
3. Di bawah Konfigurasi Penyedia Identitas Eksternal, pilih Putar rahasia klien
4. Masukkan Rahasia Klien baru
5. Pilih Simpan.

Untuk mengubah bidang konfigurasi iDP lainnya (seperti URL Penerbit, ID Klien, atau penyedia identitas), Anda harus memutuskan sambungan iDP yang ada dan mengonfigurasi yang baru.

Cara pengguna mengakses aplikasi web Agent Space

Setelah mengkonfigurasi otentikasi iDP eksternal:

- Bagikan URL aplikasi web Agent Space dengan pengguna yang berwenang
- Saat pengguna menavigasi ke URL, mereka diarahkan ke halaman login penyedia identitas Anda
- Setelah memasukkan kredensialnya (dan menyelesaikan MFA jika dikonfigurasi oleh IDP Anda), mereka dialihkan kembali ke aplikasi web Agent Space
- Sesi disegarkan secara otomatis - lihat [Manajemen sesi](#) untuk detailnya

Manajemen sesi

Sesi IDP eksternal untuk aplikasi web Agent Space memiliki karakteristik sebagai berikut:

- Durasi sesi - Sesi browser berlangsung hingga 8 jam. Ini tidak dapat dikonfigurasi di AWS DevOps Agen. Jika masa pakai sesi IDP Anda melebihi 8 jam, pengguna dapat diautentikasi ulang secara otomatis pada kunjungan berikutnya tanpa memasukkan kredensialnya. Konfigurasi sesi IDP dan masa pakai token Anda sesuai dengan persyaratan keamanan organisasi Anda.
- Penyegaran kredensial - Sesi disegarkan secara otomatis menggunakan token penyegaran OIDC tanpa mengharuskan pengguna untuk mengautentikasi ulang
- Otentikasi multi-faktor — Didukung saat dikonfigurasi di penyedia identitas Anda. IDP menangani MFA selama login — tidak ada konfigurasi tambahan yang diperlukan di Agen AWS DevOps

Perilaku logout

Saat pengguna mengklik Logout di aplikasi web:

1. Semua cookie sesi segera dihapus
2. Pengguna dialihkan ke titik akhir logout OIDC penyedia identitas untuk mengakhiri sesi SSO
3. Jika URI pengalihan keluar dikonfigurasi, pengguna akan diarahkan kembali ke halaman selamat datang aplikasi web

Mencabut akses pengguna

Untuk segera mencabut akses pengguna, Anda dapat mencabut sesi mereka secara langsung di portal admin penyedia identitas Anda:

- Okta - Di Konsol Admin Okta, arahkan ke Direktori > Orang, pilih pengguna, pilih Tindakan Lainnya > Hapus Sesi Pengguna
- Microsoft Entra ID - Di portal Azure, navigasikan ke Pengguna, pilih pengguna, dan pilih Cabut sesi

Pertimbangan keamanan

Penyimpanan rahasia klien - Rahasia klien yang Anda berikan selama penyiapan dienkripsi menggunakan kunci KMS yang dikelola pelanggan jika Anda menyediakannya saat membuat Ruang Agen, atau kunci milik layanan sebaliknya. Itu tidak pernah dikembalikan dalam respons API atau ditampilkan di konsol setelah konfigurasi awal.

Rotasi rahasia klien - Rahasia klien Entra memiliki kedaluwarsa yang dapat dikonfigurasi. Tetapkan pengingat untuk memutar rahasia sebelum kedaluwarsa menggunakan opsi Rotate client secret di konsol AWS DevOps Agen. Jika rahasia kedaluwarsa, pengguna tidak akan dapat masuk sampai diputar.

Manajemen seumur hidup token — Masa pakai token (token akses, token penyegaran) yang dikeluarkan oleh penyedia identitas Anda dikendalikan oleh konfigurasi IDP Anda. Kami merekomendasikan untuk mengonfigurasi masa pakai token yang sesuai di IDP Anda:

- Okta — Konfigurasikan masa pakai token di bawah Keamanan > API > Server Otorisasi > Kebijakan Akses
- Microsoft Entra ID — Konfigurasikan masa pakai token menggunakan kebijakan seumur hidup [token](#)

Klaim grup — Jangan aktifkan klaim grup dalam konfigurasi token penyedia identitas Anda. AWS DevOps Agen saat ini tidak menggunakan keanggotaan grup dari IDP Anda.

