



Panduan Pengguna untuk versi 1.17.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Panduan Pengguna untuk versi 1.17.0

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

.....	ix
Apa itu SimSpace Weaver?	1
Konsep utama	1
Bagaimana cara SimSpace Weaver kerja	2
Bagaimana Anda menggunakan SimSpace Weaver	5
Skema simulasi	6
Pekerja dan unit sumber daya	6
Jam simulasi	6
Partisi	7
Kain Negara	7
Entitas	7
Aplikasi	8
Contoh kasus penggunaan	11
AWS SimSpace Weaver Akhir dari Dukungan	12
Pengaturan	13
Siapkan akun Anda	13
Mendaftar untuk Akun AWS	13
Buat pengguna dengan akses administratif	14
Tambahkan izin untuk digunakan SimSpace Weaver	15
Siapkan lingkungan lokal Anda	16
AL2 di Docker	17
AL2 di WSL	19
Penggunaan perangkat lunak berlisensi	23
Memulai	24
Tutorial mulai cepat	24
Langkah 1: Aktifkan logging (opsional)	25
Langkah 2: Mulai cepat dengan klien konsol (opsi 1)	25
Langkah 2: Mulai cepat dengan klien Unreal Engine (opsi 2)	26
Hentikan dan hapus simulasi Anda	26
Pemecahan Masalah	26
Tutorial terperinci	27
Langkah 1: Aktifkan logging (opsional)	28
Langkah 2: Mulai simulasi Anda	28
Langkah 3: Periksa log (opsional)	35

Langkah 4: Lihat simulasi Anda	37
Langkah 5: Hentikan dan hapus simulasi Anda	38
Pemecahan masalah	39
Bekerja dengan SimSpace Weaver	40
Mengkonfigurasi simulasi Anda	40
Parameter konfigurasi simulasi	41
Versi SDK	42
Sifat simulasi	42
Pekerja	43
Jam	44
Strategi partisi	47
Domain	48
Durasi maksimum	58
Nilai maksimum	58
Nilai default	58
Nilai minimum	58
Memulai simulasi menggunakan konsol	59
Status simulasi yang mencapai durasi maksimumnya	59
Mengembangkan aplikasi	59
Aplikasi spasial	60
Aplikasi kustom	60
Mengembangkan aplikasi klien	61
Dapatkan alamat IP dan nomor port	62
Meluncurkan klien tampilan Unreal Engine	65
Pemecahan Masalah	66
Pembangunan lokal	67
Langkah 1: Luncurkan simulasi lokal Anda	67
Langkah 2: Lihat simulasi lokal Anda	69
Langkah 3: Hentikan simulasi lokal Anda (opsional pada Windows)	69
Memecahkan masalah pembangunan lokal	70
SimSpace Weaver aplikasi SDK	71
Metode API mengembalikan a Result	72
Berinteraksi dengan SDK aplikasi di tingkat atas	72
Manajemen simulasi	73
Langganan	76
Entitas	77

Peristiwa entitas	89
Result dan penanganan kesalahan	95
Generik dan jenis domain	98
Operasi SDK aplikasi lain-lain	98
SimSpace Weaver kerangka demo	100
Bekerja dengan kuota	101
Dapatkan batasan untuk sebuah aplikasi	102
Dapatkan jumlah sumber daya yang digunakan oleh aplikasi	103
Setel ulang metrik	104
Melebihi batas	104
Kehabisan memori	104
Praktik terbaik	105
Simulasi debugging	105
Gunakan SimSpace Weaver Local dan lihat output konsol	105
Lihat log Anda di Amazon CloudWatch Logs	106
Gunakan describe Panggilan API	106
Connect klien	107
Debugging simulasi lokal	107
Wadah khusus	108
Buat wadah khusus	109
Ubah proyek untuk menggunakan wadah khusus	110
Pertanyaan yang Sering Diajukan	113
Pemecahan masalah	113
Bekerja menggunakan Python	114
Membuat proyek Python	115
Memulai simulasi Python	116
Contoh klien Python	117
Pertanyaan yang Sering Diajukan	118
Pemecahan Masalah	118
Support untuk mesin lain	119
Unity	120
Unreal Engine	120
Penggunaan perangkat lunak berlisensi	120
Mengelola sumber daya dengan CloudFormation	120
Snapshot	123
Snapshot	124

Gunakan kasus untuk snapshot	124
SimSpace Weaver Konsol	125
AWS CLI	127
Pertanyaan yang Sering Diajukan	129
Perpesanan	130
Kasus penggunaan untuk pengiriman pesan	131
Menggunakan pesan APIs	131
Kapan menggunakan pesan	139
Kiat saat bekerja dengan perpesanan	142
Kesalahan pesan dan pemecahan masalah	144
Praktik terbaik	146
Mengatur alarm penagihan	146
Gunakan SimSpace Weaver Local	146
Hentikan simulasi yang tidak Anda butuhkan	147
Hapus sumber daya yang tidak Anda butuhkan	147
Memiliki cadangan	147
Keamanan	149
Perlindungan data	150
Enkripsi saat diam	151
Enkripsi saat bergerak	151
Privasi lalu lintas antar jaringan	152
Identity and Access Management	152
Audiens	152
Mengautentikasi dengan identitas	153
Mengelola akses menggunakan kebijakan	154
Bagaimana AWS SimSpace Weaver bekerja dengan IAM	156
Contoh kebijakan berbasis identitas	162
Izin yang SimSpace Weaver dibuat untuk Anda	166
Pencegahan "confused deputy" lintas layanan	168
Pemecahan masalah	171
Pencatatan dan pemantauan peristiwa keamanan	174
Validasi Kepatuhan	175
Ketahanan	175
Keamanan Infrastruktur	176
Model keamanan konektivitas jaringan	176
Konfigurasi dan analisis kerentanan	177

Praktik terbaik keamanan	177
Enkripsi komunikasi antara aplikasi Anda dan klien mereka	178
Cadangkan status simulasi Anda secara berkala	178
Pertahankan aplikasi Anda dan SDKs	178
Pencatatan log dan pemantauan	180
Log masuk CloudWatch	180
Mengakses log Anda SimSpace Weaver	180
SimSpace Weaver log	181
Pemantauan CloudWatch dengan	183
SimSpace Weaver metrik di tingkat akun	184
CloudTrail log	184
SimSpace Weaver informasi di CloudTrail	184
Memahami entri file SimSpace Weaver log	185
Titik akhir dan kuota layanan	188
Titik akhir layanan	188
Kuota layanan	189
Kuota pemesanan	192
Tarif jam	192
Kuota layanan untuk SimSpace Weaver Local	193
Pemecahan masalah	194
AssumeRoleAccessDenied	194
InvalidBucketName	196
ServiceQuotaExceededException	197
TooManyBuckets	197
Izin ditolak selama simulasi dimulai	198
Masalah yang berkaitan dengan waktu saat menggunakan Docker	198
Klien konsol gagal terhubung	199
Tidak <code>simspaceweaver</code> di AWS CLI	201
Referensi skema	202
Contoh skema lengkap	202
Format skema	204
Versi SDK	204
Sifat simulasi	205
Pekerja	207
Jam	208
Strategi partisi	208

Domain	210
Kendala penempatan	220
Referensi API	222
SimSpace Weaver versi	223
Versi terbaru	223
Cara menemukan versi Anda saat ini	223
Unduh versi terbaru	223
Memecahkan masalah unduhan SDK aplikasi	224
Instal versi terbaru	225
Versi layanan	225
1.17.0	240
Perubahan besar untuk 1.17.0	240
Perbarui proyek ke 1.17.0	241
Pertanyaan yang sering diajukan tentang versi 1.17.0	242
1.15.1	243
Perbarui proyek Python yang ada ke 1.15.1	243
Pemecahan masalah untuk versi 1.15.1	244
Pertanyaan yang sering diajukan tentang versi 1.15.1	244
Riwayat dokumen	245
Glosarium	253

Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS SimSpace Weaver. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses SimSpace Weaver konsol atau SimSpace Weaver sumber daya. Untuk informasi selengkapnya, lihat [AWS SimSpace Weaver akhir dukungan](#).

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

Apa itu AWS SimSpace Weaver?

AWS SimSpace Weaver adalah layanan yang dapat Anda gunakan untuk membangun dan menjalankan simulasi spasial skala besar di AWS Cloud. Misalnya, Anda dapat membuat simulasi kerumunan, lingkungan dunia nyata yang besar, dan pengalaman imersif dan interaktif.

Dengan SimSpace Weaver, Anda dapat mendistribusikan beban kerja simulasi di beberapa instans Amazon Elastic Compute Cloud (Amazon EC2). SimSpace Weaver menyebarkan AWS infrastruktur dasar untuk Anda, dan menangani manajemen data simulasi dan komunikasi jaringan antara EC2 instans Amazon yang menjalankan simulasi Anda.

Konsep kunci untuk SimSpace Weaver

Simulasi atau permainan dibatasi oleh komputer yang menjalankannya. Saat Anda menumbuhkan ukuran dan kompleksitas dunia virtual Anda, kinerja pemrosesan mulai menurun. Perhitungan memakan waktu lebih lama, sistem kehabisan memori, dan frame rate klien turun. Untuk simulasi yang tidak memerlukan kinerja waktu nyata, ini mungkin hanya mengganggu. Atau, ini bisa menjadi situasi kritis bisnis di mana peningkatan penundaan pemrosesan mengakibatkan peningkatan biaya. Jika simulasi atau game Anda membutuhkan kinerja waktu nyata, maka penurunan kinerja jelas menjadi masalah.

Solusi umum untuk simulasi yang mencapai batas kinerja adalah menyederhanakan simulasi. Game online dengan banyak pengguna sering mengatasi masalah penskalaan dengan membuat salinan dunia virtual mereka di server yang berbeda dan menyebarkan pengguna di seluruh mereka.

SimSpace Weaver memecahkan masalah penskalaan dengan membagi dunia virtual Anda secara spasial dan mendistribusikan potongan-potongan di sekelompok instance komputasi yang berjalan di AWS Cloud. Instance komputasi bekerja sama untuk memproses seluruh dunia simulasi secara paralel. Dunia simulasi Anda muncul sebagai ruang terintegrasi tunggal untuk semua yang ada di dalamnya, dan untuk semua klien yang terhubung dengannya. Anda tidak lagi harus menyederhanakan simulasi karena batas kinerja perangkat keras. Sebagai gantinya, Anda dapat menambahkan lebih banyak kapasitas komputasi di cloud.

Topik

- [Bagaimana cara SimSpace Weaver kerja](#)
- [Bagaimana Anda menggunakan SimSpace Weaver](#)

- [Skema simulasi](#)
- [Pekerja dan unit sumber daya](#)
- [Jam simulasi](#)
- [Partisi](#)
- [Kain Negara](#)
- [Entitas](#)
- [Aplikasi](#)

Bagaimana cara SimSpace Weaver kerja

Simulasi Anda terdiri dari dunia dengan benda-benda di dalamnya. Beberapa benda (seperti orang dan kendaraan) bergerak dan melakukan sesuatu. Objek lain (seperti pohon dan bangunan) bersifat statis. Dalam SimSpace Weaver, entitas adalah objek di dunia simulasi Anda.

Anda menentukan batas-batas dunia simulasi Anda dan membaginya menjadi grid. Alih-alih membuat logika simulasi yang beroperasi di seluruh grid, Anda membuat logika simulasi yang beroperasi pada satu sel grid. Di SimSpace Weaver, aplikasi spasial adalah program yang Anda tulis yang mengimplementasikan logika simulasi untuk sel kisi Anda. Ini termasuk logika untuk semua entitas dalam sel itu. Area kepemilikan aplikasi spasial adalah sel kisi yang dikendalikan oleh aplikasi spasial.

Note

Dalam SimSpace Weaver, istilah “aplikasi” dapat merujuk ke kode aplikasi atau instance kode yang sedang berjalan.



Dunia simulasi Anda dibagi menjadi grid

Anda membagi dunia simulasi Anda menjadi grid. Setiap aplikasi spasial mengimplementasikan logika simulasi untuk satu sel di kisi itu.

SimSpace Weaver menjalankan instance kode aplikasi spasial Anda untuk setiap sel grid Anda. Semua instance aplikasi spasial berjalan secara paralel. Pada dasarnya, SimSpace Weaver membagi simulasi keseluruhan Anda menjadi beberapa simulasi yang lebih kecil. Masing-masing simulasi yang lebih kecil menangani bagian dari dunia simulasi secara keseluruhan. SimSpace Weaver dapat mendistribusikan dan menjalankan simulasi yang lebih kecil ini di beberapa instans Amazon Elastic Compute Cloud (Amazon EC2) (disebut pekerja) di AWS Cloud. Seorang pekerja tunggal dapat menjalankan beberapa aplikasi spasial.

Entitas dapat bergerak melalui dunia simulasi. Jika entitas memasuki area kepemilikan aplikasi spasial lain (sel lain di grid), maka pemilik aplikasi spasial dari area baru akan mengambil alih kendali entitas. Jika simulasi Anda berjalan pada beberapa pekerja, maka entitas dapat berpindah dari kontrol aplikasi spasial pada satu pekerja ke aplikasi spasial pada pekerja yang berbeda. Ketika entitas pindah ke pekerja yang berbeda, SimSpace Weaver menangani komunikasi jaringan yang mendasarinya.

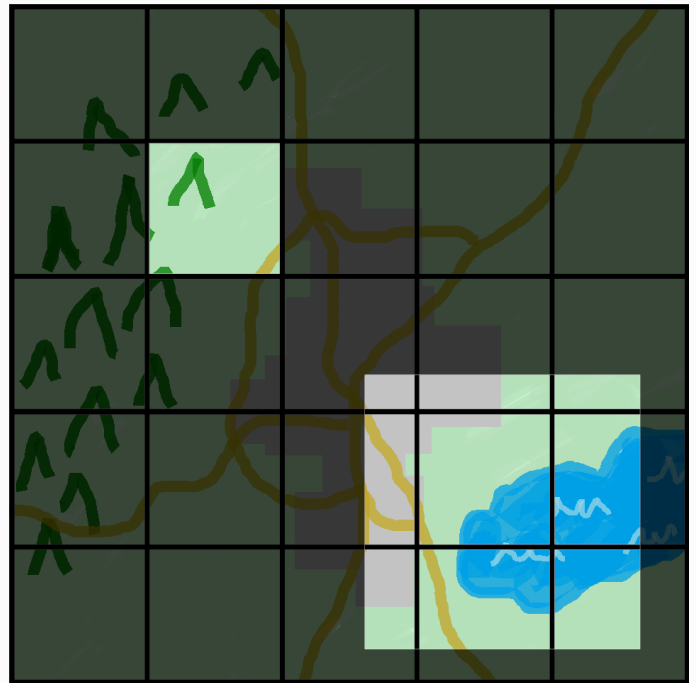
Langganan

Pandangan aplikasi spasial tentang dunia adalah area kepemilikannya sendiri. Untuk mengetahui apa yang terjadi di bagian lain dunia simulasi, aplikasi spasial membuat langganan. Area berlangganan adalah bagian dari keseluruhan area dunia simulasi. Area berlangganan dapat mencakup bagian dari beberapa area kepemilikan, termasuk area kepemilikan aplikasi spasial itu sendiri. SimSpace Weaver memberi tahu aplikasi spasial semua peristiwa entitas (misalnya, masuk, keluar, buat, perbarui, dan hapus) yang terjadi di dalam area langganan.



Tampilan aplikasi spasial tentang dunia

Pandangan aplikasi spasial tentang dunia adalah area kepemilikannya, yang merupakan satu sel di grid dunia.



Tampilan aplikasi spasial dengan area langganan tambahan

Aplikasi spasial menggunakan langganan untuk mencari tahu apa yang terjadi di bagian lain dunia simulasi. Area berlangganan dapat berisi beberapa sel kisi dan bagian sel.

Misalnya, aplikasi yang mensimulasikan entitas yang berinteraksi secara fisik mungkin perlu mengetahui tentang entitas tepat di luar batas spasial area kepemilikannya. Untuk mencapai hal ini, aplikasi dapat berlangganan ke area yang berbatasan dengan area kepemilikannya. Setelah membuat langganan, aplikasi menerima pemberitahuan tentang peristiwa entitas di area tersebut dan

dapat membaca entitas. Contoh lain adalah kendaraan otonom yang perlu melihat semua entitas 200 meter di depannya terlepas dari aplikasi apa yang memiliki area tersebut. Aplikasi untuk kendaraan dapat membuat langganan dengan filter sebagai kotak pembatas sejajar sumbu yang menutupi area (AABB) yang dapat dilihat.

Anda dapat membuat logika simulasi yang tidak bertanggung jawab untuk mengelola aspek spasial simulasi Anda. Aplikasi kustom adalah program yang dapat dieksekusi yang berjalan pada satu pekerja. Anda mengontrol siklus hidup (mulai dan berhenti) aplikasi kustom. Klien simulasi dapat terhubung ke aplikasi khusus untuk melihat atau berinteraksi dengan simulasi. Anda juga dapat membuat aplikasi layanan yang berjalan pada setiap pekerja. SimSpace Weaver memulai instance aplikasi layanan Anda pada setiap pekerja yang menjalankan simulasi Anda.

Aplikasi kustom dan aplikasi layanan membuat langganan untuk mempelajari tentang peristiwa entitas dan membaca entitas. Aplikasi ini tidak memiliki area kepemilikan karena tidak spasial. Menggunakan langganan adalah satu-satunya cara mereka dapat mengetahui apa yang terjadi di dunia simulasi.

Bagaimana Anda menggunakan SimSpace Weaver

Saat Anda menggunakan SimSpace Weaver, ini adalah langkah-langkah utama yang Anda ikuti:

1. Tulis dan buat C++ aplikasi yang mengintegrasikan SDK SimSpace Weaver aplikasi.
 - a. Aplikasi Anda melakukan panggilan API untuk berinteraksi dengan status simulasi.
2. Tulis klien yang melihat dan berinteraksi dengan simulasi Anda melalui beberapa aplikasi.
3. Konfigurasi simulasi Anda dalam file teks.
4. Unggah paket aplikasi dan konfigurasi simulasi Anda ke layanan.
5. Mulai simulasi Anda.
6. Mulai dan hentikan aplikasi kustom Anda sesuai kebutuhan.
7. Hubungkan klien ke aplikasi kustom atau layanan Anda untuk melihat atau berinteraksi dengan simulasi.
8. Periksa log simulasi Anda di Amazon CloudWatch Logs.
9. Hentikan simulasi Anda.
10. Bersihkan simulasi Anda.

Skema simulasi

Skema simulasi (atau skema) adalah file teks YAML berformat yang berisi informasi konfigurasi untuk simulasi Anda. SimSpace Weaver menggunakan skema Anda ketika memulai simulasi. Paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan mencakup skema untuk proyek sampel. Anda dapat menggunakan ini sebagai titik awal untuk skema Anda sendiri. Untuk informasi lebih lanjut tentang skema simulasi, lihat. [SimSpace Weaver referensi skema simulasi](#)

Pekerja dan unit sumber daya

Pekerja adalah instans Amazon EC2 yang menjalankan simulasi Anda. Anda menentukan jenis pekerja dalam skema simulasi Anda. SimSpace Weaver memetakan jenis pekerja Anda ke jenis instans Amazon EC2 tertentu yang digunakan layanan. SimSpace Weaver memulai dan menghentikan pekerja Anda untuk Anda, dan mengelola komunikasi jaringan antara para pekerja. SimSpace Weaver memulai satu set pekerja untuk setiap simulasi. Simulasi yang berbeda menggunakan pekerja yang berbeda.

Kapasitas komputasi (prosesor dan memori) yang tersedia pada pekerja dibagi menjadi unit logis yang disebut unit sumber daya komputasi (atau unit sumber daya). Unit sumber daya mewakili jumlah prosesor dan kapasitas memori yang tetap.

Note

Kami sebelumnya menyebut unit sumber daya komputasi sebagai slot. Anda mungkin masih melihat istilah sebelumnya ini dalam dokumentasi kami.

Jam simulasi

Setiap simulasi memiliki jam sendiri. Anda memulai dan menghentikan jam menggunakan panggilan API atau SimSpace Weaver konsol. Pembaruan simulasi hanya ketika jam berjalan. Semua operasi dalam simulasi terjadi dalam segmen waktu yang disebut kutu. Jam mengumumkan waktu mulai setiap centang untuk semua pekerja.

Tingkat jam (atau tingkat centang) adalah jumlah kutu per detik (hertz, atau Hz) yang diumumkan jam. Kecepatan jam yang diinginkan untuk simulasi adalah bagian dari skema simulasi. Semua operasi untuk tanda centang harus selesai sebelum centang berikutnya dimulai. Untuk alasan ini, clock rate efektif bisa lebih rendah dari clock rate yang diinginkan. Kecepatan clock efektif tidak akan lebih tinggi dari clock rate yang diinginkan.

Partisi

Partisi adalah segmen memori bersama pada pekerja. Setiap partisi memegang bagian dari data status simulasi.

Partisi untuk aplikasi spasial (juga disebut partisi aplikasi spasial atau partisi spasial) berisi semua entitas di area kepemilikan aplikasi spasial. SimSpace Weaver menempatkan entitas dalam partisi aplikasi spasial berdasarkan lokasi spasial setiap entitas. Ini berarti SimSpace Weaver mencoba menempatkan entitas yang secara spasial berdekatan satu sama lain pada pekerja yang sama. Ini meminimalkan jumlah pengetahuan yang dibutuhkan aplikasi dari entitas yang tidak dimilikinya untuk mensimulasikan entitas yang dimilikinya.

Kain Negara

State Fabric adalah sistem memori bersama (kumpulan semua partisi) pada semua pekerja. Ini menyimpan semua data status untuk simulasi Anda.

State Fabric menggunakan format biner kustom yang menggambarkan entitas sebagai kumpulan data awal dan log pembaruan, untuk setiap bidang data entitas tersebut. Dengan format ini, Anda dapat mengakses status entitas pada titik sebelumnya dalam waktu simulasi dan memetakannya kembali ke titik dalam waktu dunia nyata. Buffer memiliki ukuran yang terbatas, dan tidak mungkin untuk kembali ke masa lalu di luar apa yang ada di buffer. SimSpace Weaver menggunakan pointer ke offset saat ini di log pembaruan untuk setiap bidang, dan memperbarui pointer sebagai bagian dari pembaruan bidang. SimSpace Weaver memetakan log pembaruan ini ke dalam ruang proses aplikasi menggunakan memori bersama.

Format objek ini menghasilkan overhead rendah dan tidak ada biaya serialisasi. SimSpace Weaver juga menggunakan format objek ini untuk mengurai dan mengidentifikasi bidang indeks (seperti posisi entitas).

Entitas

Entitas adalah blok bangunan terkecil dari data dalam simulasi Anda. Contoh entitas termasuk aktor (seperti orang dan kendaraan) dan objek statis (seperti bangunan dan rintangan). Entitas memiliki properti (seperti posisi dan orientasi) yang dapat Anda simpan sebagai data persisten SimSpace Weaver. Entitas ada di dalam partisi.

Aplikasi

SimSpace Weaver Aplikasi adalah perangkat lunak yang Anda tulis yang berisi logika khusus yang menjalankan setiap centang simulasi. Tujuan dari sebagian besar aplikasi adalah untuk memperbarui entitas saat simulasi berjalan. Aplikasi Anda memanggil APIs SDK SimSpace Weaver aplikasi untuk melakukan tindakan (seperti membaca dan memperbarui) pada entitas dalam simulasi Anda.

Anda mengemas aplikasi dan sumber daya yang diperlukan (seperti pustaka) sebagai file.zip dan mengunggahnya. SimSpace Weaver Aplikasi berjalan di container Docker pada pekerja. SimSpace Weaver mengalokasikan setiap aplikasi sejumlah unit sumber daya tetap pada pekerja.

SimSpace Weaver menetapkan kepemilikan satu (dan hanya satu) partisi ke setiap aplikasi. Aplikasi dan partisinya terletak pada pekerja yang sama. Setiap partisi hanya memiliki satu pemilik aplikasi. Aplikasi dapat membuat, membaca, memperbarui, dan menghapus entitas di partisinya. Aplikasi memiliki semua entitas di partisinya.

Ada tiga jenis aplikasi: aplikasi spasial, aplikasi khusus, dan aplikasi layanan. Mereka berbeda dengan kasus penggunaan dan siklus hidup.

Note

Dalam SimSpace Weaver, istilah “app” dapat merujuk ke kode untuk aplikasi atau instance kode yang sedang berjalan.

Aplikasi spasial

Aplikasi spasial memperbarui status entitas yang ada secara spasial dalam simulasi Anda. Misalnya, Anda dapat menentukan `Physics` aplikasi yang bertanggung jawab untuk memindahkan dan menabrak entitas untuk setiap tick berdasarkan kecepatan, bentuk, dan ukurannya. Dalam hal ini, SimSpace Weaver jalankan beberapa instance `Physics` aplikasi secara paralel untuk menangani ukuran beban kerja.

SimSpace Weaver mengelola siklus hidup aplikasi spasial. Anda menentukan pengaturan partisi aplikasi spasial dalam skema simulasi Anda. Saat meluncurkan simulasi, SimSpace Weaver mulai aplikasi spasial untuk setiap partisi aplikasi spasial. Saat Anda menghentikan simulasi, SimSpace Weaver matikan aplikasi spasial Anda.

Jenis aplikasi lain dapat membuat entitas, tetapi hanya aplikasi spasial yang dapat memperbarui entitas. Jenis aplikasi lain harus mentransfer entitas yang mereka buat ke domain spasial. SimSpace

Weaver menggunakan lokasi spasial entitas untuk memindahkan entitas ke partisi aplikasi spasial. Ini mengalihkan kepemilikan entitas ke aplikasi spasial.

Aplikasi kustom

Anda menggunakan aplikasi khusus untuk berinteraksi dengan simulasi Anda. Aplikasi kustom membaca data entitas menggunakan langganan. Aplikasi kustom dapat membuat entitas. Namun, aplikasi harus mentransfer entitas ke aplikasi spasial untuk menyertakan entitas dalam simulasi dan memperbaruinya. Anda dapat SimSpace Weaver menetapkan titik akhir jaringan ke aplikasi khusus. Klien simulasi dapat terhubung ke titik akhir jaringan untuk berinteraksi dengan simulasi. Anda menentukan aplikasi kustom Anda dalam skema simulasi Anda, tetapi Anda bertanggung jawab untuk memulai dan menghentikannya (menggunakan panggilan SimSpace Weaver API). Setelah Anda memulai instance aplikasi kustom pada pekerja, SimSpace Weaver tidak mentransfer instance ke pekerja lain.

Aplikasi layanan

Anda dapat menggunakan aplikasi layanan saat memerlukan proses hanya-baca yang berjalan pada setiap pekerja. Misalnya, Anda dapat menggunakan aplikasi layanan jika Anda memiliki simulasi besar dan Anda memerlukan klien tampilan yang bergerak melalui simulasi dan hanya menampilkan entitas yang terlihat kepada pengguna. Dalam hal ini, satu instance aplikasi kustom tidak dapat memproses semua entitas dalam simulasi. Anda dapat mengonfigurasi aplikasi layanan untuk diluncurkan pada setiap pekerja. Masing-masing aplikasi layanan ini kemudian dapat memfilter entitas pada pekerja yang ditugaskan dan hanya mengirim entitas yang relevan ke klien yang terhubung. Klien tampilan Anda kemudian dapat terhubung ke aplikasi layanan yang berbeda saat bergerak melalui ruang simulasi. Anda mengonfigurasi aplikasi layanan dalam skema simulasi Anda. SimSpace Weaver memulai dan menghentikan aplikasi layanan Anda untuk Anda.

Ringkasan aplikasi

Tabel berikut merangkum karakteristik berbagai jenis SimSpace Weaver aplikasi.

	Aplikasi spasial	Aplikasi kustom	Aplikasi layanan
Baca entitas	Ya	Ya	Ya
Perbarui entitas	Ya	Tidak	Tidak
Buat entitas	Ya	Ya*	Ya*

	Aplikasi spasial	Aplikasi kustom	Aplikasi layanan
Siklus hidup	Dikelola (SimSpace Weaver mengendalikannya.)	Tidak dikelola (Anda mengendalikannya.)	Dikelola (SimSpace Weaver mengendalikannya.)
Metode mulai	SimSpace Weaver memulai satu instance aplikasi untuk setiap partisi spasial, seperti yang ditentukan dalam skema Anda.	Anda memulai setiap instance aplikasi.	SimSpace Weaver memulai satu atau beberapa instance aplikasi pada setiap pekerja, seperti yang ditentukan dalam skema Anda.
Klien dapat terhubung	Tidak	Ya	Ya

* Saat aplikasi kustom atau aplikasi layanan membuat entitas, aplikasi harus mentransfer kepemilikan entitas ke aplikasi spasial sehingga aplikasi spasial dapat memperbarui status entitas.

Domain

SimSpace Weaver Domain adalah kumpulan instance aplikasi yang menjalankan kode aplikasi yang dapat dieksekusi yang sama dan memiliki opsi dan perintah peluncuran yang sama. Kami merujuk ke domain berdasarkan jenis aplikasi yang dikandungnya: domain spasial, domain khusus, dan domain layanan. Anda mengonfigurasi aplikasi Anda dalam domain.

Langganan dan replikasi

Aplikasi membuat langganan ke wilayah spasial untuk mempelajari peristiwa entitas (misalnya, masuk, keluar, buat, perbarui, dan hapus) di wilayah tersebut. Aplikasi memproses peristiwa entitas dari langganan sebelum membaca data untuk entitas dalam partisi yang tidak dimilikinya.

Partisi dapat ada pada pekerja yang sama dengan aplikasi (ini disebut partisi lokal), tetapi aplikasi lain dapat memiliki partisi. Partisi juga dapat ada pada pekerja yang berbeda (ini disebut partisi jarak jauh). Jika langganan ke partisi jarak jauh, pekerja membuat salinan lokal partisi jarak jauh melalui proses yang disebut replikasi. Pekerja kemudian membaca salinan lokal (partisi jarak jauh yang direplikasi). Jika aplikasi lain pada pekerja perlu membaca dari partisi itu pada tanda centang yang sama, maka pekerja akan membaca salinan lokal yang sama.

Contoh kasus penggunaan untuk SimSpace Weaver

Anda dapat menggunakan SimSpace Weaver untuk model berbasis agen dan simulasi langkah waktu diskrit dengan komponen spasial.

Buat simulasi kerumunan besar

Anda dapat menggunakan SimSpace Weaver untuk mensimulasikan kerumunan di lingkungan dunia nyata. SimSpace Weaver memungkinkan Anda untuk menskalakan simulasi Anda ke jutaan objek dinamis dengan perilaku mereka sendiri.

Ciptakan lingkungan berskala kota

Gunakan SimSpace Weaver untuk membuat kembar digital dari seluruh kota. Buat simulasi untuk perencanaan kota, merancang perutean lalu lintas, dan merencanakan respons bahaya lingkungan. Anda dapat menggunakan sumber data geospasial Anda sendiri sebagai blok bangunan untuk lingkungan Anda.

Ciptakan pengalaman yang imersif dan interaktif

Buat pengalaman simulasi yang dapat diikuti dan berinteraksi dengan banyak pengguna. Gunakan alat pengembangan populer seperti Unreal Engine dan Unity untuk membangun dunia virtual 3 dimensi (3D). Sesuaikan pengalaman 3D Anda dengan konten dan perilaku Anda sendiri.

AWS SimSpace Weaver Akhir dari Dukungan

Setelah mempertimbangkan dengan cermat, kami memutuskan untuk mengakhiri dukungan untuk AWS SimSpace Weaver, efektif 20 Mei 2026. AWS SimSpace Weaver tidak akan lagi menerima pelanggan baru mulai 20 Mei 2025. Sebagai pelanggan lama dengan akun yang mendaftar untuk layanan sebelum 20 Mei 2025, Anda dapat terus menggunakan AWS SimSpace Weaver fitur. Setelah 20 Mei 2026, Anda tidak akan dapat lagi menggunakannya AWS SimSpace Weaver.

[Untuk informasi lebih lanjut tentang transisi ke AWS Batch untuk membantu menjalankan simulasi kontainer, lihat posting blog ini.](#)

Menyiapkan untuk SimSpace Weaver

Untuk menyiapkan penggunaan SimSpace Weaver untuk pertama kalinya, Anda harus mengatur lingkungan Akun AWS dan lingkungan lokal Anda. Ketika Anda menyelesaikan tugas-tugas ini, Anda akan siap untuk [memulai tutorial](#).

Menyiapkan tugas

1. [Siapkan Akun AWS untuk Anda gunakan SimSpace Weaver](#).
2. [Siapkan lingkungan lokal Anda untuk SimSpace Weaver](#).

Siapkan Akun AWS untuk Anda gunakan SimSpace Weaver

Selesaikan tugas-tugas berikut Akun AWS untuk mengatur penggunaan Anda SimSpace Weaver.

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/pendaftaran>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirim Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk <https://aws.amazon.com/ke/> dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftarkan Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [Konsol Manajemen AWS](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

Untuk bantuan masuk dengan menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) di AWS Sign-In Panduan Pengguna.

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

Untuk petunjuknya, lihat [Membuat set izin](#) di Panduan AWS IAM Identity Center Pengguna.

2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

Untuk petunjuk, lihat [Menambahkan grup](#) di Panduan AWS IAM Identity Center Pengguna.

Tambahkan izin untuk digunakan SimSpace Weaver

Untuk memberikan akses dan menambahkan izin bagi pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

Buat rangkaian izin. Ikuti instruksi di [Buat rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

Buat peran untuk federasi identitas. Ikuti instruksi dalam [Buat peran untuk penyedia identitas pihak ketiga \(federasi\)](#) dalam Panduan Pengguna IAM.

- Pengguna IAM:

- Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam [Buat peran untuk pengguna IAM](#) dalam Panduan Pengguna IAM.
- (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam [Menambahkan izin ke pengguna \(konsol\)](#) dalam Panduan Pengguna IAM.

Example Kebijakan IAM untuk memberikan izin penggunaan SimSpace Weaver

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateAndRunSimulations",
      "Effect": "Allow",
      "Action": [
```

```

        "simspaceweaver:*",
        "iam:GetRole",
        "iam:ListRoles",
        "iam:CreateRole",
        "iam>DeleteRole",
        "iam:UpdateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "iam:GetRolePolicy",
        "iam>DeleteRolePolicy",
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy",
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutEncryptionConfiguration",
        "s3>DeleteBucket",
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation:DescribeStacks"
    ],
    "Resource": "*"
},
{
    "Sid": "PassAppRoleToSimSpaceWeaver",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": "simspaceweaver.amazonaws.com"
        }
    }
}
]
}

```

Siapkan lingkungan lokal Anda untuk SimSpace Weaver

SimSpace Weaver simulasi berjalan dalam kontainerisasi Amazon Linux 2 (AL2) lingkungan. Anda harus memiliki AL2 lingkungan untuk mengkompilasi dan menautkan aplikasi Anda dengan

SDK SimSpace Weaver aplikasi. Lingkungan pembangunan lokal standar adalah AL2 wadah di Docker. Jika Anda memilih untuk tidak menggunakan Docker, kami memberikan instruksi alternatif untuk menjalankan AL2 lingkungan di Windows Subsystem for Linux (WSL). Anda juga dapat menggunakan metode Anda sendiri untuk menciptakan AL2 lingkungan lokal. Untuk beberapa cara tambahan untuk menjalankan AL2 secara lokal, lihat [EC2 dokumentasi Amazon](#).

Important

Docker on Microsoft Windows adalah lingkungan pengembangan standar. Untuk kenyamanan Anda, kami menyarankan cara lain untuk mengatur lingkungan pengembangan lokal Anda, tetapi mereka tidak standar dan tidak didukung.

Topik

- [Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 \(AL2\) di Docker](#)
- [Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 \(AL2\) di Windows Subsystem for Linux \(WSL\)](#)

Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 (AL2) di Docker

Bagian ini memberikan petunjuk untuk menyiapkan zip SimSpace Weaver distribusi lokal Anda dengan AL2 lingkungan di Docker. Untuk instruksi untuk mengatur dengan AL2 di Windows Subsystem for Linux (WSL), lihat [Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 \(AL2\) di Windows Subsystem for Linux \(WSL\)](#).

Persyaratan

- Microsoft Windows 10 atau lebih tinggi, atau sistem Linux yang kompatibel
- [Microsoft Visual Studio 2019](#) atau kemudian, dengan [Desktop development with C++](#) beban kerja terpasang
- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)

- [Python 3.9](#)

Untuk mengatur zip SimSpace Weaver distribusi dengan AL2 in Docker

1. Jika Anda belum mengonfigurasi AWS kredensialnya AWS CLI, ikuti petunjuk berikut: [Mengonfigurasi AWS CLI](#).
2. [Unduh paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan](#). Ini berisi yang berikut:
 - Binari dan pustaka untuk pengembangan aplikasi SimSpace Weaver
 - Skrip pembantu yang mengotomatiskan bagian dari alur kerja pengembangan
 - Contoh aplikasi yang menunjukkan SimSpace Weaver konsep
3. Buka zip file ke *sdk-folder* pilihan Anda.
4. Pergi ke *sdk-folder*.
5. Masukkan perintah berikut untuk menginstal paket Python yang diperlukan:

```
pip install -r PackagingTools/python_requirements.txt
```

6. Masukkan perintah berikut untuk mengatur SimSpace Weaver distribusi dengan gambar Docker.