Pengenal pengguna — AWS DevOps Agen menggunakan klaim khusus penyedia untuk mengidentifikasi pengguna secara unik:

- Okta — Menggunakan sub klaim dari token ID
- Microsoft Entra ID — Menggunakan klaim oid (pengenal objek) dari token ID

Pengidentifikasi ini tidak dapat diubah dan muncul di CloudTrail log untuk tujuan audit.

Memutuskan sambungan iDP eksternal

1. Di konsol AWS DevOps Agen, pilih Ruang Agen
2. Buka tab Access
3. Di bawah Akses pengguna, pilih Putuskan sambungan
4. Tinjau dampak yang tercantum dalam dialog konfirmasi dan konfirmasi

Memutuskan sambungan akan:

- Hapus konfigurasi iDP dari Ruang Agen
- Mencegah pengguna masuk melalui penyedia identitas eksternal
- Hapus obrolan individual dan riwayat artefak yang terkait dengan akun pengguna iDP

Sesi pengguna aktif akan berlanjut hingga kedaluwarsa atau penyegaran kredensial berikutnya gagal.

Pemecahan masalah

- Pengalihan ke IDP gagal — Verifikasi URL Penerbit cocok dengan titik akhir penemuan OIDC IDP Anda. Untuk Okta, pastikan Penerbit diatur ke URL Okta (bukan Dinamis) pada tab Masuk. Untuk Entra, gunakan `https://login.microsoftonline.com/{tenant-id}/v2.0` formatnya.
- Akses ditolak atau kesalahan kebijakan (Okta) - Verifikasi pengguna atau grup mereka ditetapkan ke aplikasi di bawah Penugasan. Periksa Masuk > Aturan Kebijakan Masuk.
- Kesalahan konfigurasi IDP setelah login — Penyedia identitas Anda tidak mengembalikan token penyegaran. Pastikan `offline_access` cakupan dan jenis hibah token refresh diaktifkan:
 - Okta — Buka tab Umum aplikasi Anda dan aktifkan kotak centang Refresh Token di bawah Jenis Grant
 - Entra - Buka izin API dan pastikan terdaftar di bawah izin `offline_access` yang didelegasikan
- Otentikasi berhasil tetapi aplikasi web menunjukkan kesalahan — Verifikasi URI pengalihan di iDP Anda sama persis dengan URL Callback yang ditampilkan di konsol Agen. AWS DevOps
- Kegagalan otentikasi — Jika klaim opsional grup diaktifkan di IDP Anda, nonaktifkan. AWS DevOps Agen tidak menggunakan klaim grup.
- Login gagal setelah otentikasi IDP - Untuk Entra, verifikasi tidak **requestedAccessTokenVersion** diatur ke **null** dalam Manifest aplikasi. Untuk Okta, verifikasi URL Penerbit sudah benar.

- Halaman kesalahan setelah mengklik Keluar (Okta) - Jika Anda melihat **post_logout_redirect_uri** kesalahan setelah keluar, tambahkan **https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome** sebagai URI pengalihan Keluar di tab Umum aplikasi Okta Anda.
- Pengguna tetap berada di halaman penyedia identitas setelah logout (Entra) — Untuk mengarahkan pengguna kembali ke aplikasi web setelah logout, tambahkan **https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome** sebagai URI Pengalihan di halaman Otentikasi aplikasi Entra Anda.

Enkripsi saat istirahat untuk AWS DevOps Agen

AWS DevOps Agen mengenkripsi semua data pelanggan saat istirahat. Secara default, AWS DevOps Agen menggunakan kunci yang AWS dimiliki untuk mengenkripsi data Anda secara otomatis tanpa biaya tambahan. Anda tidak dapat melihat, mengelola, atau mengaudit penggunaan kunci yang AWS dimiliki. Namun, Anda tidak perlu mengambil tindakan apa pun untuk melindungi kunci-kunci ini. Data Anda diamankan secara otomatis.

Anda dapat memilih untuk mengenkripsi data menggunakan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola di AWS Key Management Service (AWS KMS). Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti berikut:

- Menetapkan dan memelihara kebijakan utama
- Mengaktifkan dan menonaktifkan kebijakan utama
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan

Untuk informasi selengkapnya, lihat [Kunci terkelola pelanggan](#) di Panduan Pengembang Layanan Manajemen AWS Kunci.

Note

AWS DevOps Agen secara otomatis mengaktifkan enkripsi saat istirahat menggunakan kunci yang AWS dimiliki untuk melindungi data pelanggan tanpa biaya. Biaya AWS KMS

standar berlaku saat Anda menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang harga, lihat [harga Layanan Manajemen AWS Utama](#).

Kunci yang dikelola pelanggan

Kunci yang dikelola pelanggan adalah kunci KMS di AWS akun Anda yang Anda buat, miliki, dan kelola. Anda memiliki kendali penuh atas kunci KMS ini, termasuk menetapkan dan memelihara kebijakan utama mereka.

Saat Anda mengonfigurasi kunci terkelola pelanggan, AWS DevOps Agen menggunakannya untuk melindungi data sumber daya yang sensitif. AWS DevOps Agen menggunakan [enkripsi amplop dengan keyring](#) hierarkis AWS Encryption SDK. Kunci KMS Anda digunakan untuk menghasilkan kunci cabang, yang pada gilirannya melindungi data Anda.

Anda dapat menentukan kunci terkelola pelanggan saat membuat sumber daya berikut:

- Ruang Agen — Mengenkripsi detail Ruang Agen dan konten yang dibuat dari Aplikasi Web DevOps Agen yang terkait dengan investigasi, keterampilan, dan obrolan.
- Layanan — Mengenkripsi kredensi layanan pihak ketiga saat istirahat.

Untuk mengonfigurasi kunci terkelola pelanggan di AWS DevOps Agen, ikuti langkah-langkah berikut.

Langkah 1: Buat kunci yang dikelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan konsol AWS KMS atau AWS KMS API. Kuncinya harus memenuhi persyaratan berikut:

Properti	Persyaratan
Tipe Kunci	Simetris
Spesifikasi kunci	SYMMETRIC_DEFAULT
Penggunaan kunci	ENCRYPT_DECRYPT

Note

AWS DevOps Agen hanya mendukung kunci KMS enkripsi simetris dengan spesifikasi SYMMETRIC_DEFAULT kunci dan penggunaan kunci. ENCRYPT_DECRYPT Tombol Multi-Region dan kunci asimetris saat ini tidak didukung.

Untuk informasi selengkapnya, lihat [Membuat kunci terkelola pelanggan simetris](#) di Panduan Pengembang Layanan Manajemen AWS Kunci.

Langkah 2: Tetapkan kebijakan kunci

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya.

Kebijakan utama Anda harus memberikan izin kepada prinsipal panggilan (identitas IAM Anda) dan layanan AWS DevOps Agen. AWS DevOps Agen mengakses kunci Anda menggunakan dua set kredensial:

1. Kredensial pemanggil Anda — Digunakan untuk semua operasi sinkron, termasuk validasi kunci, enkripsi pada waktu pembuatan sumber daya, dan panggilan API apa pun yang mengembalikan respons langsung ke pemanggil.
2. AWS DevOps Prinsipal layanan agen — Digunakan untuk operasi asinkron yang berjalan di latar belakang, seperti investigasi operasional, analisis insiden, korelasi peristiwa, dan pembuatan analisis akar penyebab.

Tabel berikut mencantumkan tindakan KMS yang diperlukan:

Aksi KMS	Deskripsi
<code>kms:DescribeKey</code>	Validasi konfigurasi kunci pada waktu pembuatan sumber daya
<code>kms:GenerateDataKey</code>	Hasilkan kunci enkripsi data untuk enkripsi amplop
<code>kms:Decrypt</code>	Dekripsi data

Aksi KMS	Deskripsi
kms:Encrypt	Enkripsi data
kms:ReEncrypt	Enkripsi ulang data di bawah kunci yang sama atau berbeda

AWS DevOps Agen memvalidasi semua izin ini pada waktu konfigurasi menggunakan operasi dry-run. Jika ada izin yang hilang, permintaan gagal dengan pengecualian.