```
python setup.py
```

Perintah ini melakukan hal berikut:

- Membuat image AL2 docker dengan semua persyaratan untuk membangun SimSpace Weaver proyek yang diinstal.
- Menciptakan CloudFormation sumber daya yang diperlukan untuk meluncurkan simulasi.
 - Template CloudFormation tumpukan sampel dapat ditemukan di *sdk-folder/PackagingTools/sample-stack-template.yaml*
- Mengkonfigurasi proyek sampel yang disediakan dengan jalur yang benar untuk sistem lokal Anda.

Pemecahan Masalah

- Docker tampak macet

- Jika output konsol tampak macet setelah perintah Docker dipanggil, coba mulai ulang mesin Docker. Jika itu tidak berhasil, restart komputer Anda.

Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 (AL2) di Windows Subsystem for Linux (WSL)

Bagian ini memberikan petunjuk untuk mengatur zip SimSpace Weaver distribusi Anda dengan AL2 lingkungan di Windows Subsystem for Linux (WSL). Untuk instruksi untuk mengatur AL2 Docker, lihat [Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 \(AL2\) di Docker](#).

Important

Bagian ini menjelaskan solusi yang menggunakan versi AL2 yang tidak dimiliki, dikembangkan, atau didukung oleh Amazon. Solusi ini disediakan untuk kenyamanan Anda saja, jika Anda memilih untuk tidak menggunakan Docker. Amazon dan tidak AWS bertanggung jawab jika Anda memilih untuk menggunakan solusi ini.

Persyaratan

- [Hyper-V on Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- AL2 Distribusi open source pihak ketiga untuk WSL ([download versi 2.0.20200722.0-update.2](#)) ([lihat instruksi](#))

Important

Kami WSL petunjuk menggunakan versi [2.0.20200722.0-update.2](#) dari distribusi untuk AL2 WSL. Anda mungkin mengalami kesalahan jika Anda menggunakan versi lain.

Untuk mengatur zip SimSpace Weaver distribution dengan in AL2 WSL

1. Pada prompt perintah Windows, mulai AL2 lingkungan Anda WSL.

```
ws1 -d Amazon2
```

⚠ Important

Saat Anda berlari di WSL, sertakan `--al2` opsi saat menjalankan salah satu skrip pembantu `quick-start.py` Python yang terletak di `sdky-folder/Samples/sample-name/tools/cloud/quick-start.py`

2. Pada prompt shell Linux, perbarui manajer paket yum Anda.

```
yum update -y
```

⚠ Important

Jika langkah ini habis waktu, Anda mungkin perlu beralih ke WSL1 dan coba lagi prosedur ini. Keluar dari Anda WSL AL2 sesi dan masukkan yang berikut ini di prompt perintah Windows Anda:

```
wsl --set-version Amazon2 1
```

3. Instal alat unzip.

```
yum install -y unzip
```

4. Hapus semua AWS CLI yang yum diinstal. Coba kedua perintah berikut jika Anda tidak yakin apakah yum menginstal file AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Buat direktori sementara dan pergi ke sana.

```
mkdir ~/temp  
cd ~/temp
```

6. Unduh dan instal AWS CLI:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

```
unzip awscliv2.zip
./aws/install
```

7. Anda dapat menghapus direktori sementara.

```
cd ~
rm -rf temp
```

8. Mulai ulang sesi shell untuk memperbarui jalur di lingkungan.

```
exec
```

9. Konfigurasi AWS kredensial Anda untuk lingkungan AWS CLI Anda AL2 . Untuk informasi lebih lanjut, lihat [Mengonfigurasi AWS CLI](#). Jika Anda menggunakan AWS IAM Identity Center, lihat [Mengkonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center](#) dalam Panduan AWS Command Line Interface Pengguna.

```
aws configure
```

10. Instal Git.

```
yum install -y git
```

11. Instal wget.

```
yum install -y wget
```

12. Buat folder untuk SDK SimSpace Weaver aplikasi.

```
mkdir sdk-folder
```

13. Buka folder SDK Anda.

```
cd sdk-folder
```

14. Unduh paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan. Ini berisi yang berikut:

- Binari dan pustaka untuk pengembangan aplikasi SimSpace Weaver
- Skrip pembantu yang mengotomatiskan bagian dari alur kerja pengembangan
- Contoh aplikasi yang menunjukkan SimSpace Weaver konsep

```
wget https://artifacts.simspaceweaver.us-east-2.amazonaws.com/latest/  
SimSpaceWeaverAppSdkDistributable.zip
```

15. Buka filenya.

```
unzip *.zip
```

16. Jalankan skrip pengaturan WSL.

```
source ./setup-wsl-distro.sh
```

17. Masukkan perintah berikut untuk menginstal paket Python yang diperlukan:


```
pip install -r PackagingTools/python_requirements.txt
```

18. Jalankan skrip penyiapan zip SimSpace Weaver distribusi:

```
python setup.py --samples --cloudformation
```

Perintah ini melakukan hal berikut:

- Menciptakan CloudFormation sumber daya yang diperlukan untuk meluncurkan simulasi.
 - Template CloudFormation tumpukan sampel dapat ditemukan di *sdk-folder/* `PackagingTools/sample-stack-template.yaml`
- Mengkonfigurasi proyek sampel yang disediakan dengan jalur yang benar untuk sistem lokal Anda.

 Note

Anda hanya perlu melakukan ini satu kali untuk AL2 lingkungan Anda di WSL.

Penggunaan perangkat lunak berlisensi dengan AWS SimSpace Weaver

AWS SimSpace Weaver memungkinkan Anda membangun simulasi dengan mesin dan konten simulasi pilihan Anda. Sehubungan dengan penggunaan Anda SimSpace Weaver, Anda bertanggung jawab untuk memperoleh, memelihara, dan mematuhi ketentuan lisensi perangkat lunak atau konten apa pun yang Anda gunakan dalam simulasi Anda. Verifikasi perjanjian lisensi Anda memungkinkan Anda untuk menyebarkan perangkat lunak dan konten Anda di lingkungan yang dihosting virtual.

Memulai dengan SimSpace Weaver

Bagian ini menyediakan tutorial untuk membantu Anda memulai SimSpace Weaver. Tutorial ini memperkenalkan Anda pada alur kerja umum untuk membangun simulasi dengan SimSpace Weaver. Tutorial ini menunjukkan cara membuat, menyebarkan, dan menjalankan simulasi di SimSpace Weaver. Kami menyarankan Anda memulai dengan tutorial mulai cepat untuk menjalankan simulasi dalam hitungan menit. Pergi melalui tutorial lain setelah itu untuk mempelajari lebih lanjut.

Tutorial ini menggunakan contoh aplikasi (`PathfindingSample`) yang disertakan dalam file SDK SimSpace Weaver aplikasi .zip yang Anda unduh selama prosedur [penyiapan](#). Aplikasi sampel menunjukkan konsep yang dibagikan semua SimSpace Weaver simulasi, termasuk partisi spasial, handoff entitas lintas partisi, aplikasi, dan langganan.

Dalam tutorial, Anda akan membuat simulasi dengan empat partisi spasial. Instance terpisah dari aplikasi `PathfindingSample` spasial mengelola setiap partisi individu. Aplikasi spasial membuat entitas di partisi mereka sendiri. Entitas pindah ke posisi tertentu di dunia simulasi, menghindari rintangan saat mereka bergerak. Anda dapat menggunakan aplikasi klien terpisah (termasuk dalam SDK SimSpace Weaver aplikasi) untuk melihat simulasi.

Topik

- [Tutorial mulai cepat untuk SimSpace Weaver](#)
- [Tutorial terperinci: Pelajari detailnya saat membangun aplikasi sampel](#)

Tutorial mulai cepat untuk SimSpace Weaver

Tutorial ini memandu Anda melalui proses untuk membangun dan menjalankan simulasi SimSpace Weaver dalam hitungan menit. Kami menyarankan Anda memulai dengan tutorial ini dan kemudian pergi melalui [tutorial rinci](#) sesudahnya.

Persyaratan

Sebelum Anda mulai, pastikan Anda menyelesaikan langkah-langkahnya [Menyiapkan untuk SimSpace Weaver](#).

Note

Skrip yang digunakan di sini disediakan untuk kenyamanan Anda dan TIDAK diperlukan. Lihat [tutorial terperinci](#) untuk bagaimana langkah-langkah ini dapat dilakukan secara manual.

Langkah 1: Aktifkan logging (opsional)

Untuk mengaktifkan logging

1. Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools
```

2. Buka file skema di editor teks:

```
pathfinding-single-worker-schema.yaml
```

3. Temukan `simulation_properties`: bagian di awal file:

```
simulation_properties:  
  default_entity_index_key_type: "Vector3<f32>"
```

4. Masukkan 2 baris berikut setelah baris `simulation_properties::`

```
  log_destination_service: "logs"  
  log_destination_resource_name: "MySimulationLogs"
```

5. Konfirmasikan bahwa `simulation_properties`: bagian Anda sama dengan yang berikut:

```
simulation_properties:  
  log_destination_service: "logs"  
  log_destination_resource_name: "MySimulationLogs"  
  default_entity_index_key_type: "Vector3<f32>"
```

6. Simpan file dan keluar dari editor teks Anda.

Langkah 2: Mulai cepat dengan klien konsol (opsi 1)

Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Jalankan salah satu perintah berikut ini:

- buruh pelabuhan: `python quick-start.py --consoleclient`
- WSL: `python quick-start.py --consoleclient --a12`

Secara default, ini akan meluncurkan simulasi dengan partisi tunggal pada satu pekerja. Konfigurasi lain dapat diluncurkan dengan meneruskan `--schema {file name}.yaml` dari `/Samples/PathfindingSample/tools/` folder.

Note

Lihat [Tutorial terperinci: Pelajari detailnya saat membangun aplikasi sampel](#) penjelasan mendalam tentang apa yang dilakukan skrip ini.

Langkah 2: Mulai cepat dengan klien Unreal Engine (opsi 2)

Lihat [Meluncurkan klien tampilan Unreal Engine](#).

Hentikan dan hapus simulasi Anda

Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Temukan nama-nama simulasi Anda:

```
aws simspaceweaver list-simulations
```

Hentikan dan hapus simulasi

```
python stop-and-delete.py --simulation simulation-name
```

Pemecahan Masalah

- `FileNotFoundError`: `cmake`

```
subprocess.run('cmake')
...
FileNotFoundError: The system cannot find the file specified
```

- Resolusi: Script tidak dapat menemukan perintahcmake. Pastikan Anda memiliki CMake versi minimum yang direkomendasikan diinstal, dan itu dapat dipanggil dengan cmake perintah di PATH. Gunakan perintah cmake -version untuk memverifikasi.
- ImportError: Pemuatan DLL gagal saat mengimpor libweaver_app_sdk_python_v1: Modul yang ditentukan tidak dapat ditemukan.
 - Resolusi: Kesalahan ini terjadi ketika Python 3.9 tidak digunakan untuk meluncurkan Weaver Python SDK. Harap pastikan versi python yang terkait dengan perintah “python” adalah Python 3.9. Anda dapat memeriksa dengan menjalankan python --version perintah.
- Skrip mulai cepat tampak macet setelah memulai Docker Build.
 - Resolusi: Terkadang Docker membutuhkan beberapa menit untuk pemanasan. Jika masalah ini berlanjut selama lebih dari ~ 5 menit, silakan mulai ulang Docker atau sistem Anda.
- target_compile_features tidak ada fitur yang diketahui untuk kompiler CXX “GNU”:
 - Resolusi: Bersihkan cache Docker Anda, hapus gambar Docker weaverappbuilder, hapus artefak build proyek Anda, dan jalankan kembali. setup.py Ini akan mengatur ulang lingkungan Docker Anda dan menyelesaikan kesalahan.

Tutorial terperinci: Pelajari detailnya saat membangun aplikasi sampel

[Tutorial mulai cepat](#) membahas cara membangun, memulai, menghentikan, dan menghapus simulasi sampel menggunakan quick-start.py dan stop-and-delete.py. Tutorial ini akan membahas secara rinci bagaimana skrip ini bekerja, dan parameter tambahan skrip ini dapat mengambil untuk memaksimalkan fleksibilitas untuk simulasi Weaver kustom.

Persyaratan

Sebelum Anda mulai, pastikan Anda menyelesaikan langkah-langkahnya [Menyiapkan untuk SimSpace Weaver](#).

Langkah 1: Aktifkan logging (opsional)

Untuk mengaktifkan logging

1. Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools
```

2. Buka file skema di editor teks:

```
pathfinding-single-worker-schema.yaml
```

3. Temukan `simulation_properties`: bagian di awal file:

```
simulation_properties:  
  default_entity_index_key_type: "Vector3<f32>"
```

4. Masukkan 2 baris berikut setelah `simulation_properties::`

```
log_destination_service: "logs"  
log_destination_resource_name: "MySimulationLogs"
```

5. Konfirmasikan bahwa `simulation_properties`: bagian Anda sama dengan yang berikut:

```
simulation_properties:  
  log_destination_service: "logs"  
  log_destination_resource_name: "MySimulationLogs"  
  default_entity_index_key_type: "Vector3<f32>"
```

6. Simpan file dan keluar dari editor teks Anda.

Langkah 2: Mulai simulasi Anda

Seperti yang ditunjukkan dalam [tutorial mulai cepat](#), langkah paling dasar untuk meluncurkan simulasi sampel adalah:

1. Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

2. Jalankan salah satu perintah berikut:

- buruh pelabuhan: `python quick-start.py`
- WSL: `python quick-start.py --al2`

Skrip ini mengotomatiskan perintah terminal umum, yang semuanya dapat dijalankan secara manual menggunakan file. AWS CLI Langkah-langkah ini adalah:

1. Unggah skema Weaver ke S3.

- SimSpace Weaver menggunakan skema untuk mengkonfigurasi simulasi Anda. Skema adalah file teks biasa berformat YAML. Untuk informasi lebih lanjut, lihat [Mengkonfigurasi simulasi Anda](#)

2. Bangun dan unggah wadah khusus (opsional).

- Jika skema Anda mendefinisikan wadah khusus, skrip mulai cepat akan membuat gambar docker dan mengunggahnya ke Amazon ECR. Untuk informasi selengkapnya, lihat [Wadah khusus](#). Lihat `PythonBubblesSample` skema untuk contoh fitur ini.

3. Bangun proyek.

- `quick-start.py` memanggil `build_project` fungsi yang didefinisikan dalam `build.py`. Langkah ini akan bervariasi tergantung pada proyek. Untuk `PathfindingSample`, CMake digunakan. Perintah CMake and Docker yang dapat ditemukan di `build.py`.

4. Unggah artefak build ke S3.

- Anda dapat memeriksa bucket S3 Anda untuk memastikan bahwa semua unggahan berhasil. Untuk informasi selengkapnya tentang penggunaan Amazon S3, lihat [Membuat, mengonfigurasi, dan bekerja dengan bucket Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).
- Contoh ritsleting aplikasi dan bucket S3 menggunakan format nama berikut:
 - `weaver-sample-bucket-account-number-region`
 - Aplikasi spasial: `ProjectNameSpatial.zip`
 - Lihat aplikasi (kustom): `ProjectNameView.zip`

5. Mulai Simulasi.

- Ini adalah pembungkus di sekitar `aws simspaceweaver start-simulation` AWS CLI panggilan. Untuk informasi selengkapnya, lihat [Referensi AWS CLI Perintah](#) untuk SimSpace Weaver.

- Script akan loop sampai status simulasi adalah salah satu **STARTED** atau **FAILED**. Diperlukan beberapa menit untuk memulai simulasi.
6. Dapatkan detail simulasi.
- `DescribeSimulationAPI` memberikan detail tentang simulasi Anda, termasuk statusnya. Simulasi dapat berada di salah satu keadaan berikut:

Status siklus hidup simulasi

1. **STARTING**— Keadaan awal setelah Anda menelepon `StartSimulation`
2. **STARTED**— semua aplikasi spasial diluncurkan dan sehat
3. **STOPPING**— Keadaan awal setelah Anda menelepon `StopSimulation`
4. **STOPPED**— Semua sumber daya komputasi dihentikan
5. **DELETING**— Keadaan awal setelah Anda menelepon `DeleteSimulation`
6. **DELETED**— Semua sumber daya yang ditugaskan untuk simulasi dihapus
7. **FAILED**— Simulasi memiliki kritis error/failure dan berhenti
8. **SNAPSHOT_IN_PROGRESS**— Sebuah [snapshot](#) sedang berlangsung

Untuk mendapatkan detail simulasi Anda

1. Panggil `ListSimulations API`.

```
aws simspaceweaver list-simulations
```

Skrip harus menampilkan detail tentang setiap simulasi Anda, mirip dengan yang berikut:

```
{
  "Status": "STARTED",
  "CreationTime": 1664921418.09,
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15",
  "TargetStatus": "STARTED"
}
```

2. Hubungi `DescribeSimulation` untuk mendapatkan detail simulasi Anda. Gantikan *simulation-name* dengan Name simulasi Anda dari output dari langkah sebelumnya.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Skrip harus menampilkan detail lebih lanjut tentang simulasi yang Anda tentukan, mirip dengan yang berikut ini:

```
{
  "Status": "STARTED",
  "CreationTime": 1664921418.09,
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15",
  "TargetStatus": "STARTED"
}
```

7. Mulai aplikasi khusus.

- SimSpace Weaver tidak mengelola siklus hidup aplikasi khusus. Anda harus memulai aplikasi kustom Anda. Praktik terbaik adalah memulai aplikasi khusus sebelum memulai jam simulasi, tetapi Anda dapat memulai aplikasi khusus setelah memulai jam.

Anda dapat memanggil StartApp API untuk memulai aplikasi kustom Anda.

```
aws simspaceweaver start-app --simulation simulation-name --name app-name --domain domain-name
```

Panggilan StartApp API akan membuat dan memulai instance baru dari aplikasi kustom menggunakan nama yang Anda berikan. Jika Anda memberikan nama aplikasi yang sudah ada maka Anda akan menerima kesalahan. Jika Anda ingin memulai ulang aplikasi tertentu (instance), Anda harus terlebih dahulu menghentikan aplikasi itu dan menghapusnya.

Note

Status simulasi Anda harus STARTED sebelum Anda dapat memulai aplikasi khusus.

Aplikasi sampel menyediakan aplikasi ViewApp khusus untuk melihat simulasi Anda. Aplikasi ini memberi Anda alamat IP statis dan nomor port untuk menghubungkan klien simulasi (Anda akan melakukan ini di langkah selanjutnya dalam tutorial ini). Anda dapat menganggap a

domain sebagai kelas aplikasi yang memiliki kode yang dapat dieksekusi dan opsi peluncuran yang sama. app nameMengidentifikasi instance aplikasi. Untuk informasi lebih lanjut tentang SimSpace Weaver konsep, lihat[Konsep kunci untuk SimSpace Weaver](#).

Anda dapat menggunakan DescribeApp API untuk memeriksa status aplikasi khusus setelah Anda memulainya.

```
aws simspaceweaver describe-app --simulation simulation-name --app app-name --  
domain domain-name
```

Untuk memulai aplikasi tampilan dalam tutorial ini

1. Panggilan StartApp untukViewApp.

```
aws simspaceweaver start-app --simulation simulation-name --name ViewApp --  
domain MyViewDomain
```

2. Hubungi DescribeApp untuk memeriksa status aplikasi kustom Anda.

```
aws simspaceweaver describe-app --simulation simulation-name --app ViewApp --  
domain MyViewDomain
```

Setelah status aplikasi kustom Anda (instance)STARTED, output DescribeApp akan menyertakan alamat IP dan nomor port untuk aplikasi kustom tersebut (instance). Dalam contoh output berikut, alamat IP adalah nilai Address dan nomor port adalah nilai Actual di EndpointInfo blok.

```
{  
  "Status": "STARTED",  
  "Domain": "MyViewDomain",  
  "TargetStatus": "STARTED",  
  "Simulation": "MyProjectSimulation_22-10-04_22_10_15",  
  "LaunchOverrides": {  
    "LaunchCommands": []  
  },  
  "EndpointInfo": {  
    "IngressPortMappings": [  
      {
```

```
        "Declared": 7000,  
        "Actual": 4321  
    },  
    ],  
    "Address": "198.51.100.135"  
},  
"Name": "ViewApp"  
}
```

Note

Nilai Declared adalah nomor port yang harus diikat oleh kode aplikasi Anda. Nilai Actual adalah nomor port yang SimSpace Weaver mengekspos klien untuk terhubung ke aplikasi Anda. SimSpace Weaver memetakan Declared port ke Actual port.

Note

Anda dapat menggunakan prosedur yang dijelaskan di [Dapatkan alamat IP dan nomor port aplikasi khusus](#) untuk mendapatkan alamat IP dan nomor port dari setiap aplikasi kustom yang dimulai.

8. Mulai jam.


- Ketika Anda pertama kali membuat simulasi Anda, ia memiliki jam tetapi jam tidak berjalan. Saat jam Anda tidak berjalan, simulasi Anda tidak akan memperbarui statusnya. Setelah Anda memulai jam, itu akan mulai mengirim kutu ke aplikasi Anda. Setiap tanda centang, aplikasi spasial Anda melangkah melalui entitas yang mereka miliki dan berkomitmen hasilnya SimSpace Weaver

Note

Diperlukan waktu 30-60 detik untuk memulai jam.

Panggil StartClock API.

```
aws simspaceweaver start-clock --simulation simulation-name
```

 Note

StartClockAPI menggunakan `Andasimulation-name`, yang dapat Anda temukan menggunakan ListSimulations API:

```
aws simspaceweaver list-simulations
```

parameter mulai cepat

- `-h, --membantu`
 - Buat daftar parameter ini.
- `--bersih`
 - Hapus konten direktori build sebelum membangun.
- `--al2`
 - Dibangun langsung di mesin asli alih-alih Docker. Gunakan ini hanya jika berjalan di lingkungan Amazon Linux 2, seperti WSL.
- `--uploadonly`
 - Hanya unggah skema dan ritsleting aplikasi ke Amazon S3, jangan mulai simulasi.
- `--nobuild`
 - Lewati pembangunan kembali proyek.
- `--tidak ada wadah`
 - Lewati membangun kembali wadah simulasi yang tercantum dalam skema.
- `--konolelien`
 - Secara otomatis membangun dan menghubungkan klien konsol yang tercantum di `config.py`.
- `--skema SKEMA`
 - Skema apa yang akan digunakan doa ini. Default ke nilai 'SCHEMA' di `config.py`.
- `--nama NAMA`
 - Apa nama simulasi yang akan dimiliki. Default ke nilai 'project_name'-date-time di `config.py`.

Langkah 3: Periksa log (opsional)

SimSpace Weaver menulis pesan manajemen simulasi dan output konsol dari aplikasi Anda ke Amazon CloudWatch Logs. Untuk informasi selengkapnya tentang bekerja dengan log, lihat [Bekerja dengan grup log dan aliran log](#) di Panduan Pengguna Amazon CloudWatch Logs.

Setiap simulasi yang Anda buat memiliki grup lognya sendiri di CloudWatch Log. Nama grup log ditentukan dalam skema simulasi. Dalam cuplikan skema berikut, nilainya adalah `log_destination_service logs` ini berarti bahwa nilai `log_destination_resource_name` adalah nama grup log. Dalam hal ini, grup log adalah `MySimulationLogs`.

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

Anda juga dapat menggunakan `DescribeSimulation` API untuk menemukan nama grup log untuk simulasi setelah Anda memulainya.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Contoh berikut menunjukkan bagian dari output dari `DescribeSimulation` yang menjelaskan konfigurasi logging. Nama grup log ditampilkan di akhir `LogGroupArn`.

```
"LoggingConfiguration": {
  "Destinations": [
    {
      "CloudWatchLogsLogGroup": {
        "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
      }
    }
  ]
},
```

Setiap grup log simulasi berisi beberapa aliran log:

- Aliran log manajemen — pesan manajemen simulasi yang dihasilkan oleh SimSpace Weaver layanan.

```
/sim/management
```

- Aliran log kesalahan — pesan kesalahan yang dihasilkan oleh SimSpace Weaver layanan. Aliran log ini hanya ada jika ada kesalahan. SimSpace Weaver menyimpan kesalahan yang ditulis oleh aplikasi Anda di aliran log aplikasi mereka sendiri (lihat aliran log berikut).

```
/sim/errors
```

- Aliran log aplikasi spasial (1 untuk setiap aplikasi spasial pada setiap pekerja) — keluaran konsol yang dihasilkan oleh aplikasi spasial. Setiap aplikasi spasial menulis ke aliran lognya sendiri.

spatial-app-id itu semua karakter setelah garis miring di akhir. *worker-id*

```
/domain/spatial-domain-name/app/worker-worker-id/spatial-app-id
```

- Aliran log aplikasi khusus (1 untuk setiap instance aplikasi kustom) — keluaran konsol yang dihasilkan oleh aplikasi khusus. Setiap instance aplikasi kustom menulis ke aliran lognya sendiri.

```
/domain/custom-domain-name/app/custom-app-name/random-id
```

- Aliran log aplikasi layanan (1 untuk setiap instance aplikasi layanan) — keluaran konsol yang dihasilkan oleh aplikasi layanan. Setiap aplikasi layanan menulis ke aliran lognya sendiri.

service-app-id itu semua karakter setelah garis miring di akhir. *service-app-name*

```
/domain/service-domain-name/app/service-app-name/service-app-id
```

Note

Aplikasi sampel tidak memiliki aplikasi layanan.

Langkah 4: Lihat simulasi Anda

SDK SimSpace Weaver aplikasi menyediakan opsi berbeda untuk melihat contoh aplikasi. Anda dapat menggunakan klien konsol sampel jika Anda tidak memiliki dukungan lokal untuk pengembangan Unreal Engine. Instruksi untuk klien Unreal Engine mengasumsikan bahwa Anda menggunakan Windows.

Klien konsol menampilkan daftar peristiwa entitas saat terjadi. Klien mendapatkan informasi peristiwa entitas dari `ViewApp`. Jika klien konsol Anda menampilkan daftar acara, maka itu mengonfirmasi konektivitas jaringan dengan `ViewApp` dan aktivitas dalam simulasi Anda.

`PathfindingSampleSimulasi` menciptakan entitas stasioner dan bergerak pada bidang 2 dimensi. Entitas bergerak bergerak di sekitar entitas stasioner. Klien Unreal Engine menyediakan visualisasi peristiwa entitas.

Klien konsol

Klien konsol dapat dibuat dan dihubungkan secara otomatis saat meluncurkan sampel dengan `quick-start.py` jika Anda menyertakan `--consoleclient` opsi. Untuk membangun dan menghubungkan klien konsol `quick-start.py` setelah dipanggil, lakukan hal berikut:

Arahkan ke:

```
sdk-folder/Clients/TCP/CppConsoleClient
```

Jalankan skrip untuk membangun dan menghubungkan klien:

```
python start_client.py --host ip-address --port port-number
```

Script akan melakukan hal berikut:

1. Bangun klien konsol dengan CMake.
2. Luncurkan executable yang dibangun dengan alamat IP dan nomor port yang diberikan.

```
.\WeaverNngConsoleClient.exe --url tcp://ip-address:port-number
```

Klien Unreal Engine

Lihat [Meluncurkan klien tampilan Unreal Engine](#).

Langkah 5: Hentikan dan hapus simulasi Anda

Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Temukan nama-nama simulasi Anda:

```
aws simspaceweaver list-simulations
```

Hentikan dan hapus simulasi:

```
python stop-and-delete.py --simulation simulation-name
```

Script `stop-and-delete.py` akan melakukan hal berikut:

1. Panggil AWS CLI perintah untuk menghentikan simulasi.
 - `aws simspaceweaver stop-simulation`
 - Untuk informasi selengkapnya, lihat [Referensi AWS CLI Perintah](#) untuk SimSpace Weaver.
2. Panggil AWS CLI perintah untuk menghapus simulasi.
 - `aws simspaceweaver delete-simulation`
 - Untuk informasi selengkapnya, lihat [Referensi AWS CLI Perintah](#) untuk SimSpace Weaver.

`stop-and-delete` parameter

- `-h, --membantu`
 - Buat daftar parameter ini.
- `--simulasi SIMULASI`
 - Nama simulasi untuk `stop-and-delete`
- `--berhenti`
 - Hentikan saja simulasi. Tidak menghapusnya.
- `--hapus`
 - Hapus saja simulasi. Hanya akan bekerja jika dia simulasi salah satu STOPPED atau FAILED.

Pemecahan masalah

Lihat [Pemecahan Masalah](#) di tutorial mulai cepat.

Bekerja dengan SimSpace Weaver

Bab ini memberikan informasi dan panduan untuk membantu Anda membangun aplikasi Anda sendiri SimSpace Weaver.

Topik

- [Mengkonfigurasi simulasi Anda](#)
- [Durasi maksimum simulasi](#)
- [Mengembangkan aplikasi](#)
- [Mengembangkan aplikasi klien](#)
- [Dapatkan alamat IP dan nomor port aplikasi khusus](#)
- [Meluncurkan klien tampilan Unreal Engine](#)
- [Pembangunan lokal di SimSpace Weaver](#)
- [AWS SimSpace Weaver aplikasi SDK](#)
- [AWS SimSpace Weaver kerangka demo](#)
- [Bekerja dengan kuota layanan](#)
- [Simulasi debugging](#)
- [Wadah khusus](#)
- [Bekerja menggunakan Python](#)
- [Support untuk mesin lain](#)
- [Penggunaan perangkat lunak berlisensi dengan AWS SimSpace Weaver](#)
- [Mengelola sumber daya Anda dengan AWS CloudFormation](#)
- [Snapshot](#)
- [Perpesanan](#)

Mengkonfigurasi simulasi Anda

Skema simulasi (atau skema) adalah YAML-file teks berformat yang menentukan konfigurasi untuk simulasi. Anda dapat menggunakan skema yang sama untuk memulai beberapa simulasi. File skema terletak di folder proyek untuk simulasi Anda. Anda dapat menggunakan editor teks apa pun untuk

mengedit file. SimSpace Weaver hanya membaca skema Anda ketika memulai simulasi. Setiap pengeditan yang Anda buat pada file skema hanya memengaruhi simulasi baru yang Anda mulai setelah pengeditan.

Untuk mengonfigurasi simulasi Anda, edit file skema simulasi Anda (gunakan pemisah jalur yang sesuai untuk sistem operasi Anda):

```
project-folder\tools\project-name-schema.yaml
```

Anda mengunggah skema simulasi saat membuat simulasi baru. Skrip pembantu mulai cepat untuk proyek Anda akan mengunggah skema sebagai bagian dari prosesnya untuk membangun simulasi Anda:

```
project-folder\tools\windows\quick-start.py
```

Untuk informasi selengkapnya tentang menjalankan skrip mulai cepat, lihat [Tutorial terperinci](#) di bagian [Memulai](#) panduan ini.

Parameter konfigurasi simulasi

Skema simulasi berisi informasi bootstrap, termasuk:

- Properti simulasi — [Versi SDK dan konfigurasi komputasi \(jenis dan jumlah pekerja\)](#)
- Jam - tingkat centang dan toleransi
- Strategi partisi spasial — topologi spasial (seperti grid), batas, dan kelompok penempatan (pengelompokan partisi spasial pada pekerja)
- Domain dan aplikasinya — bucket aplikasi, jalur, dan perintah peluncuran

SimSpace Weaver menggunakan konfigurasi skema Anda untuk mengonfigurasi dan mengatur partisi spasial, meluncurkan aplikasi, dan memajukan simulasi pada tingkat centang yang Anda tentukan.

Note

Skrip create-project di SDK SimSpace Weaver aplikasi akan secara otomatis menghasilkan skema simulasi untuk Anda, berdasarkan contoh aplikasi.

Topik berikut menjelaskan parameter dalam skema simulasi. Untuk deskripsi lengkap tentang skema simulasi, lihat. [SimSpace Weaver referensi skema simulasi](#)

Topik

- [Versi SDK](#)
- [Sifat simulasi](#)
- [Pekerja](#)
- [Jam](#)
- [Strategi partisi](#)
- [Domain](#)

Versi SDK

`sdk_version` Bidang menentukan versi skema SimSpace Weaver yang diformat untuk. Nilai valid: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

Nilai `sdk_version` hanya mencakup nomor versi utama dan nomor versi minor pertama. Misalnya, nilai 1.12 menentukan semua versi 1.12.x, seperti, 1.12.01.12.1, dan 1.12.2.

Sifat simulasi

`simulation_properties` Bagian skema Anda menentukan konfigurasi logging dan tipe data untuk bidang indeks (biasanya lokasi spasial) entitas.

```
simulation_properties:  
  log_destination_service: "logs"  
  log_destination_resource_name: "MySimulationLogs"  
  default_entity_index_key_type: "Vector3<f32>"
```

Nilai `log_destination_service` menentukan interpretasi nilai `log_destination_resource_name`. Saat ini, satu-satunya nilai yang didukung adalah `logs`. Ini berarti bahwa nilai `log_destination_resource_name` adalah nama grup log di Amazon CloudWatch Logs

Note

Logging adalah opsional. Jika Anda tidak mengonfigurasi properti tujuan log maka simulasi Anda tidak akan menghasilkan log.

Properti `default_entity_index_key_type` diperlukan. Satu-satunya nilai yang valid adalah `Vector3<f32>`.

Pekerja

`workers` Bagian ini menentukan jenis dan jumlah pekerja yang Anda inginkan untuk simulasi Anda. SimSpace Weaver menggunakan tipe pekerjanya sendiri yang memetakan ke jenis EC2 instans Amazon.

```
workers:  
  MyComputeWorkers:  
    type: "sim.c5.24xlarge"  
    desired: 1
```

Mengaktifkan simulasi multi-pekerja

Anda dapat membuat simulasi yang menggunakan lebih dari 1 pekerja. Secara default, simulasi menggunakan 1 pekerja. Anda harus memodifikasi skema simulasi Anda sebelum memulai simulasi.

Note

Anda tidak dapat mengubah simulasi yang sudah dimulai. Jika Anda ingin mengaktifkan multi-pekerja untuk simulasi yang sedang berjalan, Anda harus menghentikan dan menghapus simulasi terlebih dahulu.

Untuk menggunakan lebih dari satu pekerja, atur `desired` jumlah instance komputasi ke nilai yang lebih besar dari 1. Ada jumlah maksimum aplikasi untuk setiap pekerja. Untuk informasi lebih lanjut, lihat [SimSpace Titik akhir dan kuota Weaver](#). SimSpace Weaver hanya akan menggunakan lebih dari 1 pekerja ketika jumlah aplikasi pada pekerja melebihi batas ini. SimSpace Weaver dapat menempatkan aplikasi pada salah satu pekerja yang tersedia. Penempatan aplikasi pada pekerja tertentu tidak dijamin.

Cuplikan skema berikut menunjukkan konfigurasi untuk simulasi yang meminta 2 pekerja. SimSpace Weaver akan mencoba mengalokasikan pekerja kedua jika jumlah aplikasi melebihi jumlah maksimum aplikasi untuk 1 pekerja.