Berikut ini adalah contoh kebijakan kunci. Ganti nilai placeholder dengan milik Anda sendiri.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCallerAccessViaService",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/DevOpsAgentUserRole"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "aidevops.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowDevOpsAgentServiceDescribeKeyAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
```

```
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowDevOpsAgentAccessForAgentSpace",
  "Effect": "Allow",
  "Principal": {
    "Service": "aidevops.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:agentspace/*"
    },
    "StringLike": {
      "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-east-1:111122223333:agentspace/*"
    }
  }
},
{
  "Sid": "AllowDevOpsAgentAccessForService",
  "Effect": "Allow",
  "Principal": {
    "Service": "aidevops.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:service/*"
    }
  },
}
```

```
    "StringLike": {
      "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-
east-1:111122223333:service/*"
    }
  }
}
]
```

Kebijakan tersebut berisi pernyataan berikut:

- **AllowKeyAdministration**— Memberikan root akun akses administratif penuh ke kunci. Ganti 111122223333 dengan ID AWS akun Anda.
- **AllowCallerAccessViaService**— Memberikan kepala sekolah IAM Anda izin KMS yang diperlukan untuk semua operasi Agen sinkron. AWS DevOps Ini termasuk validasi kunci pada waktu pembuatan sumber daya, serta mengenkripsi dan mendekripsi operasi untuk setiap panggilan API yang mengembalikan respons langsung ke pemanggil. `kms:ViaServiceKondisi` ini memastikan bahwa Anda dapat menggunakan kunci hanya melalui layanan AWS DevOps Agen. Ganti 111122223333 dengan ID AWS akun Anda dan `us-east-1` dengan AWS Wilayah Anda.
- **AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService**— Memberikan kepala `aidevops.amazonaws.com` layanan izin KMS yang diperlukan untuk operasi asinkron. AWS DevOps Agen menggunakan prinsip layanan ini untuk mengenkripsi dan mendekripsi data Anda saat melakukan operasi latar belakang seperti investigasi operasional, menganalisis insiden, menghubungkan peristiwa di seluruh layanan, dan menghasilkan analisis akar penyebab. Tanpa akses ini, AWS DevOps Agen tidak dapat membaca data terenkripsi yang diperlukan untuk melakukan penyelidikan atas nama Anda. `aws:SourceArnKondisi` ini membatasi akses ke permintaan yang berasal dari sumber daya AWS DevOps Agen Anda, dan `kms:EncryptionContext` kondisi memastikan bahwa konteks enkripsi cocok dengan sumber daya Anda. ARNs Ganti 111122223333 dengan ID AWS akun Anda dan `us-east-1` dengan AWS Wilayah Anda.

Untuk informasi selengkapnya tentang kebijakan utama, lihat [Kebijakan utama di AWS KMS](#) di Panduan Pengembang Layanan Manajemen AWS Utama.

Langkah 3: Tentukan kunci saat membuat sumber daya

Setelah membuat kunci dan mengonfigurasi kebijakan kunci, Anda dapat menentukan kunci saat membuat sumber daya AWS DevOps Agen.

Konsol

Untuk mengonfigurasi kunci terkelola pelanggan saat membuat Ruang Agen di konsol:

1. Buka konsol AWS DevOps Agen.
2. Pilih Buat Ruang Agen atau Daftar Layanan.
3. Masukkan detail ruang agen (nama, deskripsi, dan peran IAM).
4. Perluas bagian Konfigurasi Lanjutan.
5. Di bawah Jenis kunci enkripsi, pilih Kunci terkelola pelanggan.
6. Pilih kunci KMS dari daftar dropdown, atau masukkan ARN kunci KMS.
7. Tinjau kebijakan kunci yang ditampilkan di bagian Kebijakan kunci yang dapat diperluas. Pastikan bahwa Anda telah melampirkan kebijakan ini ke kunci KMS Anda. Anda dapat menggunakan tombol salin untuk menyalin kebijakan.
8. Selesaikan konfigurasi yang tersisa dan pilih Buat.

Note

Jika Anda tidak melihat kunci KMS Anda di daftar dropdown, verifikasi bahwa kunci tersebut memenuhi persyaratan di [Langkah 1](#) dan yang Anda miliki `kms:ListKeys` dan izin `kms:DescribeKey`

API