```
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 2
```

Jam

`clock` Bagian ini menentukan properti dari jam simulasi. Saat ini, Anda hanya dapat mengonfigurasi tingkat centang (jumlah kutu per detik yang dikirim jam ke aplikasi). Tingkat centang adalah tingkat maksimum. Tingkat tick efektif bisa lebih rendah karena semua operasi (seperti pembaruan entitas) untuk centang harus selesai sebelum centang berikutnya dapat dimulai. Tick rate juga disebut clock rate.

Nilai yang valid untuk `tick_rate` bergantung pada yang `sdk_version` ditentukan dalam skema Anda.

Nilai yang valid untuk tingkat centang

- Versi lebih awal dari "1.14":
 - 10
 - 15
 - 30
- Versi "1.14" atau yang lebih baru:
 - "10"
 - "15"
 - "30"
 - "unlimited"

Untuk informasi selengkapnya, lihat [Tingkat centang tak terbatas](#).

⚠ Important

- Untuk skema dengan `sdk_version` lebih awal "1.14" dari nilai `tick_rate` adalah bilangan bulat, seperti. 30
- Untuk skema dengan `sdk_version` of "1.14" atau yang lebih baru, nilai `tick_rate` adalah string, seperti "30". Nilai harus menyertakan tanda kutip ganda.

Jika Anda mengonversi versi "1.12" atau "1.13" skema ke versi "1.14" atau yang lebih baru, Anda harus menyertakan nilai `tick_rate` dalam tanda kutip ganda.

Tingkat centang tak terbatas

Anda dapat mengatur `tick_rate` to "unlimited" untuk mengaktifkan simulasi Anda berjalan secepat kode Anda dapat mengeksekusi. Dengan tingkat centang tak terbatas, SimSpace Weaver kirimkan centang berikutnya segera setelah semua aplikasi menyelesaikan komit untuk centang saat ini.

⚠ Important

Tingkat centang tak terbatas tidak didukung dalam SimSpace Weaver versi sebelum 1.14.0. Nilai minimum `sdk_version` dalam skema adalah "1.14".

Tingkat centang tak terbatas di SimSpace Weaver Local

SimSpace Weaver Local mengimplementasikan "unlimited" seolah-olah skema menentukan tingkat centang 10 kHz (10000). Efeknya sama dengan tingkat centang tak terbatas di AWS Cloud. Anda masih menentukan `tick_rate`: "unlimited" dalam skema Anda. Untuk informasi selengkapnya tentang SimSpace Weaver Local, lihat [Pembangunan lokal di SimSpace Weaver](#).

Pertanyaan yang sering diajukan tentang jam

T1. Dapatkah saya mengubah simulasi STARTED untuk menggunakan tick rate yang berbeda?

Anda tidak dapat mengubah tingkat centang simulasi yang sudah ada AWS Cloud di setiap tahap siklus hidupnya. Anda juga tidak dapat mengubah tingkat centang simulasi yang berjalan. SimSpace Weaver Local Anda dapat mengatur `tick_rate` dalam skema dan memulai simulasi baru dari skema itu.

T2. Dapatkah saya menjalankan simulasi saya dengan tingkat centang tak terbatas dalam versi yang lebih awal dari 1,14?

Tidak, tingkat centang tak terbatas tidak didukung dalam versi sebelum 1.14.0.

Memecahkan masalah kesalahan jam

Jika simulasi Anda gagal untuk memulai, Anda dapat memeriksa nilai "StartError" dalam output DescribeSimulationAPI. `tick_rate` Nilai yang tidak valid dalam skema Anda akan menghasilkan kesalahan berikut.

Note

Output kesalahan yang ditampilkan di sini ditampilkan pada beberapa baris untuk meningkatkan keterbacaan. Output kesalahan sebenarnya adalah satu baris.

- `sdk_version` lebih awal dari "1.14" dan nilai `tick_rate` adalah bilangan bulat yang tidak valid. Nilai valid: 10, 15, 30

```
"[{"errorType":"SchemaFormatInvalid","errorMessage":
  "\$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30]\"}]"
```

- `sdk_version` lebih awal dari "1.14" dan nilainya `tick_rate` adalah string. Nilai valid: 10, 15, 30

```
"[{"errorType":"SchemaFormatInvalid","errorMessage":
  "\$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30]\"},
  {"errorType":"SchemaFormatInvalid",
  "errorMessage":"$.clock.tick_rate: string found, integer expected\"}]"
```

- `sdk_version` "1.14" atau yang lebih baru dan nilai `tick_rate` adalah string yang tidak valid. Nilai valid: "10", "15", "30", "unlimited"

```
"[{"errorType":"SchemaFormatInvalid","errorMessage":
  "\$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30,
  unlimited]\"}]"
```

- `sdk_version` Adalah "1.14" atau lebih baru dan nilai dari `tick_rate` adalah bilangan bulat. Nilai valid: "10", "15", "30", "unlimited"

```
"[{"errorType": "SchemaFormatInvalid", "errorMessage":
  "$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30,
  unlimited]"},
  {"errorType": "SchemaFormatInvalid",
  "errorMessage": "$.clock.tick_rate: integer found, string expected"}]"
```

Strategi partisi

`partitioning_strategies` Bagian ini menentukan properti konfigurasi untuk partisi aplikasi spasial. Anda memberikan nama Anda sendiri untuk strategi partisi (sekumpulan properti di bagian ini) dan menggunakannya dalam konfigurasi aplikasi spasial Anda.

```
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [0, 1000]
      y: [0, 1000]
    grid_placement_groups:
      x: 1
      y: 1
```

`topology` Properti menentukan jenis sistem koordinat yang digunakan simulasi Anda. Nilai `Grid` menentukan grid 2 dimensi (2D).

Untuk `Grid` topologi, ruang simulasi dimodelkan sebagai kotak pembatas sejajar sumbu (AABB). Anda menentukan batas koordinat untuk setiap sumbu AABB Anda di properti. `aabb_bounds` Semua entitas yang ada secara spasial dalam simulasi Anda harus memiliki posisi di dalam AABB.

Grup penempatan grid

Grup penempatan adalah kumpulan partisi aplikasi spasial yang SimSpace Weaver ingin Anda tempatkan pada pekerja yang sama. Anda menentukan jumlah dan susunan grup penempatan (dalam kotak) di `grid_placement_groups` properti. SimSpace Weaver akan mencoba untuk mendistribusikan partisi secara merata di seluruh grup penempatan. Area kepemilikan aplikasi spasial dengan partisi dalam grup penempatan yang sama akan berdekatan secara spasial.

Kami merekomendasikan bahwa $x^* y$ sama dengan jumlah pekerja yang Anda inginkan. Jika tidak sama, SimSpace Weaver akan mencoba menyeimbangkan grup penempatan Anda di seluruh pekerja yang tersedia.

Jika Anda tidak menentukan konfigurasi grup penempatan, SimSpace Weaver akan menghitung satu untuk Anda.

Domain

Anda memberikan nama untuk satu set properti konfigurasi untuk domain. Setelan peluncuran untuk aplikasi dalam domain menentukan jenis domain:

- **launch_apps_via_start_app_call**— domain kustom
- **launch_apps_by_partitioning_strategy**— domain spasial
- **launch_apps_per_worker**(tidak termasuk dalam aplikasi sampel) — domain layanan

Important

SimSpace Weaver mendukung hingga 5 domain untuk setiap simulasi. Ini mencakup semua domain spasial, kustom, dan layanan.

```
domains:
  MyViewDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
```

```
app_config:
  package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
  launch_command: ["MySpatialApp"]
  required_resource_units:
    compute: 1
```

Note

SimSpace Weaver proyek SDK aplikasi versi 1.12.x menggunakan bucket terpisah untuk file aplikasi.zip dan skema:

- penenun- - *lowercase-project-name* -app-ritsleting *account-number* - *region*
- penenun- - -skema *lowercase-project-name* - *account-number region*

Topik

- [Konfigurasi aplikasi](#)
- [Mengkonfigurasi domain spasial](#)
- [Titik akhir jaringan](#)
- [Mengkonfigurasi domain layanan](#)

Konfigurasi aplikasi

Anda menentukan konfigurasi app (`app_config`) sebagai bagian dari konfigurasi domainnya. Semua jenis domain menggunakan properti konfigurasi aplikasi yang sama ini.

```
app_config:
  package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
  launch_command: ["MyViewApp"]
  required_resource_units:
    compute: 1
```

Note

SimSpace Weaver proyek SDK aplikasi versi 1.12.x menggunakan bucket terpisah untuk file aplikasi.zip dan skema:

- penenun- - *lowercase-project-name* -app-ritsleting *account-number* - *region*
- penenun- - -skema *lowercase-project-name* - *account-number* *region*

`packageProperti` menentukan URI S3 dari file zip dalam bucket S3. File zip berisi aplikasi yang dapat dieksekusi (juga disebut biner) dan sumber daya lain yang dibutuhkan (seperti pustaka). Setiap instance aplikasi yang dapat dieksekusi berjalan dalam Docker wadah pada pekerja.

`launch_commandProperti` menentukan nama opsi executable dan baris perintah apa pun untuk menjalankan aplikasi. Nilai dari `launch_command` adalah array. Setiap token dari seluruh string perintah peluncuran adalah elemen dalam array.

Contoh

- Untuk perintah peluncuran: `MyTestApp --option1 value1`
- Tentukan: `launch_command: ["MyTestApp", "-option1", "value1"]`

`required_resource_unitsProperti` menentukan jumlah unit sumber daya komputasi yang SimSpace Weaver harus dialokasikan ke aplikasi ini. Unit sumber daya komputasi adalah jumlah kapasitas pemrosesan (vCPU) dan memori yang tetap (RAM) pada pekerja. Anda dapat meningkatkan nilai ini untuk meningkatkan jumlah daya komputasi yang tersedia untuk aplikasi saat dijalankan pada pekerja. Ada sejumlah unit sumber daya komputasi terbatas pada setiap pekerja. Untuk informasi selengkapnya, lihat [SimSpace Titik akhir dan kuota Weaver](#).

Mengkonfigurasi domain spasial

Untuk domain spasial, Anda harus menentukan `partitioning_strategy`. Nilai properti ini adalah nama yang Anda berikan untuk strategi partisi yang Anda definisikan di bagian lain dari skema.

```
MySpatialDomain:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "MyGridPartitioning"
    grid_partition:
      x: 2
      y: 2
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
```

```
launch_command: ["MySpatialApp"]
required_resource_units:
  compute: 1
```

Note

SimSpace Weaver proyek SDK aplikasi versi 1.12.x menggunakan bucket terpisah untuk file aplikasi.zip dan skema:

- penenun- - *lowercase-project-name* -app-ritsleting *account-number* - *region*
- penenun- - -skema *lowercase-project-name* - *account-number* *region*

Strategi partisi dengan Grid topologi (satu-satunya topologi yang didukung dalam rilis ini) mengarahkan SimSpace Weaver untuk mengatur partisi aplikasi spasial domain ini dalam kisi. `grid_partition` Properti menentukan jumlah baris dan kolom dari grid partisi.

SimSpace Weaver akan memulai 1 instance aplikasi spasial untuk setiap sel di kisi partisi. Misalnya, jika domain spasial memiliki `grid_partition` nilai `x: 2` dan `y: 2`, ada $2 * 2 = 4$ partisi dalam domain spasial. SimSpace Weaver akan memulai 4 instance aplikasi yang dikonfigurasi dalam domain spasial dan menetapkan 1 partisi ke setiap instance aplikasi.

Topik

- [Persyaratan sumber daya untuk domain spasial](#)
- [Beberapa domain spasial](#)
- [Pertanyaan yang sering diajukan tentang domain spasial](#)
- [Memecahkan masalah domain spasial](#)

Persyaratan sumber daya untuk domain spasial

Anda dapat menetapkan hingga 17 unit sumber daya komputasi untuk setiap pekerja. Anda menentukan jumlah unit sumber daya komputasi yang digunakan setiap aplikasi spasial di `app_config` bagian domain spasial Anda.

Example cuplikan skema yang menampilkan unit sumber daya komputasi untuk aplikasi spasial

```
MySpatialDomain:
```

```
launch_apps_by_partitioning_strategy:
  partitioning_strategy: "MyGridPartitioning"
  grid_partition:
    x: 2
    y: 2
  app_config:
    package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp.zip"
    launch_command: ["MySpatialApp"]
    required_resource_units:
      compute: 1
```

Untuk menghitung jumlah unit sumber daya komputasi yang dibutuhkan domain, kalikan jumlah sel di kisi Anda (dalam `grid_partition`, $x * y$) dengan jumlah unit sumber daya komputasi yang ditetapkan ke aplikasi spasial.

Untuk contoh sebelumnya, domain `MySpatialDomain` menentukan:

- `x: 2`
- `y: 2`
- `compute: 1`

Grid untuk `MySpatialDomain` memiliki $2 * 2 = 4$ sel. Domain spasial membutuhkan $4 * 1 = 4$ unit sumber daya komputasi.

Jumlah total unit sumber daya komputasi untuk semua domain yang ditentukan dalam skema Anda harus kurang dari atau sama dengan jumlah pekerja dikalikan dengan `desired` jumlah maksimum unit sumber daya komputasi untuk setiap pekerja (17).

Beberapa domain spasial

Anda dapat mengonfigurasi simulasi Anda untuk menggunakan lebih dari 1 domain spasial. Misalnya, Anda dapat menggunakan 1 domain spasial untuk mengontrol aktor utama dalam simulasi (seperti orang dan mobil) dan domain spasial yang berbeda untuk mengontrol lingkungan.

Anda juga dapat menggunakan beberapa domain spasial untuk menetapkan sumber daya yang berbeda ke berbagai bagian simulasi Anda. Misalnya, jika simulasi Anda memiliki tipe entitas yang memiliki 10x lebih banyak instance entitas daripada tipe lain, Anda dapat membuat domain yang berbeda untuk menangani setiap jenis entitas dan mengalokasikan lebih banyak sumber daya untuk domain dengan lebih banyak entitas.

⚠ Important

SimSpace Weaver versi sebelum 1.14.0 tidak mendukung beberapa domain spasial.

⚠ Important

AWS SimSpace Weaver Local saat ini tidak mendukung beberapa domain spasial. Untuk informasi selengkapnya tentang SimSpace Weaver Local, lihat [Pembangunan lokal di SimSpace Weaver](#).

⚠ Important

SimSpace Weaver mendukung hingga 5 domain untuk setiap simulasi. Ini mencakup semua domain spasial, kustom, dan layanan.

Konfigurasi beberapa domain spasial

Untuk mengonfigurasi lebih dari 1 domain spasial, tambahkan definisi domain spasial lainnya sebagai bagian bernama terpisah dalam skema Anda. Setiap domain harus menentukan `launch_apps_by_partitioning_strategy` kuncinya. Lihat contoh skema berikut.

```
sdk_version: "1.14"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 1
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: Grid
    aabb_bounds:
      x: [0, 1000]
      y: [0, 1000]
domains:
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
```

```

    partitioning_strategy: "MyGridPartitioning"
    grid_partition:
      x: 2
      y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
  MySecondSpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp2.zip"
      launch_command: ["MySpatialApp2"]
      required_resource_units:
        compute: 1

```

Menempatkan domain spasial bersama-sama

Dalam beberapa skenario, Anda mungkin ingin menempatkan partisi untuk domain spasial pada pekerja di samping partisi dari domain lain. Ini dapat meningkatkan karakteristik kinerja jika partisi tersebut membuat langganan lintas domain satu sama lain.

Tambahkan kunci tingkat atas `placement_constraints` ke skema Anda untuk menentukan domain mana yang SimSpace Weaver harus ditempatkan bersama. `on_workers` Kunci yang diperlukan harus mengacu pada `workers` konfigurasi bernama dalam skema.

Example cuplikan skema yang menunjukkan domain spasial yang ditempatkan bersama

```

workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 2
placement_constraints:
  - placed_together: ["MySpatialDomain", "MySecondSpatialDomain"]
    on_workers: ["MyComputeWorkers"]

```

⚠ Important

- Jika Anda menggunakan grup penempatan:
 - Pastikan bahwa $x \cdot y$ adalah kelipatan dari jumlah pekerja.
 - Pastikan bahwa nilai grup penempatan adalah pembagi umum untuk dimensi kisi domain yang Anda tempatkan bersama.
- Jika Anda tidak menggunakan grup penempatan:
 - Pastikan bahwa 1 sumbu grid domain spasial Anda memiliki pembagi umum yang sama dengan jumlah pekerja.

Untuk informasi selengkapnya tentang grup penempatan, lihat [Strategi partisi](#).

Pertanyaan yang sering diajukan tentang domain spasial

T1. Bagaimana saya bisa menambahkan domain spasial lain ke simulasi yang ada?

- Untuk simulasi yang sedang berjalan — Anda tidak dapat mengubah konfigurasi untuk simulasi yang sedang berjalan. Ubah konfigurasi domain dalam skema, unggah skema dan ritsleting aplikasi, dan mulai simulasi baru.
- Untuk simulasi baru — Tambahkan konfigurasi domain ke skema, unggah skema dan ritsleting aplikasi, dan mulai simulasi baru.

Memecahkan masalah domain spasial

Mungkin mendapatkan kesalahan berikut ketika Anda mencoba memulai simulasi Anda tetapi konfigurasi domain Anda tidak valid.

```
"StartError": "[{"errorType": "SchemaFormatInvalid", "errorMessage":  
  "We were unable to determine an arrangement of your domains that would fit  
  within the provided set of workers. This can generally be resolved by  
  increasing the number of workers if able, decreasing your domains\u0027  
  [\u0027\u0027grid_partition\u0027\u0027] values, or adjusting the  
  dimensions of your [\u0027\u0027grid_placement_groups\u0027\u0027].\u0027}"]"
```

Potensi penyebab

- Skema mengalokasikan lebih banyak unit sumber daya komputasi untuk aplikasi daripada yang tersedia pada pekerja.
- SimSpace Weaver tidak dapat menentukan pengaturan untuk menempatkan domain bersama pada pekerja. Ini terjadi ketika Anda menentukan beberapa domain spasial tetapi tidak ada pembagi umum atau kelipatan antara kisi domain, seperti antara kisi 2x4 dan kisi 3x5).

Titik akhir jaringan

Aplikasi kustom dan layanan dapat memiliki titik akhir jaringan yang dapat dihubungkan oleh klien eksternal. Anda menentukan daftar nomor port sebagai nilai untuk `ingress_ports` dalam `endpoint_config`. Nomor port ini adalah TCP dan UDP. Aplikasi kustom atau layanan harus mengikat ke nomor port yang Anda tentukan. `ingress_ports` SimSpace Weaver secara dinamis mengalokasikan nomor port saat runtime dan memetakan port ini ke port dinamis. Anda dapat memanggil `describe-app` API setelah aplikasi Anda mulai menemukan nomor port dinamis (aktual). Untuk informasi lebih lanjut, lihat [Dapatkan alamat IP dan nomor port aplikasi khusus](#) dari tutorial mulai cepat.

```
domains:
  MyViewDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
```

Note

SimSpace Weaver proyek SDK aplikasi versi 1.12.x menggunakan bucket terpisah untuk file aplikasi.zip dan skema:

- penenun- - *lowercase-project-name* -app-ritsleting *account-number* - *region*
- penenun- - -skema *lowercase-project-name* - *account-number* *region*

Note

`endpoint_config` adalah properti opsional untuk aplikasi khusus dan aplikasi layanan. Jika Anda tidak menentukan, `endpoint_config` maka aplikasi tidak akan memiliki titik akhir jaringan.

Mengkonfigurasi domain layanan

Kehadiran `launch_apps_per_worker`: dalam konfigurasi domain menunjukkan bahwa itu adalah domain layanan yang memiliki aplikasi layanan. SimSpace Weaver memulai dan menghentikan aplikasi layanan untuk Anda. Saat SimSpace Weaver memulai dan menghentikan aplikasi, aplikasi dianggap memiliki siklus hidup terkelola. SimSpace Weaver saat ini mendukung memulai 1 atau 2 aplikasi layanan pada setiap pekerja.

Example Contoh domain yang dikonfigurasi untuk meluncurkan 1 aplikasi layanan pada setiap pekerja

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-app-zips-us-west-2/
PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

Example Contoh domain yang dikonfigurasi untuk meluncurkan 2 aplikasi layanan pada setiap pekerja

```
domains:
  MyServiceDomain:
```

```
launch_apps_per_worker:
  count: 2
  app_config:
    package: "s3://weaver-myproject-111122223333-app-zips-us-west-2/
    PlayerConnectionServiceApp.zip"
    launch_command: ["PlayerConnectionServiceApp"]
    required_resource_units:
      compute: 1
    endpoint_config:
      ingress_ports:
        - 9000
        - 9001
```

Durasi maksimum simulasi

Setiap simulasi AWS SimSpace Weaver memiliki pengaturan durasi maksimum yang menentukan waktu maksimum simulasi dapat dijalankan. Anda memberikan durasi maksimum sebagai parameter ketika Anda memulai simulasi. [Antarmuka pemrograman StartSimulation aplikasi \(API\)](#) memiliki parameter opsional `MaximumDuration`. Nilai parameter adalah sejumlah menit (m atau M), jam (h atau H), atau hari (d atau D). Misalnya, 1h atau 1H berarti 1 jam. SimSpace Weaver menghentikan simulasi Anda ketika mencapai batas ini.

Nilai maksimum

Nilai valid tertinggi untuk `MaximumDuration` adalah 14D, atau setara dalam jam (336H) atau menit (20160M).

Nilai default

Parameter `MaximumDuration` bersifat opsional. Jika Anda tidak memberikan nilai, SimSpace Weaver gunakan nilai 14D.

Nilai minimum

Nilai valid terendah untuk `MaximumDuration` adalah nilai yang secara numerik setara 0 dengan. Misalnya, nilai, 0M, dan 0H0D, semuanya setara secara numerik dengan 0.

Jika Anda memberikan nilai minimum untuk durasi maksimum, simulasi Anda segera beralih ke STOPPING status segera setelah mencapai status. STARTED

Memulai simulasi menggunakan konsol

Anda dapat memberikan nilai untuk Durasi maksimum saat memulai simulasi di [SimSpace Weaver konsol](#). Masukkan nilai di bidang Durasi maksimum formulir Pengaturan simulasi saat Anda memilih Mulai simulasi.

Important

Jika Anda tidak memberikan nilai untuk Durasi maksimum, SimSpace Weaver gunakan [nilai default](#) (14D).

Status simulasi yang mencapai durasi maksimumnya

Ketika SimSpace Weaver secara otomatis menghentikan simulasi yang mencapai durasi maksimumnya, status simulasi adalah STOPPING (jika sedang berlangsung) atau STOPPED Di [SimSpace Weaver konsol](#), status target simulasi masih STARTED, karena itu adalah status terakhir yang diminta oleh pengguna.

Mengembangkan aplikasi

SimSpace Weaver Pengembangan membutuhkan Amazon Linux 2 (AL2) lingkungan untuk membangun aplikasi karena simulasi Anda berjalan Amazon Linux di AWS Cloud. Jika Anda menggunakan Windows, Anda dapat menggunakan skrip di SDK SimSpace Weaver aplikasi untuk membuat dan meluncurkan Docker kontainer yang berjalan AL2 dengan dependensi yang Anda butuhkan untuk membangun SimSpace Weaver aplikasi. Anda juga dapat meluncurkan AL2 lingkungan menggunakan Windows Subsystem for Linux (WSL), atau gunakan asli AL2 sistem. Untuk informasi selengkapnya, lihat [Siapkan lingkungan lokal Anda untuk SimSpace Weaver](#).

Note

Terlepas dari bagaimana Anda mengonfigurasi lingkungan pengembangan lokal Anda, aplikasi Anda berjalan di Docker wadah saat Anda mengunggahnya untuk dijalankan di AWS Cloud. Aplikasi Anda tidak memiliki akses langsung ke sistem operasi host.

Alur umum SimSpace Weaver aplikasi

1. Buat aplikasi.
2. Lingkaran:
 - a. Mulai pembaruan dengan membuat `fileTransaction`.
 - Keluar dari loop jika simulasi dimatikan.
 - b. Memproses acara entitas berlangganan dan kepemilikan.
 - c. Perbarui simulasi.
 - d. Komit `Transaction` untuk mengakhiri pembaruan.
3. Hancurkan aplikasi.

Aplikasi spasial

Setiap aplikasi spasial memiliki area kepemilikan yang merupakan wilayah spasial dunia simulasi. Entitas yang terletak di area kepemilikan aplikasi spasial disimpan di partisi yang ditetapkan aplikasi. Aplikasi spasial tunggal memiliki kepemilikan penuh (izin baca dan tulis) atas semua entitas dalam partisi yang ditetapkan. Tidak ada aplikasi lain yang dapat menulis ke entitas tersebut. Aplikasi spasial memajukan keadaan entitasnya. Setiap aplikasi spasial hanya memiliki 1 partisi. SimSpace Weaver menggunakan lokasi spasial entitas untuk mengindeks dan menetapkannya ke partisi aplikasi spasial.

SimSpace Weaver Aplikasi SDK menyediakan contoh aplikasi. Anda dapat menemukan kode sumber untuk aplikasi spasial aplikasi sampel di folder berikut (gunakan pemisah jalur yang benar untuk sistem operasi Anda):

```
sdk-folder\Samples\PathfindingSample\src\SpatialApp
```

Aplikasi kustom

Anda membuat dan menggunakan aplikasi khusus untuk berinteraksi dengan simulasi.

Aplikasi kustom bisa

- Buat entitas
- Berlangganan partisi lain
- Komit perubahan

Alur umum aplikasi kustom

1. Buat aplikasi.
2. Berlangganan ke wilayah tertentu dalam simulasi:
 - a. Buat a `Transaction` untuk memulai pembaruan pertama.
 - b. Buat langganan untuk wilayah tertentu.
 - c. Komit `Transaction` untuk mengakhiri pembaruan pertama.
3. Lingkaran:
 - a. Buat a `Transaction` untuk memulai pembaruan.
 - Keluar dari loop jika simulasi dimatikan.
 - b. Perubahan status proses.
 - c. Komit `Transaction` untuk mengakhiri pembaruan.
4. Hancurkan aplikasi.

Setelah aplikasi kustom membuat entitas, aplikasi harus mentransfer entitas ke domain spasial agar entitas ada secara spasial dalam simulasi. SimSpace Weaver menggunakan lokasi spasial entitas untuk menempatkan entitas di partisi aplikasi spasial yang sesuai. Aplikasi kustom yang membuat entitas tidak dapat memperbarui atau menghapus entitas setelah mentransfernya ke domain spasial.

SimSpace Weaver Aplikasi SDK menyediakan contoh aplikasi. Anda dapat menggunakan aplikasi khusus yang disertakan dalam aplikasi sampel sebagai model untuk aplikasi kustom Anda sendiri. Anda dapat menemukan kode sumber untuk aplikasi tampilan (aplikasi khusus) dari aplikasi sampel di folder berikut (gunakan pemisah jalur yang benar untuk sistem operasi Anda):

```
sdk-folder\Samples\PathfindingSample\src\ViewApp
```

Mengembangkan aplikasi klien

Beberapa alasan Anda mungkin ingin menghubungkan klien ke simulasi meliputi:

- Menyuntikkan informasi lalu lintas real-time ke dalam simulasi skala kota.
- Buat human-in-the-loopsimulasi, di mana operator manusia mengontrol beberapa aspek simulasi.
- Memungkinkan pengguna untuk berinteraksi dengan simulasi, seperti untuk simulasi pelatihan.

Aplikasi khusus dalam contoh ini bertindak sebagai antarmuka antara status simulasi dan dunia luar. Klien terhubung ke aplikasi khusus untuk berinteraksi dengan simulasi.

SimSpace Weaver tidak menangani aplikasi klien dan komunikasinya dengan aplikasi khusus Anda. Anda bertanggung jawab atas desain, pembuatan, pengoperasian, dan keamanan aplikasi klien Anda dan komunikasinya dengan aplikasi khusus Anda. SimSpace Weaver hanya mengekspos alamat IP dan nomor port untuk setiap aplikasi kustom Anda sehingga klien dapat terhubung ke mereka.

SimSpace Weaver Aplikasi SDK menyediakan klien untuk aplikasi sampelnya. Anda dapat menggunakan klien ini sebagai model untuk aplikasi klien Anda sendiri. Anda dapat menemukan kode sumber untuk klien aplikasi sampel di folder berikut:

Docker

```
sdk-folder\packaging-tools\clients\PathfindingSampleClients
```

WSL

Important

Kami memberikan instruksi ini untuk kenyamanan Anda. Mereka untuk digunakan dengan Windows Subsystem for Linux (WSL), dan tidak didukung. Untuk informasi selengkapnya, lihat [Siapkan lingkungan lokal Anda untuk SimSpace Weaver](#).

```
sdk-folder/packaging-tools/clients/PathfindingSampleClients
```

Untuk informasi lebih lanjut tentang membangun dan menggunakan klien aplikasi sampel, lihat tutorial di [Memulai dengan SimSpace Weaver](#).

Dapatkan alamat IP dan nomor port aplikasi khusus

Untuk melihat simulasi, Anda membuat aplikasi khusus dan menghubungkannya dengan klien. Untuk informasi lebih lanjut, lihat tutorial di [Memulai dengan SimSpace Weaver](#). Anda dapat menggunakan prosedur berikut untuk mendapatkan alamat IP dan nomor port aplikasi khusus Anda. Gunakan pemisah jalur yang sesuai untuk sistem operasi Anda (misalnya, \ di Windows dan / Linux).

Untuk mendapatkan alamat IP dan nomor port

1. Gunakan `ListSimulationsAPI` untuk mendapatkan nama simulasi Anda.

```
aws simspaceweaver list-simulations
```

Contoh output:

```
{
  "Simulations": [
    {
      "Status": "STARTED",
      "CreationTime": 1664921418.09,
      "Name": "MyProjectSimulation_22-10-04_22_10_15",
      "Arn": "arn:aws:simspaceweaver:us-west-2: 111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15",
      "TargetStatus": "STARTED"
    }
  ]
}
```

2. Gunakan `DescribeSimulationAPI` untuk mendapatkan daftar domain dalam simulasi Anda.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Cari Domains bagian di `LiveSimulationState` bagian output.

Contoh output:

```
"LiveSimulationState": {
  "Domains": [
    {
      "Type": "",
      "Name": "MySpatialSimulation",
      "Lifecycle": "Unknown"
    },
    {
```

```

        "Type": "",
        "Name": "MyViewDomain",
        "Lifecycle": "ByRequest"
    }
],

```

- Gunakan `ListAppsAPI` untuk mendapatkan daftar aplikasi kustom di domain. Misalnya, nama domain untuk aplikasi tampilan (kustom) dalam proyek sampel adalah `MyViewDomain`. Cari nama aplikasi di output.

```
aws simspaceweaver list-apps --simulation simulation-name --domain domain-name
```

Contoh output:

```

{
  "Apps": [
    {
      "Status": "STARTED",
      "Domain": "MyViewDomain",
      "TargetStatus": "STARTED",
      "Name": "ViewApp",
      "Simulation": "MyProjectSimulation_22-10-04_22_10_15"
    }
  ]
}

```

- Gunakan `DescribeAppAPI` untuk mendapatkan alamat IP dan nomor port. Untuk proyek sampel, nama domain adalah `MyViewDomain` dan nama aplikasinya `ViewApp`.

```
aws simspaceweaver describe-app --simulation simulation-name --domain domain-name
--app app-name
```

Alamat IP dan nomor port ada di `EndpointInfo` blok di output. Alamat IP adalah nilai `Address` dan nomor port adalah nilai `Actual`.

Contoh output:

```
{
  "Status": "STARTED",
  "Domain": "MyViewDomain",
  "TargetStatus": "STARTED",
  "Simulation": "MyProjectSimulation_22-10-04_22_10_15",
  "LaunchOverrides": {
    "LaunchCommands": []
  },
  "EndpointInfo": {
    "IngressPortMappings": [
      {
        "Declared": 7000,
        "Actual": 4321
      }
    ],
    "Address": "198.51.100.135"
  },
  "Name": "ViewApp"
}
```

Note

Nilai Declared adalah nomor port yang harus diikat oleh kode aplikasi Anda. Nilai Actual adalah nomor port yang SimSpace Weaver mengekspos klien untuk terhubung ke aplikasi Anda. SimSpace Weaver memetakan Declared port ke Actual port.

Meluncurkan klien tampilan Unreal Engine

Arahkan ke:

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

1. Jalankan salah satu perintah berikut ini:

- buruh pelabuan: `python quick-start.py`

- WSL: `python quick-start.py --a12`
2. Dapatkan alamat IP dan nomor port “Aktual”. Ini akan berada di output konsol dari menjalankan `quick-start.py`, atau mendapatkannya dengan mengikuti prosedur di [Dapatkan alamat IP dan nomor port aplikasi khusus](#).

3. Arahkan ke:

```
sdk-folder/Clients/TCP/UnrealClient/lib
```

4. Jalankan perintah berikut untuk membangun perpustakaan NNG:

```
cmake -S . -B build
cmake --build build --config RelWithDebInfo
cmake --install build
```

5. Dalam editor teks, bukaview_app_url.txt.
6. Perbarui URL dengan alamat IP dan nomor port untuk aplikasi tampilan Anda: `tcp://ip-address:actual-port-number` (seharusnya terlihat seperti `tcp://198.51.100.135:1234`).
7. Di editor Unreal, pilih mainkan.

Pemecahan Masalah

- Langkah CMake instalasi NNG gagal dengan “Mungkin perlu hak administratif “:

```
CMake Error at build/_deps/nng-build/src/cmake_install.cmake:39 (file):
  file cannot create directory: C:/Program Files
  (x86)/ThirdPartyNngBuild/lib. Maybe need administrative privileges.
Call Stack (most recent call first):
  build/_deps/nng-build/cmake_install.cmake:37 (include)
  build/cmake_install.cmake:73 (include)
```

- Resolusi: Jika `nng.lib` atau `nng.so` ada di UnrealClient direktori/lib, kesalahan ini dapat diabaikan dengan aman. Jika tidak, coba jalankan perintah `cmake build` di terminal dengan hak administrator.
- “CMake untuk menemukan file konfigurasi paket yang disediakan oleh nng”:

```
CMake Error at CMakeLists.txt:23 (find_package):
  By not providing "Findnng.cmake" in CMAKE_MODULE_PATH this project has
```

```
asked CMake to find a package configuration file provided by "nng", but
CMake did not find one.
```

- **Resolusi:** CMake mengalami kesulitan menemukan Findnng.cmake file. Saat membangun dengan CMake, tambahkan argumen `-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty`. Pastikan Findnng.cmake file masih dalam ThirdParty direktori sebelum menjalankan ulang build. CMake

```
cmake -S . -B build -DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty
cmake --build build --config RelWithDebInfo
cmake --install build
```

Pembangunan lokal di SimSpace Weaver

Anda dapat menerapkan SimSpace Weaver aplikasi Anda secara lokal untuk pengujian cepat dan debugging.

Persyaratan

- Selesaikan langkah-langkah dalam [Menyiapkan untuk SimSpace Weaver](#).

Topik

- [Langkah 1: Luncurkan simulasi lokal Anda](#)
- [Langkah 2: Lihat simulasi lokal Anda](#)
- [Langkah 3: Hentikan simulasi lokal Anda \(opsional pada Windows\)](#)
- [Memecahkan masalah pembangunan lokal di SimSpace Weaver](#)

Langkah 1: Luncurkan simulasi lokal Anda

1. Arahkan ke

```
cd sdk-folder/Samples/sample-name/tools/local
```

2. Jalankan perintah berikut untuk membangun dan meluncurkan simulasi Anda secara lokal.