Membuat Ruang Agen dengan kunci yang dikelola pelanggan

Tentukan `kmsKeyArn` parameter saat membuat ruang agen. Nilai harus berupa ARN kunci KMS penuh.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Mendaftarkan Layanan dengan kunci yang dikelola pelanggan

Tentukan `kmsKeyArn` parameter saat mendaftarkan layanan. Nilai harus berupa ARN kunci KMS penuh. Parameter ini didukung di semua jenis layanan, termasuk Dynatrace,,, ServiceNow, PagerDuty GitLab GitHub, dan Server MCP.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

Note

Anda harus menentukan kunci yang dikelola pelanggan pada waktu pembuatan sumber daya. Anda tidak dapat menambahkan atau mengubah kunci terkelola pelanggan untuk sumber daya yang ada.

AWS DevOps Konteks enkripsi agen

[Konteks enkripsi](#) adalah sekumpulan pasangan nilai kunci non-rahasia yang berisi informasi kontekstual tambahan tentang data. AWS KMS menggunakan konteks enkripsi sebagai [data otentikasi tambahan untuk mendukung enkripsi yang diautentikasi](#). Saat Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus menyertakan konteks enkripsi yang sama dalam permintaan.

AWS DevOps Agen menggunakan konteks enkripsi berikut pada semua operasi kriptografi:

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

Nilai konteks enkripsi adalah ARN dari sumber daya AWS DevOps Agen yang dienkripsi. Anda dapat menggunakan konteks enkripsi ini dalam kondisi kebijakan utama Anda dan di AWS CloudTrail log untuk mengaudit bagaimana kunci Anda digunakan.

Manajemen kunci

Jika Anda menonaktifkan atau menjadwalkan penghapusan kunci KMS Anda, AWS DevOps Agen tidak dapat mendekripsi data Anda. Ini menghasilkan `AccessDeniedException` kesalahan pada operasi yang membaca data terenkripsi.

Important

Jika Anda memilih untuk menggunakan kunci yang dikelola pelanggan, Anda bertanggung jawab untuk mengelola kunci dan izinnya. Jika kunci dinonaktifkan atau dihapus, atau jika AWS DevOps Agen kehilangan izin untuk menggunakan kunci, Anda kehilangan akses ke data terenkripsi.

Tabel berikut menjelaskan skenario kegagalan umum:

Tindakan	Dampak
Izin kebijakan utama dicabut	<code>AccessDeniedException</code> pada operasi enkripsi dan dekripsi
Kunci KMS dinonaktifkan	<code>DisabledException</code> pada operasi enkripsi dan dekripsi
Kunci KMS dijadwalkan untuk dihapus	<code>KMSInvalidStateException</code> pada operasi enkripsi dan dekripsi
Kunci KMS dihapus	Kehilangan data permanen — data terenkripsi tidak dapat dipulihkan

Sebelum menonaktifkan atau menghapus kunci:

1. Verifikasi bahwa tidak ada sumber daya AWS DevOps Agen aktif yang bergantung pada kuncinya.
2. Pertimbangkan untuk menonaktifkan kunci terlebih dahulu untuk menguji dampaknya sebelum menjadwalkan penghapusan.
3. AWS KMS memberlakukan masa tunggu minimum sebelum penghapusan kunci, memberi Anda waktu untuk membatalkan jika diperlukan.

Catatan:: AWS DevOps Agen tidak secara otomatis mengenkripsi ulang data di bawah kunci baru. Jika Anda perlu memutar ke kunci yang dikelola pelanggan baru, Anda harus membuat sumber daya baru dengan kunci baru.

Memantau kunci enkripsi Anda

Saat Anda menggunakan kunci yang dikelola pelanggan dengan AWS DevOps Agen, Anda dapat menggunakannya [AWS CloudTrail](#) untuk melacak permintaan yang dikirim AWS DevOps Agen ke AWS KMS.

Anda dapat memfilter CloudTrail acara berdasarkan:

- Sumber acara - kms.amazonaws.com
- Kunci konteks enkripsi - aws-crypto-ec:aws:aidevops:arn
- ARN Kunci — ARN kunci yang dikelola pelanggan Anda dalam parameter permintaan

Untuk informasi selengkapnya, lihat [Mencatat panggilan API AWS KMS AWS CloudTrail](#) di Panduan Pengembang Layanan Manajemen AWS Kunci.

VPC Endpoint (AWS PrivateLink)

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC dan AWS DevOps Agen Anda. Anda dapat mengakses AWS DevOps Agen seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi Direct Connect. Instans di VPC Anda tidak memerlukan alamat IP publik untuk AWS DevOps mengakses Agen.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk Agen. AWS DevOps

Untuk informasi selengkapnya, lihat [Akses AWS layanan melalui AWS PrivateLink](#) di [_AWS PrivateLink Guide_](#).

Pertimbangan untuk titik akhir AWS DevOps Agen VPC

Sebelum Anda menyiapkan titik akhir antarmuka untuk AWS DevOps Agen, tinjau [Pertimbangan](#) di [PrivateLink _AWS Guide_](#).

AWS DevOps Agen mendukung melakukan panggilan API melalui titik akhir VPC berikut.

Kategori	Sufiks titik akhir
AWS DevOps Tindakan API Pesawat Kontrol Agen	aidevops
AWS DevOps Operasi Runtime Agen	aidevops-dataplane
AWS DevOps Acara Agen Webhook	event-ai

Buat titik akhir antarmuka untuk Agen AWS DevOps

Anda dapat membuat titik akhir antarmuka untuk AWS DevOps Agen menggunakan konsol Amazon VPC atau Antarmuka Baris AWS Perintah (AWS CLI). Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di `_AWS PrivateLink Guide_`.

Buat titik akhir antarmuka untuk AWS DevOps Agen menggunakan nama layanan berikut:

- `com.amazonaws. {wilayah} .aidevops`
- `com.amazonaws. {wilayah} .aidevops-dataplane`
- `com.amazonaws. {wilayah} .event-ai`

Setelah Anda membuat titik akhir, Anda memiliki opsi untuk mengaktifkan nama host DNS pribadi. Aktifkan pengaturan ini dengan memilih Aktifkan Nama DNS privat di konsol VPC saat Anda membuat VPC endpoint.

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API ke AWS DevOps Agen menggunakan nama DNS Regional defaultnya. Contoh berikut menunjukkan format nama DNS Regional default.

- `cp.aidevops. {wilayah} .api.aws`
- `dp.aidevops. {wilayah} .api.aws`
- `acara-ai. {wilayah} .api.aws`

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh ke AWS DevOps Agen melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan ke AWS DevOps Agen dari VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (AWS akun, pengguna IAM, dan peran IAM).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan menggunakan kebijakan titik akhir](#) di [_Panduan PrivateLink AWS_](#).

Validasi kepatuhan untuk Agen AWS DevOps

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS layanan sebagai bagian dari beberapa program AWS kepatuhan. AWS DevOps Agen berada dalam lingkup untuk program kepatuhan berikut: BIO, C5, CISPE, CPSTIC, ENS Tinggi, FINMA, GNS, GSMA, HITRUST, IRAP, ISMAP, ISO (ISO/IEC 27001, 27017, 27018, 27701, 22301, 20000, 9001), CSA STAR, MTC, OSPAR, PCI, Pinakes, dan SOC. PiTuKri Selain itu, AWS DevOps Agen memenuhi syarat HIPAA. Auditor pihak ketiga kami akan meninjau dan menguji AWS DevOps Agen selama siklus audit berikutnya untuk program kepatuhan ini.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWS layanan dalam lingkup oleh program kepatuhan](#). Untuk informasi umum, lihat [Program kepatuhan AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh laporan di AWS Artifak](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS DevOps Agen ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan memulai cepat keamanan dan kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS

- [AWS sumber daya kepatuhan](#) - Kumpulan buku kerja dan panduan yang mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) — AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#) — AWS Layanan ini memberikan pandangan komprehensif tentang status keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi AWS sumber daya Anda dan untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Kuota

AWS DevOps Kuota agen mencakup jumlah ruang agen, investigasi bersamaan, dan banyak lagi. Anda dapat meminta kenaikan untuk beberapa kuota, tetapi tidak semua kuota dapat ditingkatkan. Peningkatan ini tidak diberikan segera, jadi mungkin perlu beberapa jam hingga sehari-hari agar kenaikan Anda menjadi efektif. Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah.

Tabel berikut menjelaskan kuota untuk AWS DevOps Agen.

Nama	Default	Dapat disesuaikan	Deskripsi