```
python quick-start.py
```

Script ini akan melakukan hal berikut:

1. Bangun proyek.

- `quick-start.py` memanggil `build_project` fungsi yang didefinisikan dalam `build.py`. Langkah ini akan bervariasi tergantung pada proyek. Untuk `PathfindingSample`, `CMake` digunakan. Perintah `CMake` and `Docker` yang dapat ditemukan di `build.py`.

2. Luncurkan simulasi lokal Anda

- Skrip akan meluncurkan satu proses lokal untuk setiap partisi spasial yang ditentukan dalam skema.
- Skrip akan meluncurkan satu proses untuk setiap aplikasi kustom yang ditentukan dalam skema.
- Aplikasi spasial akan diluncurkan terlebih dahulu, diikuti oleh aplikasi khusus — masing-masing dalam urutan mereka muncul dalam skema.

Important

Saat meluncurkan di lingkungan yang tidak mendukung GUI, seperti sesi SSH konsol, gunakan `--noappwindow` opsi untuk mengarahkan semua output ke terminal saat ini.

Important

Untuk pengguna Linux, skrip mengasumsikan sistem Anda memiliki `xterm` perintah. Jika distribusi Linux Anda tidak memiliki `xterm` perintah, gunakan `--noappwindow` opsi untuk mengarahkan semua output ke terminal saat ini.

- `-h, --membantu`
 - Buat daftar parameter ini.
- `--bersih`
 - Hapus konten direktori `build` sebelum membangun.
- `--nobuild`
 - Lewati pembangunan kembali proyek.
- `--noappwindow`

- Jangan membuka jendela baru untuk setiap aplikasi. Sebagai gantinya, arahkan stdout ke terminal saat ini.
- --berkas log
 - Tulis output konsol ke file log.
- --konolelien
 - Secara otomatis menghubungkan klien konsol yang tercantum dalam konfigurasi.
- --skema SKEMA
 - Skema apa yang akan digunakan doa ini. Default ke 'SCHEMA' di config.py.

Langkah 2: Lihat simulasi lokal Anda

Untuk melihat simulasi lokal Anda, Anda dapat menggunakan salah satu klien yang disertakan dengan. SimSpaceWeaverAppSdkDistributable Untuk informasi lebih lanjut tentang membangun dan menggunakan klien sampel, lihat tutorial di [Memulai dengan SimSpace Weaver](#).

Anda harus memperbarui alamat IP dan nomor port di klien untuk terhubung ke aplikasi tampilan untuk simulasi lokal Anda. Selalu gunakan nilai berikut dengan SimSpace Weaver Local:

```
tcp://127.0.0.1:7000
```

Tergantung pada klien yang Anda pilih, Anda dapat memperbarui alamat IP dan nomor port sebagai berikut:

- Unreal — Ubah URL pada baris 1 `view_app_url.txt`
- Konsol — Luncurkan klien dengan alamat IP dan URL nomor port sebagai parameter

Langkah 3: Hentikan simulasi lokal Anda (opsional pada Windows)

Note

Langkah ini diperlukan di Linux tetapi opsional pada Windows.

1. Arahkan ke:

```
sdk-folder/Samples/sample-name/tools/local
```

2. Jalankan perintah berikut untuk menghentikan simulasi lokal Anda dan menghapus sumber daya memori bersama.

```
python stop-and-delete.py
```

Script ini akan melakukan hal berikut:

- Hentikan proses lokal.
- Hapus objek memori bersama (hanya diperlukan di Linux).

stop-and-deletparameter .py

- -h, --membantu
 - Buat daftar parameter ini.
- --berhenti
 - Hanya mencoba untuk menghentikan proses.
- --hapus
 - Hanya mencoba untuk menghapus sumber daya memori bersama.
- --proses
 - Nama proses untuk berhenti. Gunakan ini jika nama proses Anda tidak cocok dengan nama paket dalam skema.
- --skema SKEMA
 - Skema apa yang akan digunakan doa ini. Default ke nilai 'SCHEMA' di config.py.

Memecahkan masalah pembangunan lokal di SimSpace Weaver

- Linux: perintah xterm tidak ditemukan/tidak dapat dibuka
 - Skrip lokal mengasumsikan perintah xterm ada saat berjalan di Linux. Jika Anda tidak memiliki perintah xterm atau berjalan di lingkungan yang tidak mendukung GUI, gunakan --noappwindow opsi saat menjalankan skrip mulai cepat.
- Tidak ada jendela aplikasi yang terbuka!

- Ini terjadi ketika simulasi lokal langsung mogok. Untuk melihat keluaran konsol setelah crash, gunakan `--logfile` opsi `--noappwindow` or saat menjalankan skrip mulai cepat.
- Simulasi tidak berdetak setelah aplikasi tampilan dimulai atau melihat klien terhubung!
- Berjalan dengan `--noappwindow` opsi biasanya menyelesaikan masalah semacam ini. Jika tidak, memulai ulang beberapa kali juga berhasil (walaupun pada tingkat yang jauh lebih rendah).

AWS SimSpace Weaver aplikasi SDK

SDK SimSpace Weaver aplikasi menyediakan APIs yang dapat Anda gunakan untuk mengontrol entitas dalam simulasi dan SimSpace Weaver merespons peristiwa. Ini termasuk namespace berikut:

- API — definisi inti dari API dan penggunaannya

Tautkan dengan perpustakaan berikut:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

Pustaka tersedia untuk penautan dinamis saat Anda menjalankan aplikasi di file. AWS Cloud Anda tidak perlu mengunggahnya dengan aplikasi Anda.

Note

SDK SimSpace Weaver aplikasi APIs mengontrol data dalam simulasi Anda. Ini APIs terpisah dari SimSpace Weaver layanan APIs, yang mengontrol sumber daya SimSpace Weaver layanan Anda (seperti simulasi, aplikasi, dan jam) di. AWS Untuk informasi selengkapnya, lihat [SimSpace Weaver Referensi API](#).

Topik

- [Metode API mengembalikan a Result](#)
- [Berinteraksi dengan SDK aplikasi di tingkat atas](#)

- [Manajemen simulasi](#)
- [Langganan](#)
- [Entitas](#)
- [Peristiwa entitas](#)
- [Result dan penanganan kesalahan](#)
- [Generik dan jenis domain](#)
- [Operasi SDK aplikasi lain-lain](#)

Metode API mengembalikan a Result

Sebagian besar fungsi SimSpace Weaver API memiliki tipe pengembalian `Aws::WeaverRuntime::Result<T>`. Jika fungsi telah berhasil dijalankan, `Result` berisi `T`. Jika tidak, `Result` berisi `Aws::WeaverRuntime::ErrorCode` yang mewakili kode kesalahan dari Rust App SDK.

Example Contoh

```
Result<Transaction> BeginUpdate(Application& app)
```

Metode ini:

- Mengembalikan `Transaction` jika `BeginUpdate()` mengeksekusi berhasil.
- Pengembalian `Aws::WeaverRuntime::ErrorCode` jika `BeginUpdate()` gagal.

Berinteraksi dengan SDK aplikasi di tingkat atas

Siklus hidup

- SDK SimSpace Weaver aplikasi mengelola siklus hidup aplikasi. Anda tidak perlu membaca atau menulis status siklus hidup aplikasi.

Partisi

- Gunakan `Result <PartitionSet> AssignedPartitions(Transaction& txn);` untuk mendapatkan partisi yang dimiliki.

- Gunakan `Result <PartitionSet> AllPartitions(Transaction& txn);` untuk mendapatkan semua partisi dalam simulasi.

Manajemen simulasi

Bagian ini menjelaskan solusi untuk tugas-tugas manajemen simulasi umum.

Topik

- [Mulai simulasi](#)
- [Perbarui simulasi](#)
- [Mengakhiri simulasi](#)

Mulai simulasi

Gunakan `CreateApplication()` untuk membuat aplikasi.

Example Contoh

```
Result<Application> applicationResult = Api::CreateApplication();

if (!applicationResult)
{
    ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(applicationResult);

    std::cout << "Failed to create application. Error code " <<
        static_cast<std::underlying_type_t<ErrorCode>>(errorCode) <<
        " Last error message " << Api::LastErrorMessage() << ".";

    return 1;
}

/**
 * Run simulation
 */
RunSimulation(std::move(applicationResult.assume_value()));
```

Perbarui simulasi

Gunakan `BeginUpdate` fungsi berikut untuk memperbarui aplikasi:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`— memberi tahu Anda apakah `BeginUpdate()` akan memblokir atau tidak memblokir.

Gunakan `Result<void> Commit(Transaction& txn)` untuk melakukan perubahan.

Example Contoh

```
Result<void> AppDriver::RunSimulation(Api::Application app) noexcept
{
    while (true)
    {
        {
            bool willBlock;

            do
            {
                WEAVERRUNTIME_TRY(willBlock, Api::BeginUpdateWillBlock(m_app));
            } while (willBlock);
        }

        WEAVERRUNTIME_TRY(Transaction transaction, Api::BeginUpdate(app));

        /**
         * Simulate app.
         */
        WEAVERRUNTIME_TRY(Simulate(transaction));
        WEAVERRUNTIME_TRY(Api::Commit(std::move(transaction)));
    }

    return Success();
}
```

Mengakhiri simulasi

Gunakan `Result<void> DestroyApplication(Application&& app)` untuk menghentikan aplikasi dan simulasi.

Aplikasi lain mengetahui bahwa simulasi dimatikan ketika mereka menerima `ErrorCode::ShuttingDown` dari panggilan mereka ke `BeginUpdateWillBlock()` atau `BeginUpdate()`. Ketika sebuah aplikasi menerima `ErrorCode::ShuttingDown`, ia dapat

memanggil `Result<void> DestroyApplication(Application&& app)` untuk mengakhiri dirinya sendiri.

Example Contoh

```
Result<void> AppDriver::EncounteredAppError(Application&& application) noexcept
{
    const ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(runAppResult);

    switch (errorCode)
    {
    case ErrorCode::ShuttingDown:
        {
            // insert custom shutdown process here.

            WEAVERRUNTIME_TRY(Api::DestroyApplication(std::move(application)));
            return Success();
        }
    default:
        {
            OnAppError(errorCode);
            return errorCode;
        }
    }
}
```

Important

Hanya menelepon `Result<void> DestroyApplication(Application&& app)` setelahnya `Api::Commit()`. Menghancurkan aplikasi selama pembaruan dapat menyebabkan perilaku tidak terdefinisi.

Important

Anda harus menelepon `DestroyApplication()` sebelum program keluar untuk memastikan bahwa laporan aplikasi berhasil dihentikan. Kegagalan untuk menelepon `DestroyApplication()` ketika program keluar akan menyebabkan status dianggap sebagai FATAL.

Langganan

Anda membuat langganan dengan area berlangganan dan ID domain. ID domain mewakili domain yang memiliki area langganan tersebut. A `BoundingBox2F32` menjelaskan area berlangganan. Gunakan fungsi berikut untuk membuat langganan:

```
Result<SubscriptionHandle> CreateSubscriptionBoundingBox2F32(Transaction& txn, DomainId id, const BoundingBox2F32& boundingBox)
```

Example Contoh

```
Result<void> CreateSubscriptionInSpatialDomain(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(Api::PartitionSet partitionSet, Api::AllPartitions(transaction));

    Api::DomainId spatialDomainId;

    for (const Api::Partition& partition : partitionSet.partitions)
    {
        if (partition.domain_type == Api::DomainType::Spatial)
        {
            /**
             * Get the spatial domain ID.
             */
            spatialDomainId = partition.domain_id;
            break;
        }
    }

    constexpr Api::BoundingBox2F32 subscriptionBounds {
        /* min */ { /* x */ 0, /* y */ 0 },
        /* max */ { /* x */ 1000, /* y */ 1000 } }

    WEAVERRUNTIME_TRY(
        Api::SubscriptionHandle subscriptionHandle,
        Api::CreateSubscriptionBoundingBox2F32(
            transaction,
            spatialDomainId,
            subscriptionBounds));

    return Success();
}
```

```
}
```

Anda dapat menggunakan yang `Api::SubscriptionHandle` dikembalikan oleh `CreateSubscriptionBoundingBox2F32()` untuk memodifikasi langganan. Anda meneruskannya sebagai argumen ke fungsi-fungsi berikut:

```
Result<void> ModifySubscriptionBoundingBox2F32(Transaction& txn, SubscriptionHandle handle, const BoundingBox2F32& boundingBox)
```

```
Result<void> DeleteSubscription(Transaction& txn, SubscriptionHandle handle)
```

Entitas

Anda memanggil Store dan Load APIs menggunakan peristiwa perubahan `Api::Entity` kepemilikan yang `Result<Api::Entity>` dikembalikan dari `CreateEntity()`, atau dari saat entitas memasuki area langganan aplikasi (untuk informasi selengkapnya, lihat [Peristiwa entitas](#)). Kami menyarankan Anda melacak `Api::Entity` objek Anda sehingga Anda dapat menggunakannya dengan ini APIs.

Topik

- [Buat entitas](#)
- [Transfer entitas ke domain spasial](#)
- [Menulis dan membaca data bidang entitas](#)
- [Menyimpan posisi entitas](#)
- [Memuat posisi entitas](#)

Buat entitas

Gunakan `CreateEntity()` untuk membuat entitas. Anda mendefinisikan arti dari `Api::TypeId` yang Anda berikan ke fungsi ini.

```
Namespace
{
    constexpr Api::TypeId k_entityTypeId { /* value */ 512 };
}

Result<void> CreateEntity(Transaction& transaction)
```

```
{
  WEAVERRUNTIME_TRY(
    Api::Entity entity,
    Api::CreateEntity(
      transaction, Api::BuiltinTypeIdToTypeId(k_entityTypeId ));
  }
```

Note

Nilai 0-511 untuk dicadangkan `Api::BuiltinTypeId`. Entitas Anda `TypeId` (`k_entityTypeId` dalam contoh ini) harus memiliki nilai 512 atau lebih tinggi.

Transfer entitas ke domain spasial

Setelah aplikasi kustom atau aplikasi layanan membuat entitas, aplikasi harus mentransfer entitas ke domain spasial agar entitas ada secara spasial dalam simulasi. Entitas dalam domain spasial dapat dibaca oleh aplikasi lain dan diperbarui oleh aplikasi spasial. Gunakan `ModifyEntityDomain()` API untuk mentransfer entitas ke domain spasial.

```
AWS_WEAVERRUNTIME_API Result<void> ModifyEntityDomain(Transaction& txn, const Entity&
entity, DomainId domainId) noexcept;
```

Jika `DomainId` tidak cocok dengan yang ditetapkan `Partition` dari aplikasi panggilan, maka `DomainId` harus untuk `DomainType::SpatialDomain`. Transfer kepemilikan ke yang baru `Domain` terjadi selama `Commit(Transaction&&)`.

Parameter

`txn`

Saat ini `Transaction`.

`entity`

Target `Entity` untuk perubahan `Domain`.

`domainId`

`DomainId` Tujuan `Domain` untuk `Entity`.

API ini kembali `Success` jika domain entitas berhasil diubah.

Menulis dan membaca data bidang entitas

Semua bidang data entitas adalah tipe gumpalan. Anda dapat menulis hingga 1.024 byte data ke entitas. Kami menyarankan Anda menyimpan gumpalan sekecil mungkin karena ukuran yang lebih besar akan mengurangi kinerja. Ketika Anda menulis ke gumpalan, Anda meneruskan SimSpace Weaver pointer ke data dan panjangnya. Ketika Anda membaca dari gumpalan, SimSpace Weaver memberi Anda penunjuk dan panjang untuk dibaca. Semua pembacaan harus lengkap sebelum panggilan `aplikasiCommit()`. Pointer yang dikembalikan dari panggilan baca tidak valid saat aplikasi memanggil `Commit()`.

Important

- Membaca dari pointer blob cache setelah a tidak `Commit()` didukung dan dapat menyebabkan simulasi gagal.
- Menulis ke penunjuk gumpalan yang dikembalikan dari panggilan baca tidak didukung dan dapat menyebabkan simulasi gagal.

Topik

- [Menyimpan data bidang suatu entitas](#)
- [Memuat data bidang suatu entitas](#)
- [Memuat data bidang entitas yang dihapus](#)

Menyimpan data bidang suatu entitas

Contoh berikut menunjukkan bagaimana Anda dapat menyimpan (menulis ke struktur status) data bidang entitas yang dimiliki aplikasi. Contoh-contoh ini menggunakan fungsi berikut:

```
AWS_WEAVERRUNTIME_API Result<void> StoreEntityField(  
    Transaction& txn,  
    const Entity& entity,  
    TypeId keyTypeId,  
    FieldIndex index,  
    std::int8_t* src,  
    std::size_t length) noexcept;
```

`Api::TypeId keyTypeIdParameter` mewakili tipe data dari data yang diteruskan.

`Api::TypeId` `keyTypeIdParameter` harus menerima yang sesuai `Api::TypeId` dari `Api::BuiltinTypeId`. Jika tidak ada konversi yang sesuai, Anda dapat menggunakannya `Api::BuiltinTypeId::Dynamic`.

Untuk tipe data yang kompleks, gunakan `Api::BuiltinTypeId::Dynamic`.

Note

Nilai `FieldIndex` `index` harus lebih besar dari 0. Nilai 0 dicadangkan untuk kunci indeks (lihat `StoreEntityIndexKey()`).

Example Contoh menggunakan tipe data primitif

```
namespace
{
    constexpr Api::FieldIndex k_isTrueFieldId { /* value */ 1 };
}

Result<void> SetEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    bool value = true;

    auto* src = reinterpret_cast<std::int8_t*>(value);
    size_t length = sizeof(*value);

    WEAVERRUNTIME_TRY(Api::StoreEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Bool),
        k_isTrueFieldId,
        src,
        length));
}
```

Example Contoh menggunakan struct untuk menyimpan data

```
namespace
{
```

```

    constexpr Api::FieldIndex k_dataFieldId { /* value */ 1 };
}

struct Data
{
    bool boolData;
    float floatData;
};

Result<void> SetEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    Data data = { /* boolData */ false, /* floatData */ -25.93 };

    auto* src = reinterpret_cast<std::int8_t*>(data);
    size_t length = sizeof(*data);

    WEAVERRUNTIME_TRY(Api::StoreEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Dynamic),
        k_dataFieldId,
        src,
        length));
}

```

Memuat data bidang suatu entitas

Contoh berikut menunjukkan bagaimana Anda dapat memuat (membaca dari struktur status) data bidang suatu entitas. Contoh-contoh ini menggunakan fungsi berikut:

```

Result<std::size_t> LoadEntityField(
    Transaction& txn,
    const Entity& entity,
    TypeId keyTypeId,
    FieldIndex index,
    std::int8_t** dest) noexcept;

```

`Api::TypeId` `keyTypeIdParameter` harus menerima yang sesuai `Api::TypeId` dari `Api::BuiltinTypeId`. Jika tidak ada konversi yang sesuai, Anda dapat menggunakannya `Api::BuiltinTypeId::Dynamic`.

Note

Nilai `FieldIndex` indeks harus lebih besar dari 0. Nilai 0 dicadangkan untuk kunci indeks (lihat `StoreEntityIndexKey()`).

Example Contoh menggunakan tipe data primitif

```
namespace
{
    constexpr Api::FieldIndex k_isTrueFieldId { /* value */ 1 };
}

Result<void> LoadEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Api::LoadEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Bool),
        k_isTrueFieldId,
        &dest));

    bool isTrueValue = *reinterpret_cast<bool*>(dest);
}
```

Example Contoh menggunakan struct untuk menyimpan data

```
namespace
{
    constexpr Api::FieldIndex k_dataFieldId { /* value */ 1 };
}

struct Data
{
    bool boolData;
    float floatData;
};
```

```

Result<void> LoadEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Api::LoadEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Dynamic),
        k_dataFieldId,
        &dest));

    Data dataValue = *reinterpret_cast<Data*>(dest);
}

```

Memuat data bidang entitas yang dihapus

Anda tidak dapat memuat (membaca dari struktur status) data bidang entitas untuk entitas yang telah dihapus dari area kepemilikan dan langganan aplikasi. Contoh berikut menghasilkan kesalahan karena memanggil `Api::LoadIndexKey()` entitas sebagai akibat dari `Api::ChangeListAction::Remove`. Contoh kedua menunjukkan cara yang benar untuk menyimpan dan memuat data entitas secara langsung di aplikasi.

Example Contoh kode yang salah

```

Result<void> ProcessSubscriptionChanges(Transaction& transaction)
{
    /* ... */

    WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
        Api::AllSubscriptionEvents(transaction));

    for (const Api::SubscriptionEvent& event :
        subscriptionChangeList.changes)
    {
        switch (event.action)
        {
            case Api::ChangeListAction::Remove:
            {
                std::int8_t* dest = nullptr;

```

```

    /**
     * Error!
     * This calls LoadEntityIndexKey on an entity that
     * has been removed from the subscription area.
     */
    WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
        transaction,
        event.entity,
        Api::BuiltinTypeIdToTypeId(
            Api::BuiltinTypeId::Vector3F32),
        &dest));

    AZ::Vector3 position =
        *reinterpret_cast<AZ::Vector3*>(dest);
    break;
    }
}

/* ... */
}

```

Example Contoh cara yang benar untuk menyimpan dan memuat data entitas di aplikasi

```

Result<void> ReadAndSaveSubscribedEntityPositions(Transaction& transaction)
{
    static std::unordered_map<Api::EntityId, AZ::Vector3>
        positionsBySubscribedEntity;

    WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
        Api::AllSubscriptionEvents(transaction));

    for (const Api::SubscriptionEvent& event :
        subscriptionChangeList.changes)
    {
        switch (event.action)
        {
            case Api::ChangeListAction::Add:
            {
                std::int8_t* dest = nullptr;

```

```

    /**
     * Add the position when the entity is added.
     */
    WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
        transaction,
        event.entity,
        Api::BuiltinTypeIdToTypeId(
            Api::BuiltinTypeId::Vector3F32),
        &dest));

    AZ::Vector3 position =
        *reinterpret_cast<AZ::Vector3*>(dest);
    positionsBySubscribedEntity.emplace(
        event.entity.descriptor->id, position);

    break;
}
case Api::ChangeListAction::Update:
{
    std::int8_t* dest = nullptr;

    /**
     * Update the position when the entity is updated.
     */
    WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
        transaction,
        event.entity,
        Api::BuiltinTypeIdToTypeId(
            Api::BuiltinTypeId::Vector3F32),
        &dest));

    AZ::Vector3 position =
        *reinterpret_cast<AZ::Vector3*>(dest);
    positionsBySubscribedEntity[event.entity.descriptor->id] =
        position;

    break;
}
case Api::ChangeListAction::Remove:
{
    /**
     * Load the position when the entity is removed.
     */
    AZ::Vector3 position = positionsBySubscribedEntity[

```

```

        event.entity.descriptor->id];

        /**
         * Do something with position...
         */
        break;
    }
}

/* ... */
}

```

Menyimpan posisi entitas

Anda dapat menyimpan (menulis ke kain negara) posisi entitas menggunakan struktur data integer. Contoh-contoh ini menggunakan fungsi berikut:

```

Result<void> StoreEntityIndexKey(
    Transaction& txn,
    const Entity& entity,
    TypeId keyTypeId,
    std::int8_t* src,
    std::size_t length)

```

Note

Anda `Api::BuiltinTypeId::Vector3F32` harus memberikan `Api::StoreEntityIndexKey()`, seperti yang ditunjukkan dalam contoh berikut.

Example Contoh menggunakan array untuk mewakili posisi

```

Result<void> SetEntityPositionByFloatArray(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::array<float, 3> position = { /* x */ 25, /* y */ 21, /* z */ 0 };

    auto* src = reinterpret_cast<std::int8_t*>(position.data());
}

```

```

std::size_t length = sizeof(position);

WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
    transaction,
    entity,
    Api::BuiltinTypeIdToTypeId(Api::BuiltinTypeId::Vector3F32),
    src,
    length));
}

```

Example Contoh menggunakan struct untuk mewakili posisi

```

struct Position
{
    float x;
    float y;
    float z;
};

Result<void> SetEntityPositionByStruct(
    Api::Entity& entity,
    Transaction& transaction)
{
    Position position = { /* x */ 25, /* y */ 21, /* z */ 0 };

    auto* src = reinterpret_cast<std::int8_t*>(&position);
    std::size_t length = sizeof(position);

    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(Api::BuiltinTypeId::Vector3F32),
        src,
        length));
}

```

Memuat posisi entitas

Anda dapat memuat (membaca dari struktur status) posisi entitas menggunakan struktur data integer. Contoh-contoh ini menggunakan fungsi berikut:

Note

Anda `Api::BuiltinTypeId::Vector3F32` harus memberikan `Api::LoadEntityIndexKey()`, seperti yang ditunjukkan dalam contoh berikut.

Example Contoh menggunakan array untuk mewakili posisi

```
Result<void> GetEntityPosition(Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Aws::WeaverRuntime::Api::LoadEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Vector3F32),
        &dest));

    std::array<float, 3> position =
        *reinterpret_cast<std::array<float, 3*>>(dest);
}
```

Example Contoh menggunakan struct untuk mewakili posisi

```
struct Position
{struct
    float x;
    float y;
    float z;
};

Result<void> GetEntityPosition(Api::Entity& entity, Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Aws::WeaverRuntime::Api::LoadEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
```

```
        Aws::WeaverRuntime::Api::BuiltinTypeId::Vector3F32),
        &dest));

    Position position = *reinterpret_cast<Position*>(dest);
}
```

Peristiwa entitas

Anda dapat menggunakan fungsi berikut di SDK SimSpace Weaver aplikasi untuk mendapatkan semua peristiwa kepemilikan dan langganan:

- `Result<OwnershipChangeList> OwnershipChanges(Transaction& txn)`
- `Result<SubscriptionChangeList> AllSubscriptionEvents(Transaction& txn)`

Anda dapat menggunakan kerangka kerja SimSpace Weaver demo jika Anda memerlukan pemrosesan peristiwa entitas yang digerakkan oleh panggilan balik. Untuk informasi selengkapnya, lihat file header berikut:

- `sdk-folder/packaging-tools/samples/ext/DemoFramework/include/DemoFramework/EntityEventProcessor.h`

Anda juga dapat membuat pemrosesan peristiwa entitas Anda sendiri.

Topik

- [Iterasi melalui peristiwa untuk entitas yang dimiliki](#)
- [Iterasi melalui peristiwa untuk entitas berlangganan](#)
- [Iterasi melalui peristiwa perubahan kepemilikan untuk entitas](#)

Iterasi melalui peristiwa untuk entitas yang dimiliki

Gunakan `OwnershipChanges()` untuk mendapatkan daftar peristiwa untuk entitas yang dimiliki (entitas di area kepemilikan aplikasi). Fungsi ini memiliki tanda tangan berikut:

```
Result<OwnershipChangeList> OwnershipChanges(Transaction& txn)
```

Kemudian iterasi melalui entitas dengan loop, seperti yang ditunjukkan dalam contoh berikut.

Example Contoh

```
WEAVERRUNTIME_TRY(Result<Api::OwnershipChangeList> ownershipChangesResult,  
  Api::OwnershipChanges(transaction));  
  
for (const Api::OwnershipChange& event : ownershipChangeList.changes)  
{  
  Api::Entity entity = event.entity;  
  Api::ChangeListAction action = event.action;  
  
  switch (action)  
  {  
  case Api::ChangeListAction::None:  
    // insert code to handle the event  
    break;  
  case Api::ChangeListAction::Remove:  
    // insert code to handle the event  
    break;  
  case Api::ChangeListAction::Add:  
    // insert code to handle the event  
    break;  
  case Api::ChangeListAction::Update:  
    // insert code to handle the event  
    break;  
  case Api::ChangeListAction::Reject:  
    // insert code to handle the event  
    break;  
  }  
}
```

Jenis peristiwa

- NoneEntitas berada di area dan posisi serta data bidangnya tidak dimodifikasi.
- RemoveEntitas telah dihapus dari area tersebut.
- AddEntitas ditambahkan ke area tersebut.
- UpdateEntitas berada di area tersebut dan telah dimodifikasi.
- Reject— Aplikasi gagal menghapus entitas dari area tersebut.

Note

Jika terjadi Reject peristiwa, aplikasi akan mencoba transfer lagi pada centang berikutnya.

Iterasi melalui peristiwa untuk entitas berlangganan

Gunakan `AllSubscriptionEvents()` untuk mendapatkan daftar peristiwa untuk entitas berlangganan (entitas di area langganan aplikasi). Fungsi ini memiliki tanda tangan berikut:

```
Result<SubscriptionChangeList> AllSubscriptionEvents(Transaction& txn)
```

Kemudian iterasi melalui entitas dengan loop, seperti yang ditunjukkan dalam contoh berikut.

Example Contoh

```
WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
  Api::AllSubscriptionEvents(transaction));

for (const Api::SubscriptionEvent& event : subscriptionChangeList.changes)
{
  Api::Entity entity = event.entity;
  Api::ChangeListAction action = event.action;

  switch (action)
  {
  case Api::ChangeListAction::None:
    // insert code to handle the event
    break;
  case Api::ChangeListAction::Remove:
    // insert code to handle the event
    break;
  case Api::ChangeListAction::Add:
    // insert code to handle the event
    break;
  case Api::ChangeListAction::Update:
    // insert code to handle the event
    break;
  case Api::ChangeListAction::Reject:
    // insert code to handle the event
    break;
  }
}
```

```
}
```

Jenis peristiwa

- NoneEntitas berada di area dan posisi serta data bidangnya tidak dimodifikasi.
- RemoveEntitas telah dihapus dari area tersebut.
- AddEntitas ditambahkan ke area tersebut.
- UpdateEntitas berada di area tersebut dan telah dimodifikasi.
- Reject— Aplikasi gagal menghapus entitas dari area tersebut.

Note

Jika terjadi Reject peristiwa, aplikasi akan mencoba transfer lagi pada centang berikutnya.

Iterasi melalui peristiwa perubahan kepemilikan untuk entitas

Untuk mendapatkan peristiwa di mana entitas berpindah antara area kepemilikan dan area langganan, bandingkan perubahan antara kepemilikan entitas saat ini dan sebelumnya dan peristiwa berlangganan.

Anda dapat menangani acara ini dengan membaca:

- `Api::SubscriptionChangeList`
- `Api::OwnershipEvents`

Anda kemudian dapat membandingkan perubahan dengan data yang disimpan sebelumnya.

Contoh berikut menunjukkan bagaimana Anda dapat menangani peristiwa perubahan kepemilikan entitas. Contoh ini mengasumsikan bahwa untuk entitas yang bertransisi antara entitas berlangganan dan entitas yang dimiliki (di kedua arah), `remove/add` event occurs first followed by the subscription `remove/add` peristiwa kepemilikan di centang berikutnya.

Example Contoh

```
Result<void> ProcessOwnershipEvents(Transaction& transaction)
{
    using EntityIdsByAction =
```

```

        std::unordered_map<Api::ChangeListAction,
        std::vector<Api::EntityId>>;
using EntityIdSetByAction =
        std::unordered_map<Api::ChangeListAction,
        std::unordered_set<Api::EntityId>>;

static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

EntityIdSetByAction entityIdSetByAction;

/**
 * Enumerate Api::SubscriptionChangeList items
 * and store Add and Remove events.
 */
WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionEvents,
        Api::AllSubscriptionEvents(transaction));

for (const Api::SubscriptionEvent& event : subscriptionEvents.changes)
{
    const Api::ChangeListAction action = event.action;

    switch (action)
    {
    case Api::ChangeListAction::Add:
    case Api::ChangeListAction::Remove:

        {
            entityIdSetByAction[action].insert(
                event.entity.descriptor->id);
            break;
        }
    case Api::ChangeListAction::None:
    case Api::ChangeListAction::Update:
    case Api::ChangeListAction::Reject:
        {
            break;
        }
    }
}

EntityIdsByAction entityIdsByAction;

/**
 * Enumerate Api::OwnershipChangeList items

```

```

    * and store Add and Remove events.
    */

WEAVERRUNTIME_TRY(Api::OwnershipChangeList ownershipChangeList,
    Api::OwnershipChanges(transaction));

for (const Api::OwnershipChange& event : ownershipChangeList.changes)
{
    const Api::ChangeListAction action = event.action;

    switch (action)
    {
    case Api::ChangeListAction::Add:
    case Api::ChangeListAction::Remove:
        {
            entityIdsByAction[action].push_back(
                event.entity.descriptor->id);
            break;
        }
    case Api::ChangeListAction::None:
    case Api::ChangeListAction::Update:
    case Api::ChangeListAction::Reject:
        {
            break;
        }
    }
}

std::vector<Api::EntityId> fromSubscribedToOwnedEntities;
std::vector<Api::EntityId> fromOwnedToSubscribedEntities;

/**
 * Enumerate the *previous* Api::OwnershipChangeList Remove items
 * and check if they are now in
 * the *current* Api::SubscriptionChangeList Add items.
 *
 * If true, then that means
 * OnEntityOwnershipChanged(bool isOwned = false)
 */
for (const Api::EntityId& id : m_entityIdsByPreviousOwnershipAction[
    Api::ChangeListAction::Remove])
{
    if (entityIdSetBySubscriptionAction[

```

```

        Api::ChangeListAction::Add].find(id) !=
            entityIdSetBySubscriptionAction[
                Api::ChangeListAction::Add].end())
    {
        fromOwnedToSubscribedEntities.push_back(id);
    }
}

/**
 * Enumerate the *previous* Api::OwnershipChangeList Add items
 * and check if they are now in
 * the *current* Api::SubscriptionChangeList Remove items.
 *
 * If true, then that means
 * OnEntityOwnershipChanged(bool isOwned = true)
 */
for (const Api::EntityId& id : m_entityIdsByPreviousOwnershipAction[
    Api::ChangeListAction::Add])
{
    if (entityIdSetBySubscriptionAction[
        Api::ChangeListAction::Remove].find(id) !=

            entityIdSetBySubscriptionAction[
                Api::ChangeListAction::Remove].end())
    {
        fromSubscribedToOwnedEntities.push_back(id);
    }
}

m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;

return Success();
}

```

Result dan penanganan kesalahan

`Aws::WeaverRuntime::Result<T>` Kelas menggunakan Outcome pustaka pihak ketiga. Anda dapat menggunakan pola berikut untuk memeriksa kesalahan `Result` dan `catch` yang dikembalikan oleh panggilan API.

```

void DoBeginUpdate(Application& app)
{

```

```

Result<Transaction> transactionResult = Api::BeginUpdate(app);

if (transactionResult)
{
    Transaction transaction =
        std::move(transactionResult).assume_value();

    /**
     * Do things with transaction ...
     */
}
else
{
    ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(transactionResult);
    /**
     * Macro compiles to:
     * ErrorCode errorCode = transactionResult.assume_error();
     */
}
}

```

Result pernyataan kontrol makro

Di dalam fungsi dengan tipe pengembalian `Aws::WeaverRuntime::Result<T>`, Anda dapat menggunakan `WEAVERRUNTIME_TRY` makro alih-alih pola kode sebelumnya. Makro akan menjalankan fungsi yang diteruskan ke sana. Jika fungsi yang dilewatkan gagal, makro akan membuat fungsi terlampir mengembalikan kesalahan. Jika fungsi yang dilewatkan berhasil, eksekusi berlanjut ke baris berikutnya. Contoh berikut menunjukkan penulisan ulang dari `DoBeginUpdate()` fungsi sebelumnya. Versi ini menggunakan `WEAVERRUNTIME_TRY` makro, bukan `if-else` struktur kontrol. Perhatikan bahwa tipe pengembalian fungsi adalah `Aws::WeaverRuntime::Result<void>`.

```

Aws::WeaverRuntime::Result<void> DoBeginUpdate(Application& app)
{
    /**
     * Execute Api::BeginUpdate()
     * and return from DoBeginUpdate() if BeginUpdate() fails.
     * The error is available as part of the Result.
     */
    WEAVERRUNTIME_TRY(Transaction transaction, Api::BeginUpdate(m_app));

    /**

```

```
* Api::BeginUpdate executed successfully.
*
* Do things here.
*/

return Aws::Success();
}
```

Jika `BeginUpdate()` gagal, makro membuat `DoBeginUpdate()` pengembalian lebih awal dengan kegagalan. Anda dapat menggunakan `WEAVERRUNTIME_EXPECT_ERROR` makro untuk mendapatkan `Aws::WeaverRuntime::ErrorCode` dari `BeginUpdate()`. Contoh berikut menunjukkan bagaimana `Update()` fungsi memanggil `DoBeginUpdate()` dan mendapatkan kode kesalahan pada kegagalan.

```
void Update(Application& app)
{
    Result<void> doBeginUpdateResult = DoBeginUpdate(app);

    if (doBeginUpdateResult)
    {
        /**
         * Successful.
         */
    }
    else
    {
        /**
         * Get the error from Api::BeginUpdate().
         */
        ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(doBeginUpdateResult);
    }
}
```

Anda dapat membuat kode kesalahan dari `BeginUpdate()` tersedia ke fungsi yang memanggil `Update()` dengan mengubah jenis pengembalian `Update()` ke `Aws::WeaverRuntime::Result<void>`. Anda dapat mengulangi proses ini untuk terus mengirim kode kesalahan lebih jauh ke tumpukan panggilan.

Generik dan jenis domain

SDK SimSpace Weaver aplikasi menyediakan tipe data presisi tunggal `Api::Vector2F32` dan `Api::BoundingBox2F32`, dan presisi ganda `Api::Vector2F64` dan `Api::BoundingBox2F64`. Tipe data ini adalah struktur data pasif tanpa metode kenyamanan.

Perhatikan bahwa API hanya menggunakan `Api::Vector2F32` dan `Api::BoundingBox2F32`. Anda dapat menggunakan tipe data ini untuk membuat dan memodifikasi langganan.

Kerangka SimSpace Weaver demo menyediakan versi minimal AzCore perpustakaan matematika, yang berisi `Vector3` dan `Aabb`. Untuk informasi selengkapnya, lihat file header di:

- `sdk-folder/packaging-tools/samples/ext/DemoFramework/include/AzCore/Math`

Operasi SDK aplikasi lain-lain

Topik

- [AllSubscriptionEvents and OwnershipChanges berisi peristiwa dari panggilan terakhir](#)
- [Lepaskan kunci baca setelah diproses SubscriptionChangeList](#)
- [Membuat instance aplikasi mandiri untuk pengujian](#)

AllSubscriptionEvents and OwnershipChanges berisi peristiwa dari panggilan terakhir

Nilai pengembalian panggilan ke `Api::AllSubscriptionEvents()` dan `Api::OwnershipChanges()` berisi peristiwa dari panggilan terakhir, bukan centang terakhir. Dalam contoh berikut, `secondSubscriptionEvents` dan `secondOwnershipChangeList` kosong karena fungsinya dipanggil segera setelah panggilan pertama.

Jika Anda menunggu 10 kutu dan kemudian menelepon `Api::AllSubscriptionEvents()` dan `Api::OwnershipChanges()`, hasilnya akan berisi peristiwa dan perubahan dari 10 kutu terakhir (bukan centang terakhir).

Example Contoh

```
Result<void> ProcessOwnershipChanges(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(
        Api::SubscriptionChangeList firstSubscriptionEvents,
        Api::AllSubscriptionEvents(transaction));
}
```

```

WEAVERRUNTIME_TRY(
    Api::OwnershipChangeList firstOwnershipChangeList,
    Api::OwnershipChanges(transaction));

WEAVERRUNTIME_TRY(
    Api::SubscriptionChangeList secondSubscriptionEvents,
    Api::AllSubscriptionEvents(transaction));
WEAVERRUNTIME_TRY(
    Api::OwnershipChangeList secondOwnershipChangeList,
    Api::OwnershipChanges(transaction));

/**
 * secondSubscriptionEvents and secondOwnershipChangeList are
 * both empty because there are no changes since the last call.
 */
}

```

Note

Fungsi `AllSubscriptionEvents()` diimplementasikan tetapi fungsinya tidak **SubscriptionEvents()** diimplementasikan.

Lepaskan kunci baca setelah diproses SubscriptionChangeList

Saat Anda memulai pembaruan, ada segmen memori bersama untuk data yang dikomit di partisi lain untuk centang sebelumnya. Segmen memori bersama ini mungkin dikunci oleh pembaca. Aplikasi tidak dapat sepenuhnya berkomitmen sampai semua pembaca merilis kunci. Sebagai pengoptimalan, aplikasi harus menelepon `Api::ReleaseReadLeases()` untuk melepaskan kunci setelah memproses `Api::SubscriptionChangeList` item. Ini mengurangi pertenggaran pada waktu komit. `Api::Commit()` merilis sewa baca secara default, tetapi merupakan praktik terbaik untuk merilisnya secara manual setelah memproses pembaruan langganan.

Example Contoh

```

Result<void> ProcessSubscriptionChanges(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(ProcessSubscriptionChanges(transaction));

    /**
     * Done processing Api::SubscriptionChangeList items.
    */
}

```

```
    * Release read locks.
    */

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Membuat instance aplikasi mandiri untuk pengujian

Anda dapat menggunakan `Api::CreateStandaloneApplication()` untuk membuat aplikasi mandiri untuk menguji logika aplikasi sebelum menjalankan kode dalam simulasi aktual.

Example Contoh

```
int main(int argc, char* argv[])
{
    Api::StandaloneRuntimeConfig config = {
        /* run_for_seconds (the lifetime of the app) */ 3,
        /* tick_hertz (the app clock rate) */ 10 };

    Result<Application> applicationResult =
        Api::CreateStandaloneApplication(config);

    ...
}
```

AWS SimSpace Weaver kerangka demo

Kerangka AWS SimSpace Weaver demo (demo framework) adalah perpustakaan utilitas yang dapat Anda gunakan untuk mengembangkan SimSpace Weaver aplikasi.

Kerangka demo menyediakan

- Contoh kode dan pola pemrograman untuk Anda gunakan dan periksa
- Abstraksi dan fungsi utilitas yang merampingkan pengembangan untuk aplikasi sederhana
- Cara sederhana untuk menguji fitur eksperimental SDK SimSpace Weaver aplikasi

Kami merancang SDK SimSpace Weaver aplikasi dengan akses tingkat rendah untuk memberikan SimSpace Weaver APIs kinerja yang lebih tinggi. Sebaliknya, kami merancang kerangka demo untuk

memberikan abstraksi tingkat yang lebih tinggi dan akses ke APIs yang membuatnya SimSpace Weaver lebih mudah digunakan. Biaya kemudahan penggunaan adalah tingkat kinerja yang lebih rendah dibandingkan dengan langsung menggunakan SDK SimSpace Weaver aplikasi. Simulasi yang dapat mentolerir kinerja yang lebih rendah (seperti yang tanpa persyaratan kinerja waktu nyata) mungkin merupakan kandidat yang baik untuk menggunakan kerangka kerja demo. Sebaiknya gunakan fungsionalitas asli di SDK SimSpace Weaver aplikasi untuk aplikasi yang kompleks karena kerangka kerja demo bukanlah toolkit yang lengkap.

Kerangka demo termasuk

- Contoh kode kerja yang mendukung dan menunjukkan:
 - Manajemen aliran aplikasi
 - Pemrosesan peristiwa entitas yang digerakkan oleh panggilan balik
- Satu set perpustakaan utilitas pihak ketiga:
 - spdlog (perpustakaan logging)
 - Versi minimal AZCore (perpustakaan matematika) yang hanya berisi:
 - Vector3
 - Aabb
 - cxxopts (pustaka parser opsi baris perintah)
- Fungsi utilitas khusus untuk SimSpace Weaver

Kerangka demo terdiri dari perpustakaan, file sumber, dan CMakeLists. File-file tersebut disertakan dalam paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan.

Bekerja dengan kuota layanan

Bagian ini menjelaskan cara bekerja dengan kuota layanan untuk SimSpace Weaver. Kuota juga disebut batas. Untuk daftar kuota layanan, lihat [SimSpace Titik akhir dan kuota Weaver](#). APIs Di bagian ini berasal dari kumpulan aplikasi APIs. Aplikasi APIs berbeda dari layanan APIs. Aplikasi ini APIs merupakan bagian dari SDK SimSpace Weaver aplikasi. Anda dapat menemukan dokumentasi untuk aplikasi APIs di folder SDK aplikasi di sistem lokal Anda:

```
sdk-folder\SimSpaceWeaverAppSdk-sdk-version\documentation\index.html
```

Topik

- [Dapatkan batasan untuk sebuah aplikasi](#)
- [Dapatkan jumlah sumber daya yang digunakan oleh aplikasi](#)
- [Setel ulang metrik](#)
- [Melebihi batas](#)
- [Kehabisan memori](#)
- [Praktik terbaik](#)

Dapatkan batasan untuk sebuah aplikasi

Anda dapat menggunakan API RuntimeLimits aplikasi untuk menanyakan batasan aplikasi.

```
Result<Limit> RuntimeLimit(Application& app, LimitType type)
```

Parameter

Application& aplikasi

Referensi ke aplikasi.

LimitType jenis

Enum dengan jenis batas berikut:

```
enum LimitType {  
    Unset = 0,  
    EntitiesPerPartition = 1,  
    RemoteEntityTransfers = 2,  
    LocalEntityTransfers = 3  
};
```

Contoh berikut menanyakan batas jumlah entitas.

```
WEAVERRUNTIME_TRY(auto entity_limit,  
    Api::RuntimeLimit(m_app, Api::LimitType::EntitiesPerPartition))  
Log::Info("Entity count limit", entity_limit.value);
```

Dapatkan jumlah sumber daya yang digunakan oleh aplikasi

Anda dapat memanggil API `RuntimeMetrics` aplikasi untuk mendapatkan jumlah sumber daya yang digunakan oleh aplikasi:

```
Result<std::reference_wrapper<const AppRuntimeMetrics>> RuntimeMetrics(Application& app) noexcept
```

Parameter

`Application&` aplikasi

Referensi ke aplikasi.

API mengembalikan referensi ke struct yang berisi metrik. Metrik penghitung memiliki nilai total yang berjalan dan hanya meningkat. Metrik pengukur memiliki nilai yang dapat meningkat atau menurun. Runtime aplikasi memperbarui penghitung setiap kali acara meningkatkan nilainya. Runtime hanya memperbarui pengukur saat Anda memanggil API. SimSpace Weaver menjamin bahwa referensi berlaku untuk masa pakai aplikasi. Ulangi panggilan ke API tidak akan mengubah referensi.

```
struct AppRuntimeMetrics {
    uint64_t total_committed_ticks_gauge,

    uint32_t active_entity_gauge,
    uint32_t ticks_since_reset_counter,

    uint32_t load_field_counter,
    uint32_t store_field_counter,

    uint32_t created_entity_counter,
    uint32_t deleted_entity_counter,

    uint32_t entered_entity_counter,
    uint32_t exited_entity_counter,

    uint32_t rejected_incoming_transfer_counter,
    uint32_t rejected_outgoing_transfer_counter
}
```

Setel ulang metrik

API `ResetRuntimeMetrics` aplikasi menyetel ulang nilai dalam file. `AppRuntimeMetrics` struct

```
Result<void> ResetRuntimeMetrics(Application& app) noexcept
```

Contoh berikut menunjukkan bagaimana Anda dapat menelepon `ResetRuntimeMetrics` di aplikasi Anda.

```
if (ticks_since_last_report > 100)
{
    auto metrics = WEAVERRUNTIME_EXPECT(Api::RuntimeMetrics(m_app));
    Log::Info(metrics);

    ticks_since_last_report = 0;

    WEAVERRUNTIME_EXPECT(Api::ResetRuntimeMetrics(m_app));
}
```

Melebihi batas

Panggilan API aplikasi yang melebihi batas akan menampilkan `ErrorCode::CapacityExceeded`, kecuali transfer entitas. SimSpace Weaver menangani transfer entitas secara asinkron sebagai bagian dari operasi Komit dan API `BeginUpdate` aplikasi, jadi tidak ada operasi spesifik yang mengembalikan kesalahan jika transfer gagal karena batas transfer entitas. Untuk mendeteksi kegagalan transfer, Anda dapat membandingkan nilai saat ini `rejected_incoming_transfer_counter` dan `rejected_outgoing_transfer_counter` (dalam `AppRuntimeMetrics` struct) dengan nilai sebelumnya. Entitas yang ditolak tidak akan berada di partisi, tetapi aplikasi masih dapat mensimulasikannya.

Kehabisan memori

SimSpace Weaver menggunakan proses pengumpul sampah untuk membersihkan dan melepaskan memori yang dibebaskan. Dimungkinkan untuk menulis data lebih cepat daripada pengumpul sampah dapat melepaskan memori. Jika ini terjadi, operasi tulis mungkin melebihi batas memori cadangan aplikasi. SimSpace Weaver akan mengembalikan kesalahan internal dengan pesan yang berisi `OutOfMemory` (dan detail tambahan). Untuk informasi selengkapnya, lihat [Spread menulis sepanjang waktu](#).

Praktik terbaik

Praktik terbaik berikut adalah pedoman umum untuk merancang aplikasi Anda agar tidak melebihi batas. Mereka mungkin tidak berlaku untuk desain aplikasi spesifik Anda.

Pantau sering dan perlambat

Anda harus sering memantau metrik Anda dan memperlambat operasi yang hampir mencapai batas.

Hindari melebihi batas langganan dan batas transfer

Jika memungkinkan, rancang simulasi Anda untuk mengurangi jumlah langganan jarak jauh dan transfer entitas. Anda dapat menggunakan grup penempatan untuk menempatkan beberapa partisi pada pekerja yang sama dan mengurangi kebutuhan transfer entitas jarak jauh antar pekerja.

Spread menulis sepanjang waktu

Jumlah dan ukuran pembaruan dalam tanda centang dapat berdampak signifikan pada waktu dan memori yang diperlukan untuk melakukan transaksi. Persyaratan memori yang besar dapat menyebabkan runtime aplikasi kehabisan memori. Anda dapat menyebarkan penulisan sepanjang waktu untuk menurunkan ukuran total rata-rata pembaruan per centang. Ini dapat membantu meningkatkan kinerja dan menghindari melebihi batas. Kami menyarankan agar Anda tidak menulis lebih dari rata-rata 12 MB pada setiap centang atau 1,5 KB untuk setiap entitas.

Simulasi debugging

Anda dapat menggunakan metode berikut untuk mendapatkan informasi tentang simulasi Anda.

Topik

- [Gunakan SimSpace Weaver Local dan lihat output konsol](#)
- [Lihat log Anda di Amazon CloudWatch Logs](#)
- [Gunakan describe Panggilan API](#)
- [Connect klien](#)

Gunakan SimSpace Weaver Local dan lihat output konsol

Kami menyarankan Anda mengembangkan simulasi Anda secara lokal terlebih dahulu dan kemudian menjalankannya di AWS Cloud Anda dapat melihat output konsol secara langsung saat Anda

menjalankannya SimSpace Weaver Local. Untuk informasi selengkapnya, lihat [Pembangunan lokal di SimSpace Weaver](#).

Lihat log Anda di Amazon CloudWatch Logs

Saat Anda menjalankan simulasi di AWS Cloud output konsol aplikasi Anda dikirim ke aliran log di Amazon CloudWatch Logs. Simulasi Anda juga menulis data log lainnya. Anda harus mengaktifkan login dalam skema simulasi Anda jika Anda ingin simulasi Anda untuk menulis data log. Untuk informasi selengkapnya, lihat [SimSpace Weaver log di Amazon CloudWatch Log](#).

Warning

Simulasi Anda dapat menghasilkan data log dalam jumlah besar. Data log dapat tumbuh dengan sangat cepat. Anda harus memperhatikan log Anda dengan cermat dan menghentikan simulasi Anda ketika Anda tidak membutuhkannya berjalan lagi. Penebangan dapat menghasilkan biaya besar.

Gunakan describe Panggilan API

Anda dapat menggunakan layanan berikut APIs untuk mendapatkan informasi tentang simulasi Anda di. AWS Cloud

- ListSimulations— dapatkan daftar semua simulasi Anda di. AWS Cloud

Example Contoh

```
aws simspaceweaver list-simulations
```

- DescribeSimulation— dapatkan detail tentang simulasi.

Example Contoh

```
aws simspaceweaver describe-simulation --simulation MySimulation
```

- DescribeApp— dapatkan detail tentang aplikasi.

Example Contoh

```
aws simspaceweaver describe-app --simulation MySimulation --domain MyCustomDomain --
app MyCustomApp
```

Untuk informasi lebih lanjut tentang SimSpace Weaver APIs, lihat [SimSpace Weaver Referensi API](#).

Connect klien

Anda dapat menghubungkan klien ke aplikasi kustom atau layanan yang sedang berjalan yang Anda tentukan dengan `endpoint_config` skema simulasi Anda. SDK SimSpace Weaver aplikasi menyertakan klien sampel yang dapat Anda gunakan untuk melihat contoh aplikasi. Anda dapat melihat kode sumber untuk klien sampel ini dan aplikasi sampel untuk melihat bagaimana Anda dapat membuat klien Anda sendiri. Untuk informasi selengkapnya tentang cara membangun dan menjalankan klien sampel, lihat tutorial di [Memulai dengan SimSpace Weaver](#).

Anda dapat menemukan kode sumber untuk klien sampel di folder berikut:

- `sdk-folder\packaging-tools\clients\PathfindingSampleClients\`

Debugging simulasi lokal

Anda dapat men-debug SimSpace Weaver Local aplikasi Anda dengan Microsoft Visual Studio. [Untuk informasi selengkapnya tentang cara men-debug Visual Studio, lihat. Microsoft Visual Studio documentation](#)

Untuk men-debug simulasi lokal Anda

1. Pastikan Anda `schema.yaml` ada di direktori kerja Anda.
2. Di Visual Studio, buka menu konteks untuk setiap aplikasi yang ingin Anda debug (seperti `PathfindingSampleLocalSpatial` atau `PathfindingSampleLocalView`) dan atur direktori kerja di bagian debugging.
3. Buka menu konteks untuk aplikasi yang ingin Anda debug dan pilih `Set as Startup project`.
4. Pilih `F5` untuk mulai men-debug aplikasi.

Persyaratan untuk men-debug simulasi sama dengan persyaratan untuk menjalankan simulasi secara normal. Anda harus memulai jumlah aplikasi spasial yang ditentukan dalam skema. Misalnya, jika skema menentukan kisi 2x2 dan Anda memulai aplikasi spasial dalam mode debug, simulasi tidak akan berjalan hingga Anda memulai 3 aplikasi spasial lainnya (dalam mode debug atau tidak dalam mode debug).

Untuk men-debug aplikasi kustom, Anda harus terlebih dahulu memulai aplikasi spasial Anda dan kemudian memulai aplikasi kustom di debugger.

Perhatikan bahwa simulasi Anda berjalan dalam langkah kunci. Segera setelah aplikasi mencapai breakpoint, semua aplikasi lain akan berhenti sejenak. Setelah Anda melanjutkan dari breakpoint itu, aplikasi lain akan berlanjut.

Wadah khusus

AWS SimSpace Weaver aplikasi berjalan di lingkungan containerized Amazon Linux 2 (AL2). Di AWS Cloud, SimSpace Weaver jalankan simulasi Anda dalam wadah Docker yang dibuat dari `amazonlinux:2` gambar yang disajikan dari Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR). Anda dapat membuat gambar Docker kustom, menyimpannya di Amazon ECR, dan menggunakan gambar itu untuk simulasi Anda alih-alih gambar Docker default yang kami sediakan.

Anda dapat menggunakan wadah khusus untuk mengelola dependensi perangkat lunak Anda dan menyertakan komponen perangkat lunak tambahan yang tidak ada dalam gambar Docker standar. Misalnya, Anda dapat menambahkan pustaka perangkat lunak yang tersedia untuk umum yang digunakan aplikasi ke penampung dan hanya memasukkan kode kustom Anda di file zip aplikasi.

Important

Kami hanya mendukung gambar AL2 Docker yang dihosting di repositori Amazon ECR, baik di Galeri Publik Amazon ECR atau registri ECR Amazon pribadi Anda. Kami tidak mendukung gambar Docker yang dihosting di luar Amazon ECR. Untuk informasi selengkapnya tentang Amazon ECR, lihat [Dokumentasi Registri Amazon Elastic Container](#).

Topik

- [Buat wadah khusus](#)

- [Ubah proyek untuk menggunakan wadah khusus](#)
- [Pertanyaan yang sering diajukan tentang wadah khusus](#)
- [Memecahkan masalah wadah khusus](#)

Buat wadah khusus

Instruksi ini mengasumsikan bahwa Anda tahu cara menggunakan Docker dan Amazon Elastic Container Registry (Amazon ECR). Untuk informasi selengkapnya tentang Amazon ECR, lihat [Panduan Pengguna Amazon ECR](#).

Prasyarat

- Identitas IAM (penggunaan atau peran) yang Anda gunakan untuk melakukan tindakan ini memiliki izin yang benar untuk menggunakan Amazon ECR
- Docker diinstal pada sistem lokal Anda

Untuk membuat wadah khusus

1. Buat Dockerfile Anda.

A Dockerfile untuk menjalankan AWS SimSpace Weaver aplikasi dimulai dengan Amazon Linux 2 gambar di Amazon ECR.

```
# parent image required to run AWS SimSpace Weaver apps
FROM public.ecr.aws/amazonlinux/amazonlinux:2
```

2. Bangun AndaDockerfile.
3. Unggah gambar kontainer Anda ke Amazon ECR.
 - [Gunakan Konsol Manajemen AWS](#).
 - [Gunakan AWS Command Line Interface](#).

Note

Jika Anda mendapatkan `AccessDeniedException` kesalahan saat mencoba mengunggah gambar penampung Anda ke Amazon ECR, identitas IAM Anda (pengguna atau peran) mungkin tidak memiliki izin yang diperlukan untuk menggunakan Amazon ECR. Anda dapat melampirkan kebijakan

AmazonEC2ContainerRegistryPowerUser AWS terkelola ke identitas IAM Anda dan coba lagi. Untuk informasi selengkapnya tentang cara melampirkan kebijakan, lihat [Menambahkan dan menghapus izin identitas IAM](#) di AWS Identity and Access Management Panduan Pengguna.

Ubah proyek untuk menggunakan wadah khusus

Petunjuk ini mengasumsikan bahwa Anda sudah tahu cara menggunakan AWS SimSpace Weaver dan ingin membuat alur kerja penyimpanan dan pengembangan aplikasi Anda menjadi AWS Cloud lebih efisien.

Prasyarat

- Anda memiliki wadah khusus di Amazon Elastic Container Registry (Amazon ECR). Untuk informasi selengkapnya tentang membuat wadah kustom, lihat [Buat wadah khusus](#).

Untuk memodifikasi proyek Anda untuk menggunakan wadah kustom

1. Tambahkan izin ke peran aplikasi simulasi proyek Anda untuk menggunakan Amazon ECR.
 - a. Jika Anda belum memiliki kebijakan IAM dengan izin berikut, buat kebijakan tersebut. Kami menyarankan nama kebijakansimspaceweaver-ecr. Untuk informasi selengkapnya tentang cara membuat kebijakan IAM, lihat [Membuat kebijakan IAM](#) di AWS Identity and Access Management Panduan Pengguna.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement",
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

- b. Temukan nama peran aplikasi simulasi proyek Anda:
 - i. Dalam editor teks, buka CloudFormation template:

```
sdk-folder\PackagingTools\sample-stack-template.yaml
```

- ii. Temukan `RoleName` properti di bawah `WeaverAppRole`. Nilainya adalah nama peran aplikasi simulasi proyek Anda.

Example

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  WeaverAppRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: 'weaver-MySimulation-app-role'
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - 'simspaceweaver.amazonaws.com'
```

- c. Lampirkan `simspaceweaver-ecr` kebijakan ke peran aplikasi simulasi proyek. Untuk informasi selengkapnya tentang cara melampirkan kebijakan, lihat [Menambahkan dan menghapus izin identitas IAM](#) di AWS Identity and Access Management Panduan Pengguna.
 - d. Arahkan ke *sdk-folder* dan jalankan perintah berikut untuk memperbarui SimSpace Weaver tumpukan sampel:

```
python setup.py --cloudformation
```

2. Tentukan gambar kontainer Anda dalam skema simulasi proyek.

- Anda dapat menambahkan `default_image` properti opsional di bawah `simulation_properties` untuk menentukan gambar kontainer kustom default untuk semua domain.

- Tambahkan `image` properti di `app_config` untuk domain yang ingin Anda gunakan gambar kontainer kustom. Tentukan URI repositori Amazon ECR sebagai nilainya. Anda dapat menentukan gambar yang berbeda untuk setiap domain.
- Jika `image` tidak ditentukan untuk domain dan `default_image` ditentukan, aplikasi di domain tersebut menggunakan gambar default.
- Jika `image` tidak ditentukan untuk domain dan `default_image` tidak ditentukan, aplikasi di domain tersebut berjalan dalam SimSpace Weaver wadah standar.

Example Cuplikan skema yang mencakup pengaturan wadah khusus

```
sdk_version: "1.17.0"
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest" # image to use if no image specified for a domain
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
      image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest" # custom container image to use for this domain
    MySpatialDomain:
      launch_apps_by_partitioning_strategy:
        partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
      app_config:
        package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
        launch_command: ["MySpatialApp"]
        required_resource_units:
          compute: 1
```

```
image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest" # custom container image to use for this domain
```

3. Bangun dan unggah proyek Anda seperti biasa.

Pertanyaan yang sering diajukan tentang wadah khusus

T1. Apa yang harus saya lakukan jika saya ingin mengubah isi wadah saya?

- Untuk simulasi yang sedang berjalan — Anda tidak dapat mengubah wadah untuk simulasi yang sedang berjalan. Anda harus membangun wadah baru dan memulai simulasi baru yang menggunakan wadah itu.
- Untuk simulasi baru — Buat wadah baru, unggah ke Amazon Elastic Container Registry (Amazon ECR) Registry ECR), dan mulai simulasi baru yang menggunakan wadah itu.

T2. Bagaimana saya bisa mengubah gambar kontainer untuk simulasi saya?

- Untuk simulasi yang sedang berjalan — Anda tidak dapat mengubah wadah untuk simulasi yang sedang berjalan. Anda harus memulai simulasi baru yang menggunakan wadah baru.
- Untuk simulasi baru — Tentukan gambar kontainer baru dalam skema simulasi proyek Anda. Untuk informasi selengkapnya, lihat [Ubah proyek untuk menggunakan wadah khusus](#).

Memecahkan masalah wadah khusus

Topik

- [AccessDeniedException saat mengunggah gambar Anda ke Amazon Elastic Container Registry \(Amazon ECR\)](#)
- [Simulasi yang menggunakan wadah khusus gagal dimulai](#)

AccessDeniedException saat mengunggah gambar Anda ke Amazon Elastic Container Registry (Amazon ECR)

Jika Anda mendapatkan `AccessDeniedException` kesalahan saat mencoba mengunggah gambar penampung Anda ke Amazon ECR, identitas IAM Anda (pengguna atau peran) mungkin tidak memiliki izin yang diperlukan untuk menggunakan Amazon ECR. Anda dapat melampirkan kebijakan

AmazonEC2ContainerRegistryPowerUser AWS terkelola ke identitas IAM Anda dan coba lagi. Untuk informasi selengkapnya tentang cara melampirkan kebijakan, lihat [Menambahkan dan menghapus izin identitas IAM](#) di AWS Identity and Access Management Panduan Pengguna.

Simulasi yang menggunakan wadah khusus gagal dimulai

Tip pemecahan masalah

- Jika logging diaktifkan untuk simulasi Anda, periksa log kesalahan Anda.
- Uji simulasi Anda tanpa wadah khusus.
- Uji simulasi Anda secara lokal. Lihat informasi yang lebih lengkap di [Pembangunan lokal di SimSpace Weaver](#).

Bekerja menggunakan Python

Anda dapat menggunakan Python untuk SimSpace Weaver aplikasi dan klien Anda. Kit pengembangan perangkat lunak Python (Python SDK) disertakan sebagai bagian dari paket yang dapat didistribusikan SDK aplikasi standar SimSpace Weaver . Pengembangan dengan Python bekerja dengan cara yang sama seperti pengembangan dalam bahasa lain yang didukung.

Important

SimSpace Weaver hanya mendukung Python versi 3.9.

Important

SimSpace Weaver dukungan untuk Python membutuhkan SimSpace Weaver versi 1.15.0 atau yang lebih baru.

Topik

- [Membuat proyek Python](#)
- [Memulai simulasi Python](#)
- [Contoh klien Python](#)
- [Pertanyaan yang sering diajukan tentang penggunaan Python](#)

- [Memecahkan masalah yang terkait dengan Python](#)

Membuat proyek Python

Wadah kustom Python

Untuk menjalankan SimSpace Weaver simulasi berbasis Python di AWS Cloud, Anda dapat membuat wadah khusus yang menyertakan dependensi yang diperlukan. Untuk informasi selengkapnya, lihat [Wadah khusus](#).

Wadah kustom Python harus menyertakan yang berikut:

- gcc
- openssl-devel
- bzip2-devel
- libffi-devel
- wget
- tar
- gzip
- membuat
- Python (versi 3.9)

Jika Anda menggunakan PythonBubblesSample template untuk membuat proyek Anda, Anda dapat menjalankan `quick-start.py` skrip (terletak di `tools` folder proyek Anda) untuk membuat gambar Docker dengan dependensi yang diperlukan. Skrip mengunggah gambar ke Amazon Elastic Container Registry (Amazon ECR).

`quick-start.py` skrip menggunakan yang berikut ini Dockerfile:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2
RUN yum -y install gcc openssl-devel bzip2-devel libffi-devel
RUN yum -y install wget
RUN yum -y install tar
RUN yum -y install gzip
RUN yum -y install make
WORKDIR /opt
```

```
RUN wget https://www.python.org/ftp/python/3.9.0/Python-3.9.0.tgz
RUN tar xzf Python-3.9.0.tgz
WORKDIR /opt/Python-3.9.0
RUN ./configure --enable-optimizations
RUN make altinstall
COPY requirements.txt ./
RUN python3.9 -m pip install --upgrade pip
RUN pip3.9 install -r requirements.txt
```

Anda dapat menambahkan dependensi Anda sendiri ke: `Dockerfile`

```
RUN yum -y install dependency-name
```

`requirements.txt` File tersebut berisi daftar paket Python yang diperlukan untuk simulasi `PythonBubblesSample` sampel:

```
Flask==2.1.1
```

Anda dapat menambahkan dependensi paket Python Anda sendiri ke: `requirements.txt`

```
package-name==version-number
```

Itu `Dockerfile` dan `requirements.txt` ada di `tools` folder proyek Anda.

Important

Anda secara teknis tidak harus menggunakan wadah khusus dengan simulasi Python Anda, tetapi kami sangat menyarankan Anda menggunakan wadah khusus. Wadah Amazon Linux 2 (AL2) standar yang kami sediakan tidak memiliki Python. Oleh karena itu, jika Anda tidak menggunakan wadah khusus yang memiliki Python, Anda harus menyertakan Python dan dependensi yang diperlukan di setiap file zip aplikasi yang Anda unggah. SimSpace Weaver

Memulai simulasi Python

Anda dapat memulai simulasi berbasis Python dengan cara yang sama seperti simulasi biasa SimSpace Weaver, baik di SimSpace Weaver Local dan SimSpace Weaver di dalam AWS Cloud. Untuk informasi lebih lanjut, lihat tutorial di [Memulai dengan SimSpace Weaver](#).

PythonBubblesSample ini termasuk klien sampel Python sendiri. Untuk informasi selengkapnya, lihat [Contoh klien Python](#).

Contoh klien Python

Jika Anda menggunakan PythonBubblesSample template untuk membuat proyek maka proyek Anda berisi contoh klien Python. Anda dapat menggunakan klien sampel untuk melihat PythonBubblesSample simulasi. Anda juga dapat menggunakan klien sampel sebagai titik awal untuk membuat klien Python Anda sendiri.

Prosedur berikut mengasumsikan bahwa Anda membuat PythonBubblesSample proyek dan memulai simulasinya.

Untuk memulai klien Python

1. Di jendela prompt perintah, buka folder proyek PyBubbleClient sampel.

```
cd sdk-folder\Clients\HTTP\PyBubbleClient
```

2. Jalankan klien Python.

```
python tkinter_client.py --host ip-address --port port-number
```

Parameter

host

Alamat IP simulasi Anda. Untuk simulasi yang dimulai di AWS Cloud, Anda dapat menemukan alamat IP simulasi Anda di [SimSpace Weaver konsol](#) atau menggunakan prosedur di [Dapatkan alamat IP dan nomor port aplikasi khusus](#) tutorial mulai cepat. Untuk simulasi lokal, gunakan 127.0.0.1 sebagai alamat IP.

port

Nomor port simulasi Anda. Untuk simulasi yang dimulai pada AWS Cloud, ini adalah nomor Actual port. Anda dapat menemukan nomor port simulasi Anda di [SimSpace Weaver konsol](#) atau menggunakan prosedur di [Dapatkan alamat IP dan nomor port aplikasi khusus](#) tutorial mulai cepat. Untuk simulasi lokal, gunakan 7000 sebagai nomor port.

simsiz

Jumlah maksimum entitas untuk ditampilkan di klien.

Pertanyaan yang sering diajukan tentang penggunaan Python

T1. Versi Python apa yang didukung?

SimSpace Weaver hanya mendukung Python versi 3.9.

Memecahkan masalah yang terkait dengan Python

Topik

- [Kegagalan selama pembuatan wadah khusus](#)
- [Simulasi Python Anda gagal dimulai](#)
- [Sebuah simulasi Python atau tampilan klien melempar kesalahan ModuleNotFound](#)

Kegagalan selama pembuatan wadah khusus

Jika Anda mendapatkan kesalahan `no basic auth credentials` setelah menjalankan `quick-start.py` maka mungkin ada masalah dengan kredensi sementara Anda untuk Amazon ECR. Jalankan perintah berikut dengan Wilayah AWS ID dan nomor AWS akun Anda:

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin account_id.dkr.ecr.region.amazonaws.com
```

Example

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 111122223333.dkr.ecr.region.amazonaws.com
```

Important

Pastikan bahwa yang Wilayah AWS Anda tentukan sama dengan yang Anda gunakan untuk simulasi Anda. Gunakan salah satu Wilayah AWS yang SimSpace Weaver mendukung. Untuk informasi selengkapnya, lihat [SimSpace Titik akhir dan kuota Weaver](#).

Setelah Anda menjalankan `aws ecr` perintah, jalankan `quick-start.py` lagi.

Sumber pemecahan masalah lainnya untuk diperiksa

- [Memecahkan masalah wadah khusus](#)

- [Pemecahan masalah Amazon ECR di Panduan Pengguna Amazon ECR](#)
- [Menyiapkan dengan Amazon ECR](#) di Panduan Pengguna Amazon ECR

Simulasi Python Anda gagal dimulai

Anda mungkin melihat `Unable to start app` kesalahan dalam log manajemen simulasi Anda. Ini bisa terjadi jika pembuatan penampung kustom Anda gagal. Untuk informasi selengkapnya, lihat [Kegagalan selama pembuatan wadah khusus](#). Untuk informasi lebih lanjut tentang log, lihat [SimSpace Weaver log di Amazon CloudWatch Log](#).

Jika Anda yakin tidak ada yang salah dengan container Anda, periksa kode sumber Python aplikasi Anda. Anda dapat menggunakan SimSpace Weaver Local untuk menguji aplikasi Anda. Untuk informasi selengkapnya, lihat [Pembangunan lokal di SimSpace Weaver](#).