Ruang agen per akun per Wilayah	100	Ya	Jumlah maksimum ruang agen yang dapat Anda buat per akun di setiap AWS Wilayah.
Investigasi bersamaan per ruang agen	3	Ya	Jumlah maksimum investigasi resolusi insiden yang dapat berjalan secara bersamaan dalam satu ruang agen.
Evaluasi bersamaan per ruang agen	1	Tidak	Jumlah maksimum evaluasi pencegahan insiden yang dapat berjalan secara bersamaan dalam satu ruang agen.
Pemanggilan sesuai permintaan bersamaan per ruang agen	10	Ya	Jumlah maksimum DevOps pemanggilan sesuai permintaan yang dapat berjalan secara bersamaan

Nama	Default	Dapat disesuaikan	Deskripsi
			dalam satu ruang agen.

Meminta peningkatan kuota

Anda dapat meminta kenaikan kuota dengan menggunakan salah satu opsi berikut:

- Dari Konsol AWS Manajemen — Buka konsol [Service Quotas](#). Di panel navigasi, pilih Layanan AWS . Pilih DevOps Agen, pilih kuota, dan ikuti petunjuk untuk meminta kenaikan kuota. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.
- Dari AWS CLI — Gunakan perintah CLI [request-service-quota-increase](#) AWS . Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS DevOps Agen. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Date
Instruksi agen	Menambahkan instruksi global dan khusus agen (AGENTS.md) yang berlaku untuk setiap sesi.	21 Mei 2026
Lewati keterampilan dan status triase SKIPPED	Menambahkan contoh keterampilan penyaringan insiden (lewat jendela pemeliharaan), keputusan triase SKIPPED, instruksi untuk mengoreksi keputusan triase, dan acara Investigasi Dilewati. EventBridge	20 Mei 2026
Mengirim lampiran file	Menambahkan dokumentasi untuk melampirkan gambar, dokumen, dan file kode ke pesan obrolan. Ini termasuk jenis file yang didukung, batas, dan kasus penggunaan.	19 Mei 2026
Prioritas rekomendasi	Menambahkan peringkat AI-powered backlog, termasuk menyesuaikan prioritas melalui obrolan dan stabilitas peringkat.	13 Mei 2026

Perubahan	Deskripsi	Date
Plugin Kode Claude untuk MCP	Menambahkan referensi ke plugin Claude Code sampel di bagian integrasi MCP.	12 Mei 2026
Izin pagar pembatas	Menambahkan model pagar pembatas kebijakan sesi, yang mencakup izin default, izin tambahan yang didukung, dan izin yang diblokir oleh pagar pembatas.	7 Mei 2026
IP statis baru	Menambahkan alamat IP statis baru untuk koneksi keluar di semua wilayah yang didukung.	7 Mei 2026
Riwayat dokumen	Ditambahkan halaman sejarah dokumen untuk melacak dokumentasi baru.	5 Mei 2026
Interface dengan Agen DevOps	Menambahkan dokumentasi untuk lima metode akses: aplikasi web, MCP, ACP, webhooks, dan API.	April 28, 2026
Validasi kepatuhan	Ditambahkan halaman validasi kepatuhan khusus.	April 15, 2026
Memulai menggunakan AWS CloudFormation	CloudFormation Menambahkan panduan memulai.	Maret 29, 2026
Menghubungkan ke alat yang dihosting secara pribadi	Ditambahkan dokumentasi untuk koneksi pribadi.	Maret 29, 2026
Titik akhir antarmuka VPC	Ditambahkan VPC endpoint ()AWS PrivateLink dokumentasi.	Maret 29, 2026

Perubahan	Deskripsi	Date
EventBridge Integrasi Amazon	Menambahkan panduan EventBridge integrasi untuk aplikasi berbasis peristiwa.	Maret 28, 2026
EventBridge referensi detail acara	Menambahkan referensi detail acara untuk EventBridge integrasi.	Maret 28, 2026
Kuota	Ditambahkan halaman kuota layanan.	Maret 28, 2026
Menghubungkan Grafana	Menambahkan dokumentasi integrasi telemetri Grafana.	Maret 27, 2026
Menghubungkan Azure	Ditambahkan dokumentasi integrasi Azure.	Maret 27, 2026
Menghubungkan Sumber Daya Azure	Menambahkan panduan koneksi Azure Resources.	Maret 27, 2026
Menghubungkan Azure DevOps	Menambahkan panduan DevOps koneksi Azure.	Maret 27, 2026
Menghubungkan PagerDuty	Menambahkan dokumentasi integrasi PagerDuty komunikasi.	Maret 27, 2026
Migrasi dari Pratinjau Publik ke GA	Menambahkan panduan migrasi untuk Pratinjau Publik ke Ketersediaan Umum.	Maret 27, 2026
Ketersediaan umum	Ini adalah rilis ketersediaan umum awal AWS DevOps Agen.	Maret 30, 2026

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