Sebuah simulasi Python atau tampilan klien melempar kesalahan `ModuleNotFound`

Python melempar `ModuleNotFound` kesalahan ketika tidak dapat menemukan paket Python yang diperlukan.

Jika simulasi Anda ada di dalam AWS Cloud, pastikan bahwa penampung kustom Anda memiliki semua dependensi yang diperlukan yang tercantum dalam file Anda. `requirements.txt` Ingatlah untuk menjalankan `quick-start.py` lagi jika Anda mengedit `requirements.txt`.

Jika Anda mendapatkan kesalahan untuk `PythonBubblesSample` klien, gunakan `pip` untuk menginstal paket yang ditunjukkan:

```
pip install package-name==version-number
```

Support untuk mesin lain

Anda dapat menggunakan kustom Anda sendiri C++ mesin dengan SimSpace Weaver. Saat ini kami sedang mengembangkan dukungan untuk mesin berikut. Ada dokumentasi terpisah untuk masing-masing mesin ini.

Important

Integrasi dengan mesin yang tercantum di sini bersifat eksperimental. Mereka tersedia untuk pratinjau.

Mesin

- [Unity](#)(versi minimum 2022.3.19.F1)
- [Unreal Engine](#)(versi minimum 5.0)

Unity

Anda harus memiliki Unity lingkungan pengembangan sudah diinstal sebelum Anda membangun SimSpace Weaver simulasi dengan Unity. Untuk informasi selengkapnya, lihat petunjuk terpisah:

`sdk-folder\Unity-Guide.pdf`

Unreal Engine

Anda harus membangun sebuah Unreal Engine server khusus dari kode sumber. SimSpaceWeaverAppSdkDistributable Termasuk versi PathfindingSample untuk Unreal Engine. Untuk informasi selengkapnya, lihat petunjuk terpisah:

`sdk-folder\Unreal-Engine-Guide.pdf`

Penggunaan perangkat lunak berlisensi dengan AWS SimSpace Weaver

AWS SimSpace Weaver memungkinkan Anda membangun simulasi dengan mesin dan konten simulasi pilihan Anda. Sehubungan dengan penggunaan Anda SimSpace Weaver, Anda bertanggung jawab untuk memperoleh, memelihara, dan mematuhi ketentuan lisensi perangkat lunak atau konten apa pun yang Anda gunakan dalam simulasi Anda. Verifikasi perjanjian lisensi Anda memungkinkan Anda untuk menyebarkan perangkat lunak dan konten Anda di lingkungan yang dihosting virtual.

Mengelola sumber daya Anda dengan AWS CloudFormation

Anda dapat menggunakan AWS CloudFormation untuk mengelola AWS SimSpace Weaver sumber daya Anda. CloudFormation adalah AWS layanan terpisah yang membantu Anda menentukan, menyediakan, dan mengelola AWS infrastruktur Anda sebagai kode. [Dengan CloudFormation](#)

[Anda membuat file JSON atau YAMB, yang disebut template.](#) Template Anda menentukan rincian infrastruktur Anda. CloudFormation menggunakan template Anda untuk menyediakan infrastruktur Anda sebagai satu unit, yang disebut [tumpukan](#). Ketika Anda menghapus tumpukan Anda, Anda dapat CloudFormation menghapus semua yang ada di tumpukan secara bersamaan. Anda dapat mengelola template Anda menggunakan proses manajemen kode sumber standar (misalnya, melacaknya dalam sistem kontrol versi seperti [Git](#)). Untuk informasi selengkapnya CloudFormation, lihat [Panduan AWS CloudFormation Pengguna](#).

Sumber daya simulasi Anda

Di AWS, sumber daya adalah entitas yang dapat Anda gunakan. Contohnya termasuk EC2 instans Amazon, bucket Amazon S3, atau peran IAM. SimSpace Weaver Simulasi Anda adalah sumber daya. Dalam konfigurasi, Anda biasanya menentukan AWS sumber daya dalam formulir `AWS::service::resource`. Untuk SimSpace Weaver, Anda menentukan sumber daya simulasi Anda sebagai `AWS::SimSpaceWeaver::Simulation`. Untuk informasi selengkapnya tentang sumber daya simulasi Anda CloudFormation, lihat [SimSpace Weaver](#) bagian di Panduan AWS CloudFormation Pengguna.

Bagaimana saya bisa menggunakan CloudFormation dengan SimSpace Weaver?

Anda dapat membuat CloudFormation template yang menentukan AWS sumber daya yang ingin Anda sediakan. Template Anda dapat menentukan seluruh arsitektur, bagian dari arsitektur, atau solusi kecil. Misalnya, Anda dapat menentukan arsitektur untuk SimSpace Weaver solusi yang menyertakan bucket Amazon S3, izin IAM, database pendukung di Amazon Relational Database Service atau Amazon DynamoDB, dan sumber daya Anda. `Simulation` Anda kemudian dapat menggunakan CloudFormation untuk menyediakan semua sumber daya tersebut sebagai satu unit, dan pada saat yang sama.

Example template yang membuat sumber daya IAM dan memulai simulasi

Contoh template berikut membuat peran IAM dan izin yang SimSpace Weaver perlu melakukan tindakan di akun Anda. Skrip SDK SimSpace Weaver aplikasi membuat peran dan izin secara spesifik Wilayah AWS saat Anda membuat proyek, tetapi Anda dapat menggunakan CloudFormation templat untuk menerapkan simulasi ke yang lain Wilayah AWS tanpa menjalankan skrip lagi. Misalnya, Anda dapat melakukan ini untuk menyiapkan simulasi cadangan untuk tujuan pemulihan bencana.

Dalam contoh ini, nama simulasi aslinya adalah `MySimulation`. Sebuah ember untuk skema sudah ada di Wilayah AWS mana CloudFormation akan membangun tumpukan. Bucket berisi versi

skema yang dikonfigurasi dengan benar untuk menjalankan simulasi di dalamnya. Wilayah AWS Ingat bahwa skema menentukan lokasi file zip aplikasi Anda, yang merupakan bucket Amazon S3 yang Wilayah AWS sama dengan simulasi. Bucket ritsleting aplikasi dan file harus sudah ada di Wilayah AWS saat CloudFormation membangun tumpukan, jika tidak simulasi Anda tidak akan dimulai. Perhatikan bahwa nama bucket dalam contoh ini menyertakan Wilayah AWS, tetapi itu tidak menentukan di mana sebenarnya bucket berada. Anda harus memastikan bahwa bucket benar-benar ada di dalamnya Wilayah AWS (Anda dapat memeriksa properti bucket di konsol Amazon S3, dengan Amazon S3, atau dengan APIs perintah Amazon S3 di). AWS CLI

Contoh ini menggunakan beberapa fungsi dan parameter bawaan CloudFormation untuk melakukan substitusi variabel. Untuk informasi selengkapnya, lihat [Referensi fungsi intrinsik dan referensi parameter semu](#) di Panduan Pengguna.AWS CloudFormation

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  WeaverAppRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: SimSpaceWeaverAppRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - simspaceweaver.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
    Policies:
      - PolicyName: SimSpaceWeaverAppRolePolicy
        PolicyDocument:
          Version: 2012-10-17
          Statement:
            - Effect: Allow
              Action:
                - logs:PutLogEvents
                - logs:DescribeLogGroups
                - logs:DescribeLogStreams
                - logs>CreateLogGroup
                - logs>CreateLogStream
```

```

    Resource: *
  - Effect: Allow
    Action:
      - cloudwatch:PutMetricData
    Resource: *
  - Effect: Allow
    Action:
      - s3:ListBucket
      - s3:PutObject
      - s3:GetObject
    Resource: *
MyBackupSimulation:
  Type: AWS::SimSpaceWeaver::Simulation
  Properties:
    Name: !Sub 'mySimulation-${AWS::Region}'
    RoleArn: !GetAtt WeaverAppRole.Arn
    SchemaS3Location:
      BucketName: !Sub 'weaver-mySimulation-${AWS::AccountId}-schemas-${AWS::Region}'
      ObjectKey: !Sub 'schema/mySimulation-${AWS::Region}-schema.yaml'

```

Menggunakan snapshot dengan AWS CloudFormation

[Snapshot](#) adalah cadangan dari simulasi. Contoh berikut memulai simulasi baru dari snapshot bukan dari skema. Snapshot dalam contoh ini dibuat dari simulasi proyek SDK SimSpace Weaver aplikasi. CloudFormation membuat sumber daya simulasi baru dan menginisialisasi dengan data dari snapshot. Simulasi baru dapat berbeda `MaximumDuration` dari simulasi aslinya.

Kami menyarankan Anda membuat dan menggunakan salinan peran aplikasi simulasi asli Anda. Peran aplikasi simulasi asli dapat dihapus jika Anda menghapus tumpukan simulasi itu. CloudFormation

```

Description: "Example - Start a simulation from a snapshot"
Resources:
  MyTestSimulation:
    Type: "AWS::SimSpaceWeaver::Simulation"
    Properties:
      MaximumDuration: "2D"
      Name: "MyTestSimulation_from_snapshot"
      RoleArn: "arn:aws:iam::111122223333:role/weaver-MyTestSimulation-app-role-copy"

      SnapshotS3Location:

```

```
BucketName: "weaver-mytestsimulation-111122223333-artifacts-us-west-2"  
ObjectKey: "snapshot/MyTestSimulation_22-12-15_12_00_00-230428-1207-13.zip"
```

Snapshot

Anda dapat membuat snapshot untuk mencadangkan data entitas simulasi Anda kapan saja. SimSpace Weaver membuat file.zip di bucket Amazon S3. Anda dapat membuat simulasi baru dengan snapshot. SimSpace Weaver menginisialisasi State Fabric simulasi baru Anda dengan data entitas yang disimpan dalam snapshot, memulai aplikasi spasial dan layanan yang berjalan saat snapshot dibuat, dan menyetel jam ke centang yang sesuai. SimSpace Weaver mendapatkan konfigurasi simulasi Anda dari snapshot alih-alih dari file skema. File zip aplikasi Anda harus berada di lokasi yang sama di Amazon S3 seperti di simulasi asli. Anda harus memulai aplikasi khusus apa pun secara terpisah.

Topik

- [Gunakan kasus untuk snapshot](#)
- [Gunakan SimSpace Weaver konsol untuk bekerja dengan snapshot](#)
- [Gunakan AWS CLI untuk bekerja dengan snapshot](#)
- [Menggunakan snapshot dengan AWS CloudFormation](#)
- [Pertanyaan yang sering diajukan tentang snapshot](#)

Gunakan kasus untuk snapshot

Kembali ke keadaan sebelumnya dan jelajahi skenario percabangan

Anda dapat membuat snapshot simulasi Anda untuk menyimpannya pada status tertentu. Anda kemudian dapat membuat beberapa simulasi baru dari snapshot itu dan menjelajahi berbagai skenario yang dapat bercabang dari status itu.

Praktik terbaik pemulihan bencana dan keamanan

Kami menyarankan Anda secara teratur mencadangkan simulasi Anda, terutama untuk simulasi yang berjalan selama lebih dari 1 jam atau menggunakan banyak pekerja. Cadangan dapat membantu Anda pulih dari bencana dan insiden keamanan. Snapshots menyediakan cara bagi Anda untuk

membuat cadangan simulasi Anda. Snapshot mengharuskan file.zip aplikasi Anda ada di lokasi yang sama di Amazon S3 seperti sebelumnya. Jika Anda harus dapat memindahkan file.zip aplikasi Anda ke lokasi lain, Anda harus menggunakan solusi cadangan khusus.

Untuk informasi lebih lanjut tentang praktik terbaik lainnya, lihat [Praktik terbaik saat bekerja dengan SimSpace Weaver](#) dan [Praktik terbaik keamanan untuk SimSpace Weaver](#).

Perpanjang durasi simulasi Anda

Sumber daya simulasi Anda adalah representasi dari simulasi Anda. SimSpace Weaver Semua sumber daya simulasi memiliki `MaximumDuration` pengaturan. Sumber daya simulasi secara otomatis berhenti ketika mencapainya `MaximumDuration`. Nilai maksimum `MaximumDuration` adalah 14D (14 hari).

Jika Anda membutuhkan simulasi untuk bertahan lebih lama dari sumber daya simulasi, Anda dapat membuat snapshot sebelum sumber daya simulasi mencapai sumber daya simulasi. `MaximumDuration` `MaximumDuration` Anda dapat memulai simulasi baru (membuat sumber daya simulasi baru) dengan snapshot Anda. SimSpace Weaver menginisialisasi data entitas Anda dari snapshot, memulai aplikasi spasial dan layanan yang sama yang berjalan sebelumnya, dan memulihkan jam. Anda dapat memulai aplikasi kustom dan melakukan inisialisasi kustom tambahan. Anda dapat mengatur sumber daya simulasi baru ke nilai yang berbeda saat Anda memulainya. `MaximumDuration`

Gunakan SimSpace Weaver konsol untuk bekerja dengan snapshot

Anda dapat menggunakan SimSpace Weaver konsol untuk membuat snapshot simulasi Anda.

Topik

- [Gunakan konsol untuk membuat snapshot](#)
- [Gunakan konsol untuk memulai simulasi dari snapshot](#)

Gunakan konsol untuk membuat snapshot


Untuk membuat snapshot

1. Masuk ke Konsol Manajemen AWS dan sambungkan ke [SimSpace Weaver konsol](#).
2. Pilih Simulasi dari panel navigasi.
3. Pilih tombol radio di sebelah nama simulasi. Status simulasi Anda harus Dimulai.

4. Di bagian atas halaman, pilih Buat snapshot.
5. Di bawah pengaturan Snapshot, untuk tujuan Snapshot, masukkan URI Amazon S3 dari bucket atau bucket dan folder tempat Anda SimSpace Weaver ingin membuat snapshot. Anda dapat memilih Browse S3 jika Anda lebih suka menelusuri bucket yang tersedia dan memilih lokasi.

 Important

Bucket Amazon S3 harus Wilayah AWS sama dengan simulasi.

 Note

SimSpace Weaver membuat snapshot folder di dalam tujuan snapshot yang Anda pilih. SimSpace Weaver membuat file snapshot .zip di folder itusnapshot.

6. Pilih Buat snapshot.

Gunakan konsol untuk memulai simulasi dari snapshot

Untuk memulai simulasi dari snapshot, file snapshot .zip Anda harus ada di bucket Amazon S3 yang dapat diakses simulasi Anda. Simulasi Anda menggunakan izin yang ditentukan dalam peran aplikasi yang Anda pilih saat memulai simulasi. Semua file aplikasi.zip dari simulasi asli harus ada di lokasi yang sama seperti saat snapshot dibuat.

Untuk memulai simulasi dari snapshot

1. Masuk ke Konsol Manajemen AWS dan sambungkan ke [SimSpace Weaver konsol](#).
2. Pilih Simulasi dari panel navigasi.
3. Di bagian atas halaman, pilih Mulai simulasi.
4. Di bawah Pengaturan simulasi, masukkan nama dan deskripsi opsional untuk simulasi Anda. Nama simulasi Anda harus unik di Anda Akun AWS.
5. Untuk metode mulai simulasi, pilih Gunakan snapshot di Amazon S3.
6. Untuk URI Amazon S3 untuk snapshot, masukkan URI Amazon S3 dari file snapshot Anda, atau pilih Browse S3 untuk menelusuri dan memilih file.

⚠ Important

Bucket Amazon S3 harus Wilayah AWS sama dengan simulasi.

7. Untuk peran IAM, pilih peran aplikasi yang akan digunakan simulasi Anda.
8. Untuk Durasi maksimum, masukkan jumlah waktu maksimum yang harus dijalankan oleh sumber daya simulasi Anda. Nilai maksimumnya adalah 14D. Untuk informasi selengkapnya tentang durasi maksimum, lihat [_](#).
9. Di bawah Tag - opsional, pilih Tambahkan tag baru jika Anda ingin menambahkan tag.
10. Pilih Mulai simulasi.

Gunakan AWS CLI untuk bekerja dengan snapshot

Anda dapat menggunakan AWS CLI untuk memanggil SimSpace Weaver APIs dari command prompt. Anda harus AWS CLI menginstal dan mengkonfigurasi dengan benar. Untuk informasi selengkapnya, lihat [Menginstal atau memperbarui AWS CLI versi terbaru](#) di Panduan AWS Command Line Interface Pengguna untuk Versi 2.

Topik

- [Gunakan AWS CLI untuk membuat snapshot](#)
- [Gunakan AWS CLI untuk memulai simulasi dari snapshot](#)

Gunakan AWS CLI untuk membuat snapshot

Untuk membuat snapshot

- Pada prompt perintah, panggil CreateSnapshot API.

```
aws simspaceweaver create-snapshot --simulation simulation-name --destination s3-destination
```

Parameter

simulasi

Nama simulasi yang dimulai. Anda dapat menggunakan `aws simspaceweaver list-simulations` untuk melihat nama dan status simulasi Anda.

tujuan

String yang menentukan bucket Amazon S3 tujuan dan awalan key objek opsional untuk file snapshot Anda. Object key prefix Anda biasanya berupa folder di bucket Anda. SimSpace Weaver membuat snapshot Anda di dalam snapshot folder di tujuan ini.

Important

Bucket Amazon S3 harus Wilayah AWS sama dengan simulasi.

Contoh

```
aws simspaceweaver create-snapshot --simulation
MyProjectSimulation_23-04-29_12_00_00 --destination BucketName=weaver-
myproject-111122223333-artifacts-us-west-2,ObjectKeyPrefix=myFolder
```

Untuk informasi selengkapnya tentang CreateSnapshot API, lihat [CreateSnapshot](#) di Referensi AWS SimSpace Weaver API.

Gunakan AWS CLI untuk memulai simulasi dari snapshot

Untuk memulai simulasi dari snapshot

- Pada prompt perintah, panggil StartSimulation API.

```
aws simspaceweaver start-simulation --name simulation-name --role-arn role-arn --
snapshot-s3-location s3-location
```

Parameter

name

Nama simulasi baru. Nama simulasi harus unik di Akun AWS. Anda dapat menggunakan `aws simspaceweaver list-simulations` untuk melihat nama-nama simulasi yang ada.

peran-arn

Nama Sumber Daya Amazon (ARN) dari peran aplikasi yang akan digunakan simulasi Anda.

snapshot-s3-lokasi

String yang menentukan bucket Amazon S3 dan kunci objek file snapshot Anda.

Important

Bucket Amazon S3 harus Wilayah AWS sama dengan simulasi.

Contoh

```
aws simspaceweaver start-simulation --name MySimulation --role-arn
arn:aws:iam::111122223333:role/weaver-MyProject-app-role --snapshot-s3-location
BucketName=weaver-myproject-111122223333-artifacts-us-west-2,ObjectKey=myFolder/
snapshot/MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Untuk informasi selengkapnya tentang `StartSimulation` API, lihat [StartSimulation](#) di Referensi AWS SimSpace Weaver API.

Pertanyaan yang sering diajukan tentang snapshot

Apakah simulasi saya terus berjalan selama snapshot?

Sumber daya simulasi Anda terus berjalan selama snapshot dan Anda terus menerima biaya penagihan untuk waktu itu. Waktu dihitung untuk durasi maksimum simulasi Anda. Aplikasi Anda tidak menerima kutu saat snapshot sedang berlangsung. Jika status jam Anda `STARTED` saat pembuatan snapshot dimulai, jam Anda akan tetap menunjukkan `STARTED` status. Aplikasi Anda menerima kutu lagi setelah snapshot selesai. Jika status jam Anda adalah `STOPPED` maka status jam Anda akan tetap ada `STOPPED`. Perhatikan bahwa simulasi dengan `STARTED` status berjalan meskipun status jamnya. `STOPPED`

Apa yang terjadi jika snapshot sedang berlangsung dan simulasi saya mencapai durasi maksimumnya?

Simulasi Anda akan menyelesaikan snapshot dan kemudian berhenti segera setelah proses snapshot berakhir (baik berhasil atau tidak berhasil). Kami menyarankan Anda menguji proses snapshot sebelumnya untuk mengetahui berapa lama waktu yang dibutuhkan, ukuran file snapshot yang dapat Anda harapkan, dan apakah itu harus berhasil diselesaikan.

Apa yang terjadi jika saya menghentikan simulasi yang memiliki snapshot sedang berlangsung?

Cuplikan yang sedang berlangsung segera berhenti saat Anda menghentikan simulasi. Itu tidak akan membuat file snapshot.

Bagaimana saya bisa menghentikan snapshot yang sedang berlangsung?

Satu-satunya cara untuk menghentikan snapshot yang sedang berlangsung adalah dengan menghentikan simulasi. Anda tidak dapat memulai ulang simulasi setelah Anda menghentikannya.

Berapa lama waktu yang dibutuhkan untuk menyelesaikan snapshot saya?

Waktu yang dibutuhkan untuk membuat snapshot tergantung pada simulasi Anda. Kami menyarankan Anda menguji proses snapshot sebelumnya untuk mengetahui berapa lama waktu yang dibutuhkan untuk simulasi Anda.

Seberapa besar file snapshot saya?

Ukuran file snapshot tergantung pada simulasi Anda. Kami menyarankan Anda menguji proses snapshot terlebih dahulu untuk mengetahui seberapa besar file tersebut untuk simulasi Anda.

Perpesanan

API perpesanan menyederhanakan aplikasi untuk komunikasi aplikasi dalam simulasi. APIs untuk mengirim dan menerima pesan adalah bagian dari SDK SimSpace Weaver aplikasi. Pesan saat ini menggunakan pendekatan upaya terbaik untuk mengirim dan menerima pesan. SimSpace Weaver mencoba send/receive mengirim pesan pada centang simulasi berikutnya, tetapi tidak ada jaminan waktu pengiriman, pemesanan, atau kedatangan.

Topik

- [Kasus penggunaan untuk pengiriman pesan](#)

- [Menggunakan pesan APIs](#)
- [Kapan menggunakan pesan](#)
- [Kiat saat bekerja dengan perpesanan](#)
- [Kesalahan pesan dan pemecahan masalah](#)

Kasus penggunaan untuk pengiriman pesan

Berkomunikasi antara aplikasi simulasi

Gunakan API perpesanan untuk berkomunikasi antar aplikasi dalam simulasi Anda. Gunakan untuk mengubah status entitas di kejauhan, mengubah perilaku entitas, atau menyiarkan informasi ke seluruh simulasi.

Akui tanda terima pesan

Pesan terkirim berisi informasi tentang pengirim di header pesan. Gunakan informasi ini untuk mengirim kembali balasan pengakuan pada saat menerima pesan.

Meneruskan data yang diterima oleh aplikasi khusus ke aplikasi lain dalam simulasi

Perpesanan bukanlah pengganti cara klien terhubung ke aplikasi khusus yang berjalan SimSpace Weaver. Namun, perpesanan memungkinkan pengguna metode penerusan data dari aplikasi khusus yang menerima data klien ke aplikasi lain yang tidak memiliki koneksi eksternal. Alur pesan juga dapat bekerja secara terbalik, memungkinkan aplikasi tanpa koneksi eksternal untuk meneruskan data ke aplikasi khusus kemudian ke klien.

Menggunakan pesan APIs

Pesan APIs terkandung dalam SDK SimSpace Weaver aplikasi (versi minimum 1.16.0). Pesan didukung dalam C ++, Python, dan integrasi kami dengan Unreal Engine 5 dan Unity.

Ada dua fungsi yang menangani transaksi pesan: `SendMessage` dan `ReceiveMessages`. Semua pesan terkirim berisi tujuan dan muatan. `ReceiveMessagesAPI` menampilkan daftar pesan yang saat ini berada dalam antrean pesan masuk aplikasi.

C++

Kirim pesan

```
AWS_WEAVERRUNTIME_API Result<void> SendMessage(  
    Transaction& txn,  
    const MessagePayload& payload,  
    const MessageEndpoint& destination,  
    MessageDeliveryType deliveryType = MessageDeliveryType::BestEffort  
    ) noexcept;
```

Menerima pesan

```
AWS_WEAVERRUNTIME_API Result<MessageList> ReceiveMessages(  
    Transaction& txn) noexcept;
```

Python

Kirim pesan

```
api.send_message(  
    txn, # Transaction  
    payload, # api.MessagePayload  
    destination, # api.MessageDestination  
    api.MessageDeliveryType.BestEffort # api.MessageDeliveryType  
    )
```

Menerima pesan

```
api.receive_messages(  
    txn, # Transaction  
    ) -> api.MessageList
```

Topik

- [Mengirim pesan](#)
- [Menerima pesan](#)
- [Membalas pengirim](#)

Mengirim pesan

Pesan terdiri dari transaksi (mirip dengan panggilan Weaver API lainnya), payload, dan tujuan.

Muatan pesan

Payload pesan adalah struktur data fleksibel hingga 256 byte. Kami merekomendasikan hal berikut sebagai praktik terbaik untuk membuat muatan pesan Anda.

Untuk membuat payload pesan

1. Buat struktur data (seperti `struct` dalam C++) yang mendefinisikan isi pesan.
2. Buat payload pesan yang berisi nilai yang akan dikirim dalam pesan Anda.
3. Buat `MessagePayload` objek.

Tujuan pesan

Tujuan pesan ditentukan oleh `MessageEndpoint` objek. Ini termasuk tipe endpoint dan endpoint ID. Satu-satunya jenis endpoint yang saat ini didukung adalah `Partition`, yang memungkinkan Anda untuk menangani pesan ke partisi lain dalam simulasi. Endpoint ID adalah ID partisi tujuan target Anda.

Anda hanya dapat memberikan 1 alamat tujuan dalam satu pesan. Buat dan kirim beberapa pesan jika Anda ingin mengirim pesan ke lebih dari 1 partisi secara bersamaan.

Untuk panduan tentang cara menyelesaikan titik akhir pesan dari suatu posisi, lihat [Kiat saat bekerja dengan perpesanan](#).

Kirim pesan

Anda dapat menggunakan `SendMessage` API setelah membuat objek tujuan dan payload.

C++

```
Api::SendMessage(transaction, payload, destination,  
MessageDeliveryType::BestEffort);
```

Python

```
api.send_message(txn, payload, destination, api.MessageDeliveryType.BestEffort)
```

Contoh lengkap pengiriman pesan

Contoh berikut menunjukkan bagaimana Anda dapat membangun dan mengirim pesan generik. Contoh ini mengirimkan 16 pesan individual. Setiap pesan berisi muatan dengan nilai antara 0 dan 15, dan centang simulasi saat ini.

Example

C++

```
// Message struct definition
struct MessageTickAndId
{
    uint32_t id;
    uint32_t tick;
};

Aws::WeaverRuntime::Result<void> SendMessages(Txn& txn) noexcept
{
    // Fetch the destination MessageEndpoint with the endpoint resolver
    WEAVERRUNTIME_TRY(
        Api::MessageEndpoint destination,
        Api::Utils::MessageEndpointResolver::ResolveFromPosition(
            txn,
            "MySpatialSimulation",
            Api::Vector2F32 {231.3, 654.0}
        )
    );
    Log::Info("destination: ", destination);

    WEAVERRUNTIME_TRY(auto tick, Api::CurrentTick(txn));

    uint16_t numSentMessages = 0;
    for (std::size_t i=0; i<16; i++)
    {
        // Create the message that'll be serialized into payload
        MessageTickAndId message {i, tick.value};

        // Create the payload out of the struct
        const Api::MessagePayload& payload = Api::Utils::CreateMessagePayload(
            reinterpret_cast<const std::uint8_t*>(&message),
            sizeof(MessageTickAndId)
        );

        // Send the payload to the destination
    }
}
```

```

    Result<void> result = Api::SendMessage(txn, payload, destination);
    if (result.has_failure())
    {
        // SendMessage has failure modes, log them
        auto error = result.as_failure().error();
        std::cout<< "SendMessage failed, ErrorCode: " << error << std::endl;
        continue;
    }

    numSentMessages++;
}

std::cout << numSentMessages << " messages is sent to endpoint"
    << destination << std::endl;
return Aws::WeaverRuntime::Success();
}

```

Python

```

# Message data class
@dataclasses.dataclass
class MessageTickAndId:
    tick: int = 0
    id: int = 0

# send messages
def _send_messages(self, txn):
    tick = api.current_tick(txn)
    num_messages_to_send = 16

    # Fetch the destination MessageEndpoint with the endpoint resolver
    destination = api.utils.resolve_endpoint_from_domain_name_position(
        txn,
        "MySpatialSimulation",
        pos
    )
    Log.debug("Destination_endpoint = %s", destination_endpoint)

    for id in range(num_messages_to_send):
        # Message struct that'll be serialized into payload
        message_tick_and_id = MessageTickAndId(id = id, tick = tick.value)

        # Create the payload out of the struct

```

```

    message_tick_and_id_data = struct.pack(
        '<ii',
        message_tick_and_id.id,
        message_tick_and_id.tick
    )
    payload = api.MessagePayload(list(message_tick_and_id_data))

    # Send the payload to the destination
    Log.debug("Sending message: %s, endpoint: %s",
        message_tick_and_id,
        destination
    )
    api.send_message(
        txn,
        payload,
        destination,
        api.MessageDeliveryType.BestEffort
    )

    Log.info("Sent %s messages to %s", num_messages_to_send, destination)
    return True

```

Menerima pesan

SimSpace Weaver mengirimkan pesan ke antrian pesan masuk partisi. Gunakan `ReceiveMessages` API untuk mendapatkan `MessageList` objek yang berisi pesan dari antrian. Proses setiap pesan dengan `ExtractMessage` API untuk mendapatkan data pesan.

Example

C++

```

Result<void> ReceiveMessages(Txn& txn) noexcept
{
    // Fetch all the messages sent to the partition owned by the app
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
    std::cout << "Received" << messages.messages.size() << " messages" << std::endl;
    for (Api::Message& message : messages.messages)
    {
        std::cout << "Received message: " << message << std::endl;

        // Deserialize payload to the message struct
    }
}

```

```

    const MessageTickAndId& receivedMessage
        = Api::Utils::ExtractMessage<MessageTickAndId>(message);
    std::cout << "Received MessageTickAndId, Id: " << receivedMessage.id
        <<"", Tick: " << receivedMessage.tick << std::endl;
}

return Aws::WeaverRuntime::Success();
}

```

Python

```

# process incoming messages
def _process_incoming_messages(self, txn):
    messages = api.receive_messages(txn)
    for message in messages:
        payload_list = message.payload.data
        payload_bytes = bytes(payload_list)
        message_tick_and_id_data_struct
            = MessageTickAndId(*struct.unpack('<ii', payload_bytes))

        Log.debug("Received message. Header: %s, message: %s",
            message.header, message_tick_and_id_data_struct)

    Log.info("Received %s messages", len(messages))
    return True

```

Membalas pengirim

Setiap pesan yang diterima berisi header pesan dengan informasi tentang pengirim asli pesan. Anda dapat menggunakan `message.header.source_endpoint` untuk mengirim balasan.

Example

C++

```

Result<void> ReceiveMessages(Txn& txn) noexcept
{
    // Fetch all the messages sent to the partition owned by the app
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
    std::cout << "Received" << messages.messages.size() << " messages" << std::endl;
    for (Api::Message& message : messages.messages)
    {

```

```

std::cout << "Received message: " << message << std::endl;

// Deserialize payload to the message struct
const MessageTickAndId& receivedMessage
    = Api::Utils::ExtractMessage<MessageTickAndId>(message);
std::cout << "Received MessageTickAndId, Id: " << receivedMessage.id
    <<" , Tick: " << receivedMessage.tick << std::endl;

// Get the sender endpoint and payload to bounce the message back
Api::MessageEndpoint& sender = message.header.source_endpoint;
Api::MessagePayload& payload = message.payload;
Api::SendMessage(txn, payload, sender);
}

return Aws::WeaverRuntime::Success();
}

```

Python

```

# process incoming messages
def _process_incoming_messages(self, txn):
    messages = api.receive_messages(txn)
    for message in messages:
        payload_list = message.payload.data
        payload_bytes = bytes(payload_list)
        message_tick_and_id_data_struct
            = MessageTickAndId(*struct.unpack('<ii', payload_bytes))

        Log.debug("Received message. Header: %s, message: %s",
            message.header, message_tick_and_id_data_struct)
        # Get the sender endpoint and payload
        # to bounce the message back
        sender = message.header.source_endpoint
        payload = payload_list
        api.send_message(
            txn,
            payload_list,
            sender,
            api.MessageDeliveryType.BestEffort

        Log.info("Received %s messages", len(messages))
    return True

```

Kapan menggunakan pesan

Messaging in SimSpace Weaver menawarkan pola lain untuk bertukar informasi antara aplikasi simulasi. Langganan menyediakan mekanisme tarik untuk membaca data dari aplikasi atau area simulasi tertentu; pesan menyediakan mekanisme push untuk mengirim data ke aplikasi atau area simulasi tertentu.

Di bawah ini adalah dua kasus penggunaan di mana lebih membantu untuk mendorong data menggunakan pesan daripada menarik atau membaca data melalui langganan.

Example 1: Mengirim perintah ke aplikasi lain untuk mengubah posisi entitas

```
// Message struct definition
struct MessageMoveEntity
{
    uint64_t entityId;
    std::array<float, 3> destinationPos;
};

// Create the message
MessageMoveEntity message {45, {236.67, 826.22, 0.0} };

// Create the payload out of the struct
const Api::MessagePayload& payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const std::uint8_t*>(&message),
    sizeof(MessageTickAndId)
);

// Grab the MessageEndpoint of the recipient app.
Api::MessageEndpoint destination = ...

// One way is to resolve it from the domain name and position
WEAVERRUNTIME_TRY(
    Api::MessageEndpoint destination,
    Api::Utils::MessageEndpointResolver::ResolveFromPosition(
        txn,
        "MySpatialSimulation",
        Api::Vector2F32 {200.0, 100.0}
    )
);

// Then send the message
```

```
Api::SendMessage(txn, payload, destination);
```

Di sisi penerima, aplikasi memperbarui posisi entitas dan menuliskannya ke State Fabric.

```
Result<void> ReceiveMessages(Txn& txn) noexcept
{
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
    for (Api::Message& message : messages.messages)
    {
        std::cout << "Received message: " << message << std::endl;
        // Deserialize payload to the message struct
        const MessageMoveEntity& receivedMessage
            = Api::Utils::ExtractMessage<MessageMoveEntity>(message);

        ProcessMessage(txn, receivedMessage);
    }

    return Aws::WeaverRuntime::Success();
}

void ProcessMessage(Txn& txn, const MessageMoveEntity& receivedMessage)
{
    // Get the entity corresponding to the entityId
    Entity entity = EntityFromEntityId (receivedMessage.entityId);

    // Update the position and write to StateFabric
    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        txn,
        entity,
        k_vector3f32TypeId, // type id of the entity
        reinterpret_cast<std::int8_t*>(&receivedMessage.destinationPos),
        sizeof(receivedMessage.destinationPos)));
}
}
```

Example 2: Mengirim pesan entitas buat ke aplikasi spasial

```
struct WeaverMessage
{
    const Aws::WeaverRuntime::Api::TypeId messageTypeId;
};

const Aws::WeaverRuntime::Api::TypeId k_createEntityMessageTypeId = { 1 };
```

```

struct CreateEntityMessage : WeaverMessage
{
    const Vector3 position;
    const Aws::WeaverRuntime::Api::TypeId typeId;
};

CreateEntityMessage messageData {
    k_createEntityMessageTypeId,
    Vector3{ position.GetX(), position.GetY(), position.GetZ() },
    Api::TypeId { 0 }
}

WEAVERRUNTIME_TRY(Api::MessageEndpoint destination,
    Api::Utils::MessageEndpointResolver::ResolveFromPosition(
        transaction, "MySpatialDomain", DemoFramework::ToVector2F32(position)
    ));

Api::MessagePayload payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const uint8_t*>(&messageData),
    sizeof(CreateEntityMessage));

Api::SendMessage(transaction, payload, destination);

```

Di sisi penerima, aplikasi membuat entitas baru di State Fabric dan memperbarui posisinya.

```

Result<void> ReceiveMessages(Txn& txn) noexcept
{
    WEAVERRUNTIME_TRY(auto messageList, Api::ReceiveMessages(transaction));
    WEAVERRUNTIME_TRY(auto tick, Api::CurrentTick(transaction));
    for (auto& message : messageList.messages)
    {
        // cast to base WeaverMessage type to determine MessageTypeId
        WeaverMessage weaverMessageBase =
        Api::Utils::ExtractMessage<WeaverMessage>(message);
        if (weaverMessageBase.messageTypeId == k_createEntityMessageTypeId)
        {
            CreateEntityMessage createEntityMessageData =
                Api::Utils::ExtractMessage<CreateEntityMessage>(message);
            CreateActorFromMessage(transaction, createEntityMessageData);
        }
        else if (weaverMessageBase.messageTypeId == k_tickAndIdMessageTypeId)

```

```

        {
            ...
        }
    }
}

void ProcessMessage(Txn& txn, const CreateEntityMessage& receivedMessage)
{
    // Create entity
    WEAVERRUNTIME_TRY(
        Api::Entity entity,
        Api::CreateEntity(transaction, receivedMessage.typeId)
    );

    // Update the position and write to StateFabric
    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        transaction,
        entity,
        receivedMessage.typeId,
        reinterpret_cast<std::int8_t*>(&receivedMessage.position),
        sizeof(receivedMessage.position)));
}

```

Kiat saat bekerja dengan perpesanan

Menyelesaikan titik akhir dari posisi atau nama aplikasi

Anda dapat menggunakan `AllPartitions` fungsi untuk mendapatkan batas spasial dan ID domain yang Anda butuhkan untuk menentukan partisi pesan IDs dan tujuan pesan. Namun, jika Anda tahu posisi yang ingin Anda pesan, tetapi bukan ID partisi, Anda dapat menggunakan `MessageEndpointResolver` fungsi tersebut.

```

/**
 * Resolves MessageEndpoint's from various inputs
 */
class MessageEndpointResolver
{
public:
    /**
     * Resolves MessageEndpoint from position information
     */
    Result<MessageEndpoint> ResolveEndpointFromPosition(

```

```

        const DomainId& domainId,
        const weaver_vec3_f32_t& pos);

/**
 * Resolves MessageEndpoint from custom app name
 */
Result<MessageEndpoint> ResolveEndpointFromCustomAppName(
    const DomainId& domainId,
    const char* agentName);
};

```

Serialisasi dan deserialisasi payload pesan

Anda dapat menggunakan fungsi berikut untuk membuat dan membaca muatan pesan. Untuk informasi selengkapnya, lihat `MessagingUtils.h` di pustaka SDK aplikasi di sistem lokal Anda.

```

/**
 * Utility function to create MessagePayload from a custom type
 *
 * @return The @c MessagePayload.
 */
template <class T>
AWS_WEAVERRUNTIME_API MessagePayload CreateMessagePayload(const T& message) noexcept
{
    const std::uint8_t* raw_data = reinterpret_cast<const std::uint8_t*>(&message);

    MessagePayload payload;
    std::move(raw_data, raw_data + sizeof(T), std::back_inserter(payload.data));

    return payload;
}

/**
 * Utility function to convert MessagePayload to custom type
 */
template <class T>
AWS_WEAVERRUNTIME_API T ExtractMessage(const MessagePayload& payload) noexcept
{
    return *reinterpret_cast<const T*>(payload.data.data());
}

```

Kesalahan pesan dan pemecahan masalah

Anda mungkin mengalami kesalahan berikut saat menggunakan pesan APIs.

Kesalahan resolusi titik akhir

Kesalahan ini dapat terjadi sebelum aplikasi mengirim pesan.

Pemeriksaan nama domain

Mengirim pesan ke titik akhir yang tidak valid menghasilkan kesalahan berikut:

```
ManifoldError::InvalidArgument {"No DomainId found for the given domain name" }
```

Ini dapat terjadi ketika Anda mencoba mengirim pesan ke aplikasi khusus dan aplikasi khusus itu belum bergabung dengan simulasi. Gunakan `DescribeSimulation` API untuk memastikan aplikasi kustom Anda telah diluncurkan sebelum Anda mengirim pesan ke sana. Perilaku ini sama di dalam SimSpace Weaver Local dan AWS Cloud.

Pemeriksaan posisi

Mencoba menyelesaikan titik akhir dengan nama domain yang valid tetapi posisi yang tidak valid menghasilkan kesalahan berikut.

```
ManifoldError::InvalidArgument {"Could not resolve endpoint from domain : DomainId { value: domain-id } and position: Vector2F32 { x: x-position, y: y-position}" }
```

Sebaiknya gunakan `MessageUtils` pustaka `MessageEndpointResolver` di dalam aplikasi yang terdapat dalam SDK SimSpace Weaver aplikasi.

Kesalahan pengiriman pesan

Kesalahan berikut dapat terjadi saat aplikasi mengirim pesan.

Batas pengiriman pesan per aplikasi, per centang, terlampaui

Batas saat ini untuk jumlah pesan yang dapat dikirim per aplikasi per tick simulasi adalah 128. Panggilan berikutnya pada centang yang sama akan gagal dengan kesalahan berikut:

```
ManifoldError::CapacityExceeded {"At Max Outgoing Message capacity: {}", 128}
```

SimSpace Weaver mencoba mengirim pesan yang tidak terkirim pada centang berikutnya. Turunkan frekuensi kirim untuk mengatasi masalah ini. Gabungkan muatan pesan yang lebih kecil dari batas 256 byte untuk menurunkan jumlah pesan keluar.

Perilaku ini sama di dalam SimSpace Weaver Local dan di AWS Cloud.

Batas ukuran muatan pesan terlampaui

Batas saat ini untuk ukuran payload pesan adalah 256 byte di keduanya SimSpace Weaver Local dan di. AWS Cloud Mengirim pesan dengan muatan yang lebih besar dari 256 byte menghasilkan kesalahan berikut:

```
ManifoldError::CapacityExceeded {"Message data too large! Max size: {}", 256}
```

SimSpace Weaver memeriksa setiap pesan dan hanya menolak pesan yang melebihi batas. Misalnya, jika aplikasi Anda mencoba mengirim 10 pesan dan 1 gagal dicentang, hanya 1 pesan yang ditolak. SimSpace Weaver mengirim 9 pesan lainnya.

Perilaku ini sama di dalam SimSpace Weaver Local dan AWS Cloud.

Tujuan adalah sama dengan sumber

Aplikasi tidak dapat mengirim pesan ke partisi yang mereka miliki. Anda mendapatkan kesalahan berikut jika aplikasi mengirim pesan ke partisi yang dimilikinya.

```
ManifoldError::InvalidArgument { "Destination is the same as source" }
```

Perilaku ini sama di dalam SimSpace Weaver Local dan AWS Cloud.

Pesan upaya terbaik

SimSpace Weaver tidak menjamin pengiriman pesan. Layanan akan mencoba menyelesaikan pengiriman pesan pada centang simulasi berikutnya, tetapi pesan mungkin hilang atau tertunda.

Praktik terbaik saat bekerja dengan SimSpace Weaver

Kami merekomendasikan praktik terbaik berikut saat bekerja dengan SimSpace Weaver.

Topik

- [Mengatur alarm penagihan](#)
- [Gunakan SimSpace Weaver Local](#)
- [Hentikan simulasi yang tidak Anda butuhkan](#)
- [Hapus sumber daya yang tidak Anda butuhkan](#)
- [Memiliki cadangan](#)

Mengatur alarm penagihan

Sangat mudah untuk menyediakan sumber daya AWS dan membiarkannya berjalan sepanjang waktu, bahkan ketika mereka tidak dibutuhkan lagi. Hal ini dapat mengakibatkan biaya pelarian yang bisa menjadi kejutan ketika Anda mendapatkan tagihan Anda. Anda dapat mengonfigurasi alarm di Amazon CloudWatch yang akan memicu dan memberi tahu Anda ketika biaya Anda melebihi ambang batas yang Anda tetapkan. Anda dapat memeriksa biaya Anda menggunakan alat manajemen biaya. Untuk informasi lebih lanjut, lihat:

- [Buat alarm penagihan untuk memantau perkiraan biaya AWS](#)
- [Apa itu AWS Cost Management](#)

Gunakan SimSpace Weaver Local

Kami menyarankan Anda menggunakan SimSpace Weaver Local untuk mengembangkan dan menguji simulasi Anda sebelum mengunggahnya ke SimSpace Weaver layanan di AWS Cloud. Manfaat mengembangkan dengan SimSpace Weaver Local meliputi:

- Tidak perlu menunggu upload besar
- Tidak ada batasan jumlah simulasi lokal yang dapat Anda buat
- Anda tidak dikenakan biaya untuk waktu komputasi di komputer lokal Anda
- Akses langsung ke output konsol dari aplikasi Anda

- Ubah, bangun kembali, dan mulai ulang simulasi lokal Anda tanpa harus membuatnya ulang di AWS Cloud

Hentikan simulasi yang tidak Anda butuhkan

Anda mendapatkan biaya penagihan untuk simulasi saat sedang berjalan. Anda harus menghentikan simulasi untuk berhenti mendapatkan biaya untuk itu. Menjalankan simulasi juga dihitung terhadap kuota Anda untuk jumlah simulasi maksimum. Simulasi berjalan yang telah dikonfigurasi logging juga dapat menghasilkan log dalam jumlah besar, yang juga Anda dapatkan biaya penagihan. Anda harus menghentikan simulasi apa pun yang tidak Anda perlukan untuk berhenti mendapatkan biaya tambahan.

Important

Menghentikan jam simulasi tidak menghentikan simulasi, jam hanya berhenti menerbitkan kutu ke aplikasi Anda. Anda tidak dapat memulai ulang simulasi setelah Anda menghentikannya.

Hapus sumber daya yang tidak Anda butuhkan

Setiap simulasi yang Anda buat SimSpace Weaver juga menciptakan sumber daya di AWS layanan lain. Anda bisa mendapatkan biaya penagihan untuk sumber daya dan data di layanan lain ini. Simulasi yang berjalan dan gagal dihitung terhadap kuota Anda untuk jumlah simulasi maksimum. Anda harus menghapus simulasi gagal yang tidak diperlukan sehingga Anda dapat memulai simulasi baru. Saat Anda menghapus simulasi, sumber daya untuk simulasi Anda yang ada di AWS layanan lain mungkin tidak akan dihapus. Misalnya, data log simulasi apa pun di Amazon CloudWatch Logs akan tetap ada sampai Anda menghapusnya. Anda akan mendapatkan biaya penagihan untuk data log tersebut. Anda harus membersihkan semua sumber daya terkait untuk simulasi Anda jika Anda tidak membutuhkannya lagi.

Memiliki cadangan

Merupakan ide bagus untuk memiliki cadangan dan rencana cadangan untuk semuanya. Anda tidak boleh berasumsi bahwa hanya karena data Anda ada di AWS dalamnya Anda tidak perlu mencadangkannya. Anda harus membuat sistem Anda sendiri jika Anda perlu mencadangkan

status simulasi Anda. Pertimbangkan untuk menggunakan beberapa Wilayah AWS dan memiliki rencana untuk dapat dengan cepat mengalihkan beban kerja produksi Anda ke yang lain Wilayah AWS jika perlu. Untuk informasi lebih lanjut tentang dukungan Wilayah AWS itu SimSpace Weaver, lihat [SimSpace Titik akhir dan kuota Weaver](#).

Keamanan di AWS SimSpace Weaver

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program AWS Kepatuhan Program AWS Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku AWS SimSpace Weaver, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#) .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan SimSpace Weaver. Topik berikut menunjukkan cara mengonfigurasi SimSpace Weaver untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan SimSpace Weaver sumber daya Anda.

Topik

- [Perlindungan data di AWS SimSpace Weaver](#)
- [Identity and Access Management untuk AWS SimSpace Weaver](#)
- [Pencatatan dan pemantauan peristiwa keamanan di AWS SimSpace Weaver](#)
- [Validasi Kepatuhan untuk AWS SimSpace Weaver](#)
- [Ketahanan di AWS SimSpace Weaver](#)
- [Keamanan Infrastruktur di AWS SimSpace Weaver](#)
- [Analisis konfigurasi dan kerentanan di AWS SimSpace Weaver](#)
- [Praktik terbaik keamanan untuk SimSpace Weaver](#)

Perlindungan data di AWS SimSpace Weaver

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS SimSpace Weaver. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan SimSpace Weaver atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi saat diam

Data dianggap diam ketika berada di penyimpanan data non-volatile (persisten), seperti disk. Data yang terletak di penyimpanan data yang mudah menguap, seperti memori dan register, tidak dianggap diam.

Saat Anda menggunakan SimSpace Weaver, satu-satunya data yang diam adalah:

- Aplikasi dan skema yang Anda unggah ke Amazon Simple Storage Service (Amazon S3)
- Data log simulasi disimpan di Amazon CloudWatch

Data lain yang SimSpace Weaver digunakan secara internal tidak bertahan setelah Anda menghentikan simulasi Anda.

Untuk mempelajari cara mengenkripsi data Anda saat istirahat, lihat:

- [Enkripsi data Anda di Amazon S3](#)
- [Enkripsi data log Anda](#)

Enkripsi saat bergerak

Koneksi Anda ke SimSpace Weaver API melalui AWS Command Line Interface (AWS CLI), AWS SDK, dan SDK SimSpace Weaver aplikasi, gunakan enkripsi TLS dengan proses [penandatanganan Signature Version 4](#). AWS mengelola otentikasi menggunakan kebijakan akses yang ditentukan oleh IAM untuk kredensial keamanan yang Anda gunakan untuk menyambung.

Secara internal, SimSpace Weaver menggunakan TLS untuk terhubung ke AWS layanan lain yang digunakannya.

Important

Komunikasi antara aplikasi Anda dan klien mereka tidak melibatkan SimSpace Weaver. Anda bertanggung jawab untuk mengenkripsi komunikasi dengan klien simulasi, jika diperlukan.

Kami menyarankan Anda membuat solusi untuk mengenkripsi semua data dalam perjalanan di seluruh koneksi klien.

Untuk mempelajari lebih lanjut tentang AWS layanan yang dapat mendukung solusi enkripsi Anda, lihat [Blog AWS Keamanan](#).

Privasi lalu lintas antar jaringan

SimSpace Weaver sumber daya komputasi berada dalam 1 VPC Amazon yang dibagikan oleh semua pelanggan. SimSpace Weaver Semua lalu lintas SimSpace Weaver layanan internal tetap berada dalam AWS jaringan dan tidak melakukan perjalanan melalui internet. Komunikasi antara klien simulasi dan aplikasi Anda berjalan di internet.

Identity and Access Management untuk AWS SimSpace Weaver

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. SimSpace Weaver IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana AWS SimSpace Weaver bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#)
- [Izin yang SimSpace Weaver dibuat untuk Anda](#)
- [Pencegahan "confused deputy" lintas layanan](#)
- [Memecahkan masalah AWS SimSpace Weaver identitas dan akses](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Memecahkan masalah AWS SimSpace Weaver identitas dan akses](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana AWS SimSpace Weaver bekerja dengan IAM](#))
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#))

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensi dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensial sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukannya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .

- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana AWS SimSpace Weaver bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses SimSpace Weaver, pelajari fitur IAM yang tersedia untuk digunakan. SimSpace Weaver

Fitur IAM yang dapat Anda gunakan AWS SimSpace Weaver

Fitur IAM	SimSpace Weaver dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya

Fitur IAM	SimSpace Weaver dukungan
Izin principal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara SimSpace Weaver dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk SimSpace Weaver

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk SimSpace Weaver

Untuk melihat contoh kebijakan SimSpace Weaver berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#)

Kebijakan berbasis sumber daya dalam SimSpace Weaver

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan

kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk SimSpace Weaver

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar SimSpace Weaver tindakan, lihat [Tindakan yang ditentukan oleh AWS SimSpace Weaver](#) dalam Referensi Otorisasi Layanan.

Tindakan kebijakan SimSpace Weaver menggunakan awalan berikut sebelum tindakan:

```
simspaceweaver
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [  
  "simspaceweaver:action1",  
  "simspaceweaver:action2"  
]
```

Untuk melihat contoh kebijakan SimSpace Weaver berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#)

Sumber daya kebijakan untuk SimSpace Weaver

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

Untuk melihat daftar jenis sumber daya dan jenis SimSpace Weaver sumber daya ARNs, lihat [Sumber daya yang ditentukan oleh AWS SimSpace Weaver](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari dengan tindakan mana Anda dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS SimSpace Weaver](#).

Untuk melihat contoh kebijakan SimSpace Weaver berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#)

Kunci kondisi kebijakan untuk SimSpace Weaver

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci SimSpace Weaver kondisi, lihat [Kunci kondisi untuk AWS SimSpace Weaver](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS SimSpace Weaver](#).

Untuk melihat contoh kebijakan SimSpace Weaver berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver](#)

Daftar kontrol akses (ACLs) di SimSpace Weaver

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan SimSpace Weaver

Mendukung ABAC (tanda dalam kebijakan): Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Tentukan izin dengan otorisasi ABAC](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

Menggunakan kredensi sementara dengan SimSpace Weaver

Mendukung kredensial sementara: Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#) dan [Layanan AWS yang berfungsi dengan IAM](#) dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk SimSpace Weaver

Mendukung sesi akses terusan (FAS): Ya

Sesi akses terusan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses terusan](#).

Peran layanan untuk SimSpace Weaver

Mendukung peran layanan: Ya

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Warning

Mengubah izin untuk peran layanan dapat merusak SimSpace Weaver fungsionalitas. Edit peran layanan hanya jika SimSpace Weaver memberikan panduan untuk melakukannya.

Skrip SDK SimSpace Weaver aplikasi menggunakan CloudFormation templat untuk membuat sumber daya di AWS layanan lain untuk mendukung simulasi Anda. Salah satu sumber daya ini adalah peran aplikasi untuk simulasi Anda. SimSpace Weaver mengasumsikan peran aplikasi untuk melakukan tindakan atas nama Anda, seperti menulis data log ke CloudWatch Log. Akun AWS Untuk informasi selengkapnya tentang peran aplikasi, lihat [Izin yang SimSpace Weaver dibuat untuk Anda](#).

Peran terkait layanan untuk SimSpace Weaver

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#). Cari layanan dalam tabel yang memiliki Yes di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk AWS SimSpace Weaver

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya SimSpace Weaver. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM \(konsol\) di Panduan Pengguna IAM](#).

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh SimSpace Weaver, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk AWS SimSpace Weaver](#) dalam Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan SimSpace Weaver konsol](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)
- [Memungkinkan pengguna untuk membuat dan menjalankan simulasi](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus SimSpace Weaver sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [Kebijakan yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi

selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan dengan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

Menggunakan SimSpace Weaver konsol

Untuk mengakses AWS SimSpace Weaver konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang SimSpace Weaver sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan SimSpace Weaver konsol, lampirkan juga kebijakan SimSpace Weaver *ConsoleAccess* atau *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

```

    ]
  }

```

Memungkinkan pengguna untuk membuat dan menjalankan simulasi

Contoh kebijakan IAM ini memberikan izin dasar yang diperlukan untuk membuat dan menjalankan simulasi di SimSpace Weaver

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateAndRunSimulations",
      "Effect": "Allow",
      "Action": [
        "simspaceweaver:*",
        "iam:GetRole",
        "iam:ListRoles",
        "iam:CreateRole",
        "iam>DeleteRole",
        "iam:UpdateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "iam:GetRolePolicy",
        "iam>DeleteRolePolicy",
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy",
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutEncryptionConfiguration",
        "s3>DeleteBucket",
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation:DescribeStacks"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PassAppRoleToSimSpaceWeaver",
      "Effect": "Allow",

```

```
        "Action": "iam:PassRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": "simspaceweaver.amazonaws.com"
            }
        }
    }
]
```

Izin yang SimSpace Weaver dibuat untuk Anda

Saat Anda membuat SimSpace Weaver proyek, layanan akan membuat peran AWS Identity and Access Management (IAM) dengan nama `weaver-project-name-app-role` dan kebijakan kepercayaan IAM. Kebijakan kepercayaan memungkinkan SimSpace Weaver untuk mengambil peran sehingga dapat melakukan operasi untuk Anda.

Kebijakan izin peran aplikasi

Peran aplikasi simulasi memiliki kebijakan izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*"
    }
  ],
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:PutObject",
    "s3:GetObject"
  ],
  "Resource": "*"
}
```

Kebijakan kepercayaan peran aplikasi

SimSpace Weaver menambahkan hubungan kepercayaan ke peran aplikasi simulasi sebagai [kebijakan kepercayaan](#). SimSpace Weaver membuat kebijakan kepercayaan untuk setiap simulasi, mirip dengan contoh berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "simspaceweaver.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MySimName*"
        }
      }
    }
  ]
}
```

Note

Dalam contoh ini, nomor akun adalah 111122223333 dan nama simulasi adalah MySimName. Nilai-nilai ini berbeda dalam kebijakan kepercayaan Anda.

Pencegahan "confused deputy" lintas layanan

[Masalah deputi yang membingungkan](#) adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat menipu entitas yang lebih istimewa untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan principal layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan dalam kebijakan sumber daya untuk membatasi izin yang AWS SimSpace Weaver memberikan layanan lain ke sumber daya. Jika `aws:SourceArn` nilai tidak berisi ID akun, seperti bucket Amazon S3 Nama Sumber Daya Amazon (ARN), Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Nilai `aws:SourceArn` harus menggunakan ARN ekstensi.

Cara paling efektif untuk melindungi dari masalah "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap ekstensi atau jika Anda menentukan beberapa ekstensi, gunakan kunci konteks kondisi `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:simspaceweaver:*:111122223333:*`.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan SimSpace Weaver untuk mencegah masalah wakil yang membingungkan. Kebijakan ini hanya akan mengizinkan SimSpace Weaver untuk mengambil peran ketika permintaan berasal dari akun sumber yang ditentukan, dan dilengkapi dengan ARN yang ditentukan. Dalam hal ini, hanya SimSpace Weaver dapat mengambil peran untuk permintaan dari simulasi di akun pemohon sendiri (`111122223333`), dan hanya di Region tertentu (`us-west-2`).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/*"
        }
      }
    }
  ]
}
```

Cara yang lebih aman untuk menulis kebijakan ini adalah dengan memasukkan nama simulasi dalam `aws:SourceArn`, seperti yang ditunjukkan pada contoh berikut, yang membatasi kebijakan untuk simulasi bernama: `MyProjectSimulation_22-10-04_22_10_15`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/*"
        }
      }
    }
  ]
}
```

```

    ]
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "111122223333"
    },
    "StringLike": {
      "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation"
    }
  }
}
]
}

```

Jika Anda secara `aws:SourceArn` eksplisit menyertakan nomor akun, Anda dapat mengabaikan pengujian `Condition` elemen untuk `aws:SourceAccount` (lihat [Panduan Pengguna IAM](#) untuk informasi selengkapnya), seperti dalam kebijakan sederhana berikut ini:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation"
        }
      }
    }
  ]
}

```

```
}
```

Memecahkan masalah AWS SimSpace Weaver identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan SimSpace Weaver dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di SimSpace Weaver](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin melihat access key saya](#)
- [Saya seorang administrator dan ingin mengizinkan orang lain mengakses SimSpace Weaver](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses SimSpace Weaver sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di SimSpace Weaver

Jika Konsol Manajemen AWS memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberikan nama pengguna dan kata sandi Anda.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya fiktif `my-example-widget`, tetapi tidak memiliki izin fiktif `simspaceweaver:GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
simspaceweaver:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia dapat mengakses `my-example-widget` menggunakan `simspaceweaver:GetWidget` tindakan.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran SimSpace Weaver.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di SimSpace Weaver. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin melihat access key saya

Setelah membuat access key pengguna IAM, Anda dapat melihat access key ID Anda setiap saat. Namun, Anda tidak dapat melihat secret access key Anda lagi. Jika Anda kehilangan secret key, Anda harus membuat pasangan access key baru.

Access key terdiri dari dua bagian: access key ID (misalnya, `AKIAIOSFODNN7EXAMPLE`) dan secret access key (misalnya, `wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Seperti nama pengguna dan kata sandi, Anda harus menggunakan access key ID dan secret access key sekaligus untuk mengautentikasi permintaan Anda. Kelola access key Anda seaman nama pengguna dan kata sandi Anda.

Important

Jangan memberikan access key Anda kepada pihak ke tiga, bahkan untuk membantu [menemukan ID pengguna kanonis Anda](#). Dengan melakukan ini, Anda mungkin memberi seseorang akses permanen ke Akun AWS Anda.

Saat Anda membuat pasangan access key, Anda diminta menyimpan access key ID dan secret access key di lokasi yang aman. secret access key hanya tersedia saat Anda membuatnya. Jika

Anda kehilangan secret access key Anda, Anda harus menambahkan access key baru ke pengguna IAM Anda. Anda dapat memiliki maksimum dua access key. Jika Anda sudah memiliki dua, Anda harus menghapus satu pasangan kunci sebelum membuat pasangan baru. Untuk melihat instruksi, lihat [Mengelola access keys](#) di Panduan Pengguna IAM.

Saya seorang administrator dan ingin mengizinkan orang lain mengakses SimSpace Weaver

Untuk memungkinkan orang lain mengakses SimSpace Weaver, Anda harus memberikan izin kepada orang atau aplikasi yang membutuhkan akses. Jika Anda menggunakan AWS IAM Identity Center untuk mengelola orang dan aplikasi, Anda menetapkan set izin kepada pengguna atau grup untuk menentukan tingkat akses mereka. Set izin secara otomatis membuat dan menetapkan kebijakan IAM ke peran IAM yang terkait dengan orang atau aplikasi. Untuk informasi selengkapnya, lihat [Set izin](#) di Panduan AWS IAM Identity Center Pengguna.

Jika Anda tidak menggunakan IAM Identity Center, Anda harus membuat entitas IAM (pengguna atau peran) untuk orang atau aplikasi yang membutuhkan akses. Anda kemudian harus melampirkan kebijakan yang memberi mereka izin yang tepat di SimSpace Weaver. Setelah izin diberikan, berikan kredensialnya kepada pengguna atau pengembang aplikasi. Mereka akan menggunakan kredensi tersebut untuk mengakses. AWS Untuk mempelajari selengkapnya tentang membuat pengguna, grup, kebijakan, dan izin IAM, lihat [Identitas dan Kebijakan IAM dan izin di IAM di Panduan Pengguna IAM](#).

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses SimSpace Weaver sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah SimSpace Weaver mendukung fitur-fitur ini, lihat [Bagaimana AWS SimSpace Weaver bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM](#).

Pencatatan dan pemantauan peristiwa keamanan di AWS SimSpace Weaver

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja SimSpace Weaver dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi.

AWS dan SimSpace Weaver menyediakan beberapa alat untuk memantau sumber daya simulasi Anda dan menanggapi insiden potensial.

Log di Amazon CloudWatch

SimSpace Weaver menyimpan lognya di CloudWatch. Anda dapat menggunakan log ini untuk memantau peristiwa dalam simulasi Anda (seperti memulai dan menghentikan aplikasi) serta untuk debugging. Untuk informasi selengkapnya, lihat [SimSpace Weaver log di Amazon CloudWatch Log](#).

CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm Amazon, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, notifikasi akan dikirim ke topik Amazon SNS atau kebijakan Auto AWS Scaling. CloudWatch alarm dipicu ketika statusnya berubah dan dipertahankan untuk sejumlah periode tertentu, bukan dengan berada dalam keadaan tertentu. Untuk informasi selengkapnya, lihat [Pemantauan SimSpace Weaver dengan Amazon CloudWatch](#).

AWS CloudTrail Log

CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di SimSpace Weaver. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat SimSpace Weaver, alamat IP dari mana permintaan dibuat,

siapa yang membuat permintaan, kapan dibuat, dan detail tambahan. Untuk informasi selengkapnya, lihat [Logging panggilan AWS SimSpace Weaver API menggunakan AWS CloudTrail](#).

Validasi Kepatuhan untuk AWS SimSpace Weaver

SimSpace Weaver tidak dalam lingkup program AWS kepatuhan apa pun.

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS layanan lain sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat [Dokumentasi AWS Keamanan](#).

Ketahanan di AWS SimSpace Weaver

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Keamanan Infrastruktur di AWS SimSpace Weaver

Sebagai layanan terkelola, AWS SimSpace Weaver dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses SimSpace Weaver melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Model keamanan konektivitas jaringan

Simulasi Anda berjalan pada instans komputasi dalam VPC Amazon yang terletak di dalam Wilayah yang AWS Anda pilih. Amazon VPC adalah jaringan virtual di AWS Cloud, yang mengisolasi infrastruktur berdasarkan beban kerja atau entitas organisasi. Komunikasi antara instance komputasi dalam VPC Amazon tetap berada dalam AWS jaringan dan tidak melakukan perjalanan melalui internet. Beberapa komunikasi layanan internal melintasi internet, dan dienkripsi. Simulasi untuk semua pelanggan yang berjalan di AWS Wilayah yang sama berbagi VPC Amazon yang sama. Simulasi untuk pelanggan yang berbeda menggunakan instans komputasi terpisah dalam VPC Amazon yang sama.

Komunikasi antara klien simulasi Anda dan simulasi Anda berjalan dalam SimSpace Weaver perjalanan melalui internet. SimSpace Weaver tidak menangani koneksi ini. Adalah tanggung jawab Anda untuk mengamankan koneksi klien Anda.

Koneksi Anda ke SimSpace Weaver layanan melintasi internet dan dienkripsi. Ini termasuk koneksi menggunakan Konsol Manajemen AWS, AWS Command Line Interface (AWS CLI), kit pengembangan AWS perangkat lunak (SDK), dan SDK SimSpace Weaver aplikasi.

Analisis konfigurasi dan kerentanan di AWS SimSpace Weaver

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara Anda AWS dan Anda. Untuk informasi lebih lanjut, lihat [model tanggung jawab AWS bersama model](#) . AWS menangani tugas keamanan dasar untuk infrastruktur yang mendasarinya, seperti menambal sistem operasi pada instance komputasi, konfigurasi firewall, dan pemulihan bencana AWS infrastruktur. Prosedur ini telah ditinjau dan disertifikasi oleh pihak ketiga yang sesuai. Untuk detail selengkapnya, lihat [Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan](#).

Anda bertanggung jawab atas keamanan perangkat lunak simulasi Anda:

- Pertahankan kode aplikasi Anda, termasuk pembaruan dan patch keamanan.
- Mengautentikasi dan mengenkripsi komunikasi antara klien simulasi Anda dan aplikasi yang mereka sambungkan.
- Perbarui simulasi Anda untuk menggunakan versi SDK terbaru, termasuk SDK dan AWS SimSpace Weaver SDK aplikasi.

Note

SimSpace Weaver tidak mendukung pembaruan ke aplikasi dalam simulasi yang sedang berjalan. Jika Anda perlu memperbarui aplikasi, Anda harus menghentikan dan menghapus simulasi, lalu membuat simulasi baru dengan kode aplikasi yang diperbarui. Kami menyarankan Anda menyimpan status simulasi di penyimpanan data eksternal sehingga Anda dapat mengembalikannya jika Anda perlu membuat ulang simulasi.

Praktik terbaik keamanan untuk SimSpace Weaver

Bagian ini menjelaskan praktik terbaik keamanan yang khusus untuk SimSpace Weaver. Untuk mempelajari lebih lanjut tentang praktik terbaik keamanan di AWS, lihat [Praktik Terbaik untuk Keamanan, Identitas, dan Kepatuhan](#).

Topik

- [Enkripsi komunikasi antara aplikasi Anda dan klien mereka](#)
- [Cadangkan status simulasi Anda secara berkala](#)
- [Pertahankan aplikasi Anda dan SDKs](#)

Enkripsi komunikasi antara aplikasi Anda dan klien mereka

SimSpace Weaver tidak mengelola komunikasi antara aplikasi Anda dan klien mereka. Anda harus menerapkan beberapa bentuk otentikasi dan enkripsi untuk sesi klien.

Cadangkan status simulasi Anda secara berkala

SimSpace Weaver tidak menyimpan status simulasi Anda. Simulasi yang dihentikan (sebagai akibat dari panggilan API, opsi konsol, atau kerusakan sistem) tidak menyimpan statusnya dan tidak memiliki cara yang melekat untuk memulihkannya. Simulasi yang dihentikan tidak dapat dimulai ulang. Satu-satunya cara untuk melakukan yang setara dengan restart adalah dengan membuat ulang simulasi Anda menggunakan konfigurasi dan data yang sama. Anda dapat menggunakan cadangan status simulasi Anda untuk menginisialisasi simulasi baru. AWS menawarkan [penyimpanan](#) cloud dan layanan [database](#) yang sangat andal dan tersedia yang dapat Anda gunakan untuk menyimpan status simulasi Anda.

Pertahankan aplikasi Anda dan SDKs

Pertahankan aplikasi Anda, instalasi lokal kit pengembangan AWS perangkat lunak (SDKs), dan SDK SimSpace Weaver aplikasi. Anda dapat mengunduh dan menginstal versi baru dari file AWS SDKs. Uji versi baru SDK SimSpace Weaver aplikasi dengan build aplikasi non-produksi untuk memastikan aplikasi Anda terus berjalan seperti yang diharapkan. Anda tidak dapat memperbarui aplikasi Anda dalam simulasi yang sedang berjalan. Untuk memperbarui aplikasi Anda:

1. Perbarui dan uji kode aplikasi secara lokal (atau di lingkungan pengujian).
2. Berhenti mengubah status simulasi Anda dan simpan (jika perlu).
3. Hentikan simulasi Anda (setelah berhenti, tidak dapat dimulai ulang).
4. Hapus simulasi Anda (simulasi berhenti yang tidak dihapus dihitung terhadap batas layanan Anda).
5. Buat ulang simulasi Anda dengan konfigurasi yang sama dan kode aplikasi yang diperbarui.
6. Inisialisasi simulasi Anda menggunakan data status tersimpan (jika tersedia).
7. Mulai simulasi baru Anda.

Note

Simulasi baru yang dibuat dengan konfigurasi yang sama terpisah dari simulasi lama. Ini akan memiliki ID simulasi baru dan mengirim log ke aliran log baru di Amazon CloudWatch.

Penebangan dan pemantauan di SimSpace Weaver

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja SimSpace Weaver dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton SimSpace Weaver, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses data log dari SimSpace Weaver pekerja CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam data log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#).
- AWS CloudTrail merekam panggilan API dan kejadian terkait yang dilakukan oleh atau atas Akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun yang memanggil AWS, alamat IP asal panggilan dilakukan, dan waktu panggilan terjadi. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudTrail](#).

Topik

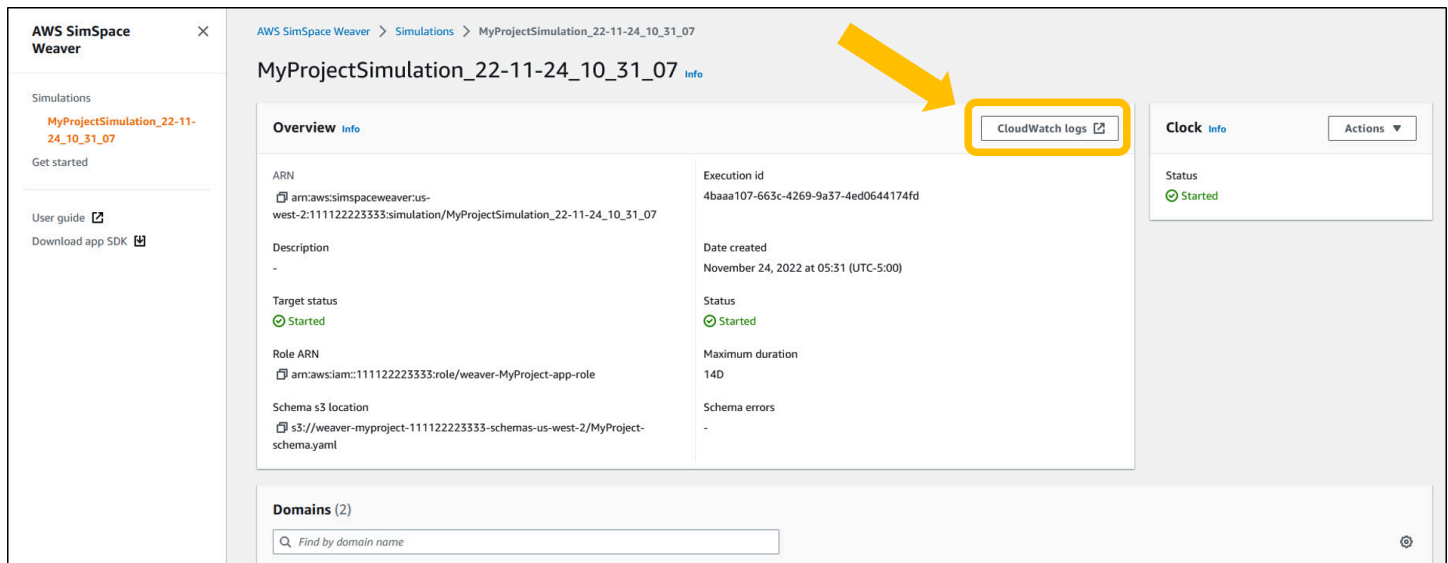
- [SimSpace Weaver log di Amazon CloudWatch Log](#)
- [Pemantauan SimSpace Weaver dengan Amazon CloudWatch](#)
- [Logging panggilan AWS SimSpace Weaver API menggunakan AWS CloudTrail](#)

SimSpace Weaver log di Amazon CloudWatch Log

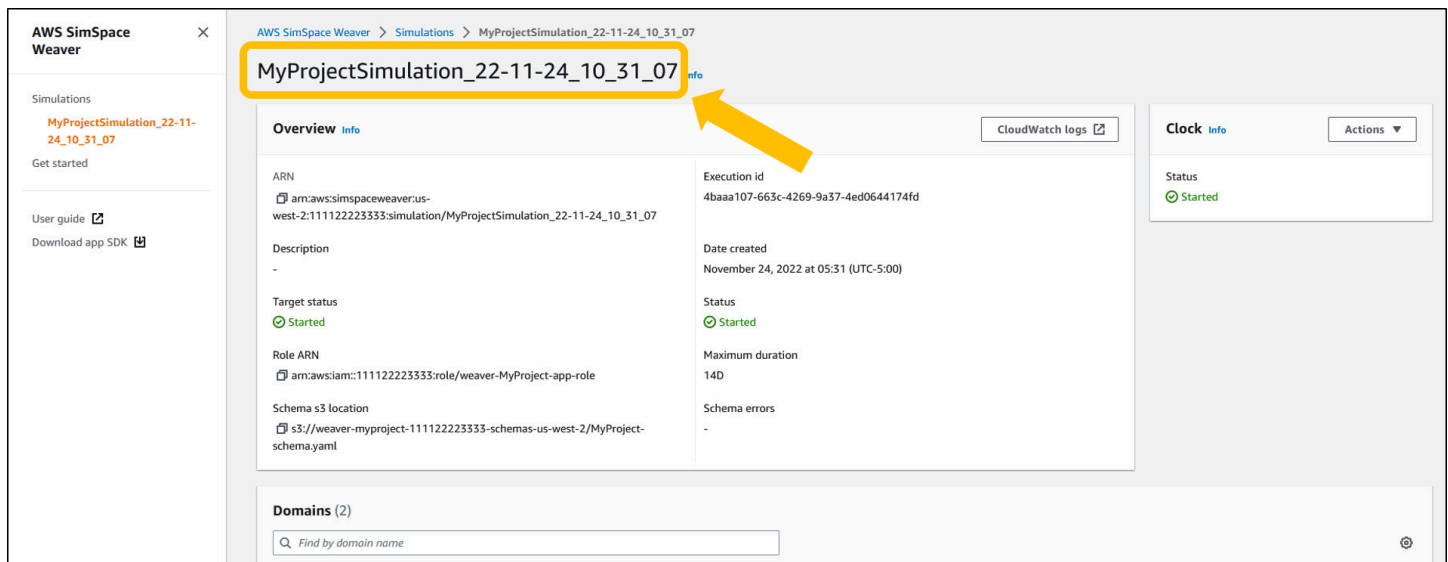
Mengakses log Anda SimSpace Weaver

Semua log yang dihasilkan dari SimSpace Weaver simulasi Anda disimpan di Amazon CloudWatch Logs. Untuk mengakses log Anda, Anda dapat menggunakan tombol CloudWatch log di panel

Ikhtisar simulasi Anda di SimSpace Weaver konsol, yang akan membawa Anda langsung ke log untuk simulasi tertentu.



Anda juga dapat mengakses log melalui CloudWatch konsol. Anda akan memerlukan nama simulasi Anda untuk mencari lognya.



SimSpace Weaver log

SimSpace Weaver menulis pesan manajemen simulasi dan output konsol dari aplikasi Anda ke Amazon CloudWatch Logs. Untuk informasi selengkapnya tentang bekerja dengan log, lihat [Bekerja dengan grup log dan aliran log](#) di Panduan Pengguna CloudWatch Log Amazon.

Setiap simulasi yang Anda buat memiliki grup lognya sendiri di CloudWatch Log. Nama grup log ditentukan dalam skema simulasi. Dalam cuplikan skema berikut, nilainya adalah.

`log_destination_service` logs ini berarti bahwa nilai `log_destination_resource_name` adalah nama grup log. Dalam hal ini, grup log adalah `MySimulationLogs`.

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

Anda juga dapat menggunakan `DescribeSimulation` API untuk menemukan nama grup log untuk simulasi setelah Anda memulainya.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Contoh berikut menunjukkan bagian dari output dari `DescribeSimulation` yang menjelaskan konfigurasi logging. Nama grup log ditampilkan di akhir `LogGroupArn`.

```
"LoggingConfiguration": {
  "Destinations": [
    {
      "CloudWatchLogsLogGroup": {
        "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
      }
    }
  ]
},
```

Setiap grup log simulasi berisi beberapa aliran log:

- Aliran log manajemen — pesan manajemen simulasi yang dihasilkan oleh SimSpace Weaver layanan.

```
/sim/management
```

- Aliran log kesalahan — pesan kesalahan yang dihasilkan oleh SimSpace Weaver layanan. Aliran log ini hanya ada jika ada kesalahan. SimSpace Weaver menyimpan kesalahan yang ditulis oleh aplikasi Anda di aliran log aplikasi mereka sendiri (lihat aliran log berikut).

```
/sim/errors
```

- Aliran log aplikasi spasial (1 untuk setiap aplikasi spasial pada setiap pekerja) — keluaran konsol yang dihasilkan oleh aplikasi spasial. Setiap aplikasi spasial menulis ke aliran lognya sendiri. *spatial-app-id* itu semua karakter setelah garis miring di akhir. *worker-id*

```
/domain/spatial-domain-name/app/worker-worker-id/spatial-app-id
```

- Aliran log aplikasi khusus (1 untuk setiap instance aplikasi kustom) — keluaran konsol yang dihasilkan oleh aplikasi khusus. Setiap instance aplikasi kustom menulis ke aliran lognya sendiri.

```
/domain/custom-domain-name/app/custom-app-name/random-id
```

- Aliran log aplikasi layanan (1 untuk setiap instance aplikasi layanan) — keluaran konsol yang dihasilkan oleh aplikasi layanan. Setiap aplikasi layanan menulis ke aliran lognya sendiri. *service-app-id* itu semua karakter setelah garis miring di akhir. *service-app-name*

```
/domain/service-domain-name/app/service-app-name/service-app-id
```

Pemantauan SimSpace Weaver dengan Amazon CloudWatch

Anda dapat memantau SimSpace Weaver menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

SimSpace Weaver Layanan melaporkan metrik berikut di `AWS/simspaceweaver` namespace.

SimSpace Weaver metrik di tingkat akun

SimSpace Weaver Namespace menyertakan metrik berikut yang terkait dengan aktivitas di tingkat akun. AWS

Metrik	Deskripsi
SimulationCount	Jumlah simulasi untuk akun saat ini. Unit: Hitungan Dimensi: tidak ada Statistik: Rata-rata, Minimum, Maksimum

Logging panggilan AWS SimSpace Weaver API menggunakan AWS CloudTrail

AWS SimSpace Weaver terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di SimSpace Weaver. CloudTrail menangkap semua panggilan API untuk SimSpace Weaver sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari SimSpace Weaver konsol dan panggilan kode ke operasi SimSpace Weaver API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk SimSpace Weaver. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol di Event history. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat SimSpace Weaver, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

SimSpace Weaver informasi di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di SimSpace Weaver, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya di Event history. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan Riwayat CloudTrail acara](#).

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk SimSpace Weaver, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail layanan dan integrasi yang didukung](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua SimSpace Weaver tindakan dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi AWS SimSpace Weaver API](#). Misalnya, panggilan ke `ListSimulations`, `DescribeSimulation` dan `DeleteSimulation` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan itu dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat [Elemen userIdentity CloudTrail](#).

Memahami entri file SimSpace Weaver log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber mana pun dan mencakup informasi tentang tindakan yang diminta, seperti tanggal dan waktu tindakan, parameter permintaan, dan detail lainnya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan ListSimulations tindakan.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:aws-console-signin-utils",
    "arn": "arn:aws:sts::111122223333:assumed-role/ConsoleSigninRole/aws-console-
signin-utils",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/ConsoleSigninRole",
        "accountId": "111122223333",
        "userName": "ConsoleSigninRole"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-02-14T15:57:02Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-02-14T15:57:08Z",
  "eventSource": "simspaceweaver.amazonaws.com",
  "eventName": "ListSimulations",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.10",
  "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/86.0.4240.0 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "1234abcd-1234-5678-abcd-12345abcd123",
  "eventID": "5678abcd-5678-1234-ab12-123abc123abc",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

```
}
```

SimSpace Titik akhir dan kuota Weaver

Tabel berikut menjelaskan titik akhir layanan dan kuota layanan untuk SimSpace Weaver. Kuota layanan, juga disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS. Untuk informasi selengkapnya, lihat [Service Quotas AWS](#) di Referensi Umum AWS .

Titik akhir layanan

Nama wilayah	Wilayah	Titik Akhir	Protokol
US East (N. Virginia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nama wilayah	Wilayah	Titik Akhir	Protokol
Europe (Frankfurt)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europe (Ireland)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (AS-Timur)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (AS-Barat)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Kuota layanan

Nama	Default	Dapat disesuai	Deskripsi
Menghitung unit sumber daya untuk setiap aplikasi	Setiap Wilayah yang didukung: 4	Tidak	Jumlah maksimum unit sumber daya komputasi yang dapat Anda alokasikan ke setiap aplikasi.
Menghitung unit sumber daya untuk setiap pekerja	Setiap Wilayah yang didukung: 17	Tidak	Jumlah unit sumber daya komputasi yang tersedia untuk setiap pekerja.
Bidang data untuk setiap entitas	Setiap Wilayah yang didukung: 7	Tidak	Jumlah maksimum bidang data (non-inde

Nama	Default	Dapat disesuaikan	Deskripsi
			ks) yang dapat dimiliki entitas.
Entitas dalam partisi	Setiap Wilayah yang didukung: 8,192	Tidak	Jumlah maksimum entitas dalam 1 partisi.
Ukuran bidang data entitas	Setiap Wilayah yang didukung: 1.024 Bytes	Tidak	Ukuran maksimum bidang data (non-indeks) suatu entitas.
Transfer entitas antar pekerja	Setiap Wilayah yang didukung: 25	Tidak	Jumlah maksimum transfer entitas antar pekerja, untuk setiap partisi dan setiap centang.
Transfer entitas pada pekerja yang sama	Setiap Wilayah yang didukung: 500	Tidak	Jumlah maksimum transfer entitas pada pekerja yang sama, untuk setiap partisi dan setiap centang.
Kolom indeks untuk setiap entitas	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum kolom indeks yang dapat dimiliki entitas.

Nama	Default	Dapat disesu an	Deskripsi
Durasi maksimum terbesar (dalam beberapa hari) untuk simulasi	Setiap Wilayah yang didukung: 14	Tidak	Jumlah hari terbesar yang dapat Anda tentukan sebagai durasi maksimum untuk simulasi. Semua simulasi memiliki durasi maksimum, bahkan jika Anda tidak menentukan nilainya. Simulasi secara otomatis berhenti ketika mencapai durasi maksimumnya.
Memori untuk setiap unit sumber daya komputasi	Setiap Wilayah yang didukung: 1 Gigabytes	Tidak	Jumlah memori akses acak (RAM) yang diperoleh aplikasi untuk setiap unit sumber daya komputasi.
Langganan jarak jauh untuk setiap pekerja	Setiap Wilayah yang didukung: 24	Tidak	Jumlah maksimum langganan jarak jauh untuk setiap pekerja.
Hitungan simulasi	Setiap Wilayah yang didukung: 2	Ya	Jumlah maksimum simulasi dengan status target STARTED di akun Anda. Anda dapat meminta kenaikan kuota hingga 10.

Nama	Default	Dapat disesuaikan	Deskripsi
Pekerja untuk simulasi	Setiap Wilayah yang didukung: 2	Ya	Jumlah maksimum pekerja yang dapat Anda tetapkan ke 1 simulasi. Anda dapat meminta kenaikan kuota hingga 10.
v CPUs untuk setiap unit sumber daya komputasi	Setiap Wilayah yang didukung: 2	Tidak	Jumlah unit pemrosesan pusat virtual (vCPUs) yang diperoleh aplikasi untuk setiap unit sumber daya komputasi.

Kuota perpesanan

Kuota berikut berlaku untuk pesan aplikasi ke aplikasi untuk SimSpace Weaver Local dan di. AWS Cloud

Nama	Default	Dapat disesuaikan	Deskripsi
Ukuran pesan maksimal	Setiap Wilayah yang didukung: 256 byte	Tidak	Ukuran maksimum payload pesan.
Tingkat pengiriman pesan maksimum	Setiap Wilayah yang didukung: 128	Tidak	Jumlah maksimum pesan yang dapat dikirim setiap aplikasi per centang.

Tarif jam

Skema simulasi menentukan clock rate (juga disebut tick rate) untuk simulasi. Tabel berikut menentukan tarif jam valid yang dapat Anda gunakan.

Nama	Nilai yang valid	Deskripsi
Tingkat jam	Setiap Wilayah yang didukung: "10", "15", "30", "tidak terbatas"	Tarif jam yang valid untuk simulasi.
Tingkat jam (versi 1.13 dan 1.12)	Setiap Wilayah yang didukung: 10, 15, 30	Tarif jam yang valid untuk simulasi.

Kuota layanan untuk SimSpace Weaver Local

Kuota layanan berikut SimSpace Weaver Local hanya berlaku untuk. Semua kuota lainnya juga berlaku untuk SimSpace Weaver Local.

Nama	Default	Dapat disesuaikan	Deskripsi
Partisi maksimum	SimSpace Weaver Local: 24	Tidak	Jumlah maksimum partisi untuk simulasi.
Aplikasi maksimum	SimSpace Weaver Local: 24	Tidak	Jumlah maksimum aplikasi (dari jenis apa pun) untuk simulasi.
Domain maksimum	SimSpace Weaver Local: 24	Tidak	Jumlah maksimum domain (dari jenis apa pun) untuk simulasi.
Entitas per partisi	SimSpace Weaver Local: 4.096	Tidak	Jumlah maksimum entitas di setiap partisi.
Bidang per entitas	SimSpace Weaver Local: 8	Tidak	Jumlah maksimum bidang untuk setiap entitas.
Ukuran bidang	SimSpace Weaver Local: 1024 byte	Tidak	Ukuran maksimum bidang entitas.

Pemecahan masalah di SimSpace Weaver

Topik

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [Izin ditolak selama simulasi dimulai](#)
- [Masalah yang berkaitan dengan waktu saat menggunakan Docker](#)
- [PathfindingSample klien konsol gagal terhubung](#)
- [Yang AWS CLI tidak mengenali simspaceweaver](#)

AssumeRoleAccessDenied

Anda mungkin menerima kesalahan berikut jika simulasi Anda gagal untuk memulai:

```
Unable to assume role arn:aws:iam::111122223333:role/weaver-project-name-app-role;
verify the role exists and has trust policy on SimSpace Weaver
```

Anda dapat menerima kesalahan ini jika salah satu dari berikut ini benar tentang peran AWS Identity and Access Management (IAM) untuk simulasi Anda:

- Nama Sumber Daya Amazon (ARN) mengacu pada peran IAM yang tidak ada.
- Kebijakan kepercayaan untuk peran IAM yang tidak memungkinkan nama simulasi baru untuk mengambil peran.

Periksa untuk memastikan bahwa peran itu ada. Jika peran itu ada, periksa kebijakan kepercayaan Anda untuk peran tersebut. Contoh kebijakan kepercayaan berikut hanya mengizinkan simulasi (di akun 111122223333) yang namanya dimulai dengan untuk mengambil peran. `aws:SourceArn MySimulation`

JSON

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "simspaceweaver.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*"
      }
    }
  }
]
}

```

Untuk memungkinkan simulasi lain yang namanya dimulai `MyOtherSimulation` untuk mengambil peran, kebijakan kepercayaan harus dimodifikasi seperti pada contoh yang diedit berikut:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "simspaceweaver.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*",
            "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MyOtherSimulation*"
          ]
        }
      }
    }
  ]
}

```

```
}  
  }  
}
```

InvalidBucketName

Anda mungkin menerima kesalahan berikut saat membuat proyek:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

Anda menerima kesalahan ini karena nama yang SimSpace Weaver diteruskan ke Amazon Simple Storage Service (Amazon S3) melanggar aturan penamaan bucket (untuk informasi selengkapnya, [lihat Aturan penamaan Bucket di Panduan Pengguna](#) Layanan Penyimpanan Sederhana Amazon).

create-projectScrip di SDK SimSpace Weaver aplikasi membuat nama bucket menggunakan nama project yang Anda berikan ke skrip. Nama bucket menggunakan format berikut:

- Versi 1.13.x atau yang lebih baru
 - `weaver-lowercase-project-name-account-number-region`
- Versi 1.12.x
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

Misalnya, mengingat properti proyek berikut:

- Nama proyek: MyProject
- Akun AWS nomor: 111122223333
- Wilayah AWS: us-west-2

Proyek ini akan memiliki ember berikut:

- Versi 1.13.x atau yang lebih baru

- `weaver-myproject-111122223333-us-west-2`
- Versi 1.12.x
 - `weaver-myproject-111122223333-app-zips-us-west-2`
 - `weaver-myproject-111122223333-schemas-us-west-2`

Nama proyek Anda tidak boleh melanggar aturan penamaan Amazon S3. Anda juga harus menggunakan nama proyek yang cukup pendek sehingga nama bucket yang dibuat oleh `create-project` skrip tidak melebihi batas panjang nama untuk bucket Amazon S3.

ServiceQuotaExceededException

Anda mungkin menerima kesalahan berikut saat memulai simulasi:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

Anda akan menerima kesalahan ini jika Anda mencoba memulai simulasi baru tetapi akun Anda saat ini memiliki jumlah simulasi maksimum dengan status target `STARTED`. Ini termasuk menjalankan simulasi, simulasi gagal, dan simulasi yang berhenti karena mereka mencapai durasi maksimumnya. Anda dapat menghapus simulasi berhenti atau gagal untuk memungkinkan Anda memulai simulasi baru. Jika semua simulasi Anda berjalan, Anda dapat menghentikan dan menghapus simulasi yang sedang berjalan. Anda juga dapat meminta peningkatan kuota layanan Anda jika Anda belum mencapai batas permintaan.

TooManyBuckets

Anda mungkin menerima kesalahan berikut saat membuat proyek:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) memiliki batasan jumlah bucket yang dapat Anda miliki di AWS akun (untuk informasi selengkapnya, [lihat Pembatasan dan batasan Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#)).

Anda harus melakukan salah satu hal berikut sebelum Anda dapat melanjutkan:

- Hapus 2 atau lebih bucket Amazon S3 yang sudah ada yang tidak Anda perlukan.
- Minta peningkatan batas Amazon S3 (untuk informasi selengkapnya, lihat [Pembatasan dan batasan Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon).
- Gunakan AWS akun yang berbeda.

Note

DeleteSimulationAPI di SimSpace Weaver tidak menghapus sumber daya Amazon S3 yang terkait dengan simulasi Anda. Kami menyarankan Anda menghapus semua sumber daya yang terkait dengan simulasi Anda saat Anda tidak lagi membutuhkannya.

Izin ditolak selama simulasi dimulai

Saat memulai simulasi, Anda mungkin mendapatkan pesan kesalahan yang menunjukkan bahwa izin ditolak atau ada kesalahan saat mengakses artefak aplikasi Anda. Masalah ini dapat terjadi saat Anda menentukan bucket Amazon S3 untuk simulasi yang SimSpace Weaver tidak dibuat untuk Anda (baik melalui konsol atau skrip SDK SimSpace Weaver aplikasi).

Situasi berikut adalah akar penyebab yang paling mungkin:

- Layanan ini tidak memiliki izin untuk mengakses satu atau beberapa bucket Amazon S3 yang Anda tentukan dalam skema simulasi — periksa kebijakan izin peran aplikasi, kebijakan bucket Amazon S3, dan izin bucket Amazon S3 untuk memastikan `simspaceweaver.amazonaws.com` bahwa memiliki izin yang benar untuk mengakses bucket Anda. Untuk informasi selengkapnya tentang kebijakan izin peran aplikasi, lihat [izin yang SimSpace Weaver dibuat untuk Anda](#).
- Bucket Amazon S3 Anda mungkin berbeda Wilayah AWS dari simulasi Anda — Bucket Amazon S3 Anda untuk artefak simulasi Anda harus sama dengan simulasi Anda. Wilayah AWS Periksa konsol Amazon S3 Anda untuk melihat apa yang ada di bucket Wilayah AWS Anda. Jika bucket Amazon S3 Anda berbeda Wilayah AWS, pilih bucket yang Wilayah AWS sama dengan simulasi Anda.

Masalah yang berkaitan dengan waktu saat menggunakan Docker

Jika Anda menggunakan Docker dan menerima kesalahan terkait waktu saat menjalankan skrip dari SDK SimSpace Weaver aplikasi, penyebabnya bisa jadi jam mesin Docker virtual Anda salah. Ini bisa terjadi jika komputer Anda berjalan Docker dan kemudian dilanjutkan dari tidur atau hibernasi.

Solusi untuk dicoba

- Mulai ulang Docker.
- Nonaktifkan dan kemudian aktifkan kembali sinkronisasi waktu di: Windows PowerShell

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample klien konsol gagal terhubung

Anda mungkin mendapatkan kesalahan berikut dari klien konsol saat Anda terhubung ke PathfindingSample simulasi yang dijelaskan dalam tutorial di [Memulai dengan SimSpace Weaver](#). Kesalahan ini terjadi karena klien tidak dapat membuka koneksi jaringan ke ViewApp alamat IP gabungan dan nomor port yang Anda berikan.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Untuk simulasi di AWS Cloud

- Apakah koneksi jaringan Anda berfungsi dengan benar? Verifikasi bahwa Anda dapat terhubung ke alamat IP lain atau situs web yang seharusnya berfungsi. Pastikan browser web Anda tidak memuat situs web dari cache-nya.
- Apakah simulasi Anda berjalan? Anda dapat menggunakan ListSimulationsAPI untuk mendapatkan status simulasi Anda. Untuk informasi selengkapnya, lihat [Dapatkan alamat IP dan nomor port aplikasi khusus](#). Anda juga dapat menggunakan [SimSpace Weaver konsol](#) untuk memeriksa status simulasi Anda.
- Apakah aplikasi Anda berjalan? Anda dapat menggunakan DescribeAppAPI untuk mendapatkan status aplikasi Anda. Untuk informasi selengkapnya, lihat [Dapatkan alamat IP dan nomor port aplikasi khusus](#). Anda juga dapat menggunakan [SimSpace Weaver konsol](#) untuk memeriksa status simulasi Anda.
- Apakah aplikasi Anda berjalan? Anda dapat menggunakan DescribeAppAPI untuk mendapatkan status aplikasi Anda. Untuk informasi selengkapnya, lihat [Dapatkan alamat IP dan nomor port aplikasi khusus](#). Anda juga dapat menggunakan [SimSpace Weaver konsol](#) untuk memeriksa status simulasi Anda.

- Apakah Anda menggunakan alamat IP dan nomor port yang benar? Ketika Anda terhubung melalui internet, Anda harus menggunakan alamat IP dan nomor Actual portViewApp. Anda dapat menemukan IP Address dan nomor Actual port di EndpointInfo blok output DescribeAppAPI. Anda juga dapat menggunakan [SimSpace Weaver konsol](#) untuk menemukan alamat IP (URI) dan nomor port (port Ingress) untuk Anda ViewApp di halaman MyViewDomain detail.
- Apakah koneksi jaringan Anda melalui firewall? Firewall Anda mungkin memblokir koneksi Anda ke alamat IP atau nomor port (atau keduanya). Periksa pengaturan firewall Anda atau periksa dengan administrator firewall Anda.

Untuk simulasi lokal

- Bisakah Anda terhubung ke alamat loopback Anda (127.0.0.1)? Jika Anda memiliki alat baris ping perintah di Windows, Anda dapat membuka jendela prompt perintah dan mencoba melakukan ping 127.0.0.1. Tekan Ctrl - C untuk mengakhiri ping.

```
ping 127.0.0.1
```

Example keluaran ping

```
C:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
Control-C
^C
C:\>
```

Jika ping mengatakan kehilangan paket, Anda mungkin memiliki perangkat lunak lain (seperti firewall lokal, pengaturan keamanan, atau program anti-malware) yang memblokir koneksi Anda.

- Apakah aplikasi Anda berjalan? Simulasi lokal Anda berjalan sebagai jendela terpisah untuk setiap aplikasi. Pastikan jendela untuk aplikasi spasial ViewApp Anda dan terbuka. Untuk informasi selengkapnya, lihat [Pembangunan lokal di SimSpace Weaver](#).
- Apakah Anda menggunakan alamat IP dan nomor port yang benar? Anda harus menggunakan `tcp://127.0.0.1:7000` ketika Anda terhubung ke simulasi lokal. Untuk informasi selengkapnya, lihat [Pembangunan lokal di SimSpace Weaver](#).
- Apakah Anda memiliki perangkat lunak keamanan lokal yang dapat memblokir koneksi Anda? Periksa pengaturan keamanan Anda, firewall lokal, atau program anti-malware untuk melihat apakah mereka memblokir koneksi Anda `127.0.0.1` pada port TCP. `7000`

Yang AWS CLI tidak mengenali **simspaceweaver**

Jika AWS CLI memberi Anda kesalahan yang menunjukkan bahwa itu tidak tahu tentang SimSpace Weaver, jalankan perintah berikut.

```
aws simspaceweaver help
```

Jika Anda mendapatkan kesalahan yang dimulai dengan baris berikut dan mencantumkan semua pilihan yang tersedia maka Anda AWS CLI mungkin versi yang lebih lama.

```
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
```

To see help text, you can run:

```
aws help
aws <command> help
aws <command> <subcommand> help
```

```
aws: error: argument command: Invalid choice, valid choices are:
```

Jalankan perintah berikut untuk memeriksa versi Anda AWS CLI.

```
aws --version
```

Jika nomor versi lebih awal dari 2.9.19 maka Anda harus memperbarui. AWS CLI Perhatikan bahwa versi saat ini lebih lambat dari 2.9.19. AWS CLI

Untuk memperbarui AWS CLI, lihat [Menginstal atau memperbarui versi terbaru dari AWS CLI](#)
Panduan AWS Command Line Interface Pengguna untuk Versi 2.

SimSpace Weaver referensi skema simulasi

SimSpace Weaver menggunakan file YAMG untuk mengkonfigurasi properti simulasi. File ini disebut skema simulasi (atau hanya skema). Simulasi sampel yang disertakan dalam SDK SimSpace Weaver aplikasi menyertakan skema yang dapat Anda salin dan edit untuk simulasi Anda sendiri.

Topik

- [Contoh skema lengkap](#)
- [Format skema](#)

Contoh skema lengkap

Contoh berikut menunjukkan YAML-file teks berformat yang menggambarkan simulasi. SimSpace Weaver Contoh ini mencakup nilai dummy untuk properti. Format file bervariasi berdasarkan nilai yang `sdk_version` ditentukan di dalamnya. Lihat [Format skema](#) deskripsi lengkap tentang properti dan nilai validnya.

```
sdk_version: "1.17"
simulation_properties:
  log_destination_resource_name: "MySimulationLogs"
  log_destination_service: "logs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 3
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [-1000, 1000]
      y: [-1000, 1000]
    grid_placement_groups:
      x: 3
      y: 3
```

```
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports: [9000, 9001]
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/
MyConnectionServiceApp.zip"
      launch_command: ["MyConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
  MySpatialDomainWithCustomContainer:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp2.zip"
      launch_command: ["MySpatialApp2"]
      required_resource_units:
```

```
    compute: 1
    image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest"
  placement_constraints:
    - placed_together: ["MySpatialDomain", "MySpatialDomainWithCustomContainer"]
    on_workers: ["MyComputeWorkers"]
```

Format skema

Contoh berikut menunjukkan struktur keseluruhan skema. Urutan properti di setiap tingkat skema tidak masalah, selama hubungan orang tua-anak sama. Urutan penting untuk elemen dalam array.

```
sdk_version: "sdk-version-number"
simulation_properties:
  simulation-properties
workers:
  worker-group-configurations
clock:
  tick_rate: tick-rate
partitioning_strategies:
  partitioning-strategy-configurations
domains:
  domain-configurations
placement_constraints:
  placement-constraints-configuration
```

Bagian-bagian

- [Versi SDK](#)
- [Sifat simulasi](#)
- [Pekerja](#)
- [Jam](#)
- [Strategi partisi](#)
- [Domain](#)
- [Kendala penempatan](#)

Versi SDK

`sdk_version` Bagian (wajib) mengidentifikasi versi SDK SimSpace Weaver aplikasi yang mendukung skema ini. Nilai valid: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

⚠ Important

Nilai `sdk_version` hanya mencakup nomor versi utama dan nomor versi minor pertama. Misalnya, nilai `1.12` menentukan semua versi `1.12.x`, seperti, `1.12.0`, `1.12.1`, dan `1.12.2`.

```
sdk_version: "1.17"
```

Sifat simulasi

`simulation_properties` Bagian (wajib) menentukan berbagai properti simulasi Anda. Gunakan bagian ini untuk mengonfigurasi logging dan menentukan gambar kontainer default. Bagian ini diperlukan bahkan jika Anda tidak mengonfigurasi logging atau memilih untuk menentukan gambar kontainer default.

```
simulation_properties:  
  log_destination_resource_name: "log-destination-resource-name"  
  log_destination_service: "log-destination-service"  
  default_entity_index_key_type: "Vector3<f32>"  
  default_image: "ecr-repository-uri"
```

Sifat-sifat

`log_destination_resource_name`

Menentukan sumber daya yang SimSpace Weaver akan menulis log ke.

Diperlukan: Tidak. Jika properti ini tidak disertakan, tidak SimSpace Weaver akan menulis log untuk simulasi.

Tipe: String

Nilai yang valid:

- Nama grup CloudWatch log Log (misalnya, `MySimulationLogs`)
- Nama Sumber Daya Amazon (ARN) dari grup CloudWatch log Log (misalnya,) `arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`

Note

SimSpace Weaver hanya mendukung tujuan log di akun yang sama dan Wilayah AWS sebagai simulasi.

log_destination_service

Menunjukkan jenis sumber daya tujuan pencatatan saat Anda menentukan `logging_destination_resource_name` yang bukan ARN.

Wajib: Anda harus menentukan properti ini jika ditentukan dan bukan ARN.

`log_destination_resource_name` Anda tidak dapat menentukan properti ini jika `log_destination_resource_name` tidak ditentukan atau ARN.

Tipe: String

Nilai yang valid:

- `logs`: Sumber daya tujuan log adalah grup log.

default_entity_index_key_type

Menentukan tipe data untuk bidang kunci indeks entitas simulasi.

Wajib: Ya

Tipe: String

Nilai yang valid: `Vector3<f32>`

default_image

Menentukan gambar kontainer default untuk simulasi Anda (tidak didukung untuk versi 1.13 dan 1.12). Jika properti ini ditentukan, domain yang tidak menentukan `image` menggunakan `default_image`

Wajib: Tidak

Tipe: String

Nilai yang valid:

- URI dari repositori di Amazon Elastic Container Registry (Amazon ECR) (misalnya, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Pekerja

`workers` Bagian (wajib) menentukan konfigurasi untuk kelompok pekerja (kelompok pekerja). SimSpace Weaver menggunakan informasi ini bersama-sama dengan `placement_constraints` untuk mengkonfigurasi infrastruktur dasar simulasi Anda. Hanya 1 grup pekerja yang saat ini didukung.

Untuk menentukan properti untuk grup pekerja, ganti *worker-group-name* dengan nama pilihan Anda. Nama harus 3-64 karakter dan dapat berisi karakter A - Z, a - z, 0 - 9, dan `_` (tanda hubung). Tentukan properti grup pekerja setelah nama.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Sifat-sifat

type

Menentukan jenis pekerja.

Wajib: Ya

Tipe: String

Nilai yang valid: `sim.c5.24xlarge`

desired

Menentukan jumlah yang diinginkan pekerja untuk kelompok pekerja ini.

Wajib: Ya

Tipe: Integer

Nilai yang valid: 1 -3. Kuota layanan Anda (limit) untuk jumlah pekerja untuk simulasi Anda menentukan nilai maksimum properti ini. Misalnya, jika kuota layanan Anda adalah 2 maka nilai maksimum untuk properti ini adalah 2. Anda dapat meminta peningkatan kuota layanan Anda. Untuk informasi selengkapnya, lihat [SimSpace Titik akhir dan kuota Weaver](#).

Jam

`clock`Bagian (wajib) menentukan properti jam simulasi.

```
clock:  
  tick_rate: tick-rate
```

Sifat-sifat

`tick_rate`

Menentukan jumlah kutu per detik yang dipublikasikan jam ke aplikasi.

Wajib: Ya

Jenis:

- Versi *1.14* dan *1.15*: String
- Versi *1.13* dan *1.12*: Integer

Nilai yang valid:

- Versi *1.14* dan *1.15*: "10" | "15" | "30" | "unlimited"
 - "unlimited": jam mengirimkan centang berikutnya segera setelah semua aplikasi menyelesaikan operasi komit mereka untuk centang saat ini.
- Versi *1.13* dan *1.12*: 10 | 15 | 30

Strategi partisi

`partitioning_strategies`Bagian (wajib) menentukan organisasi partisi untuk domain spasial.

Note

SimSpace Weaver hanya mendukung 1 strategi partisi.

Untuk menentukan properti untuk strategi partisi, ganti *partitioning-strategy-name* dengan nama pilihan Anda. Nama harus 3-64 karakter dan dapat berisi karakter A - Z, a - z, 0 - 9, dan _ - (tanda hubung). Tentukan properti strategi partisi setelah nama.

```
partitioning_strategies:  
  partitioning-strategy-name:  
    topology: "Grid"  
    aabb_bounds:  
      x: [aabb-min-x, aabb-max-x]  
      y: [aabb-min-y, aabb-max-y]  
    grid_placement_groups:  
      x: number-of-placement-groups-along-x-axis  
      y: number-of-placement-groups-along-y-axis
```

Sifat-sifat

topology

Menentukan topologi (skema pengaturan partisi) untuk strategi partisi ini.

Wajib: Ya

Tipe: String

Nilai yang valid: "Grid"

aabb_bounds

Menentukan batas-batas kotak pembatas sejajar sumbu utama (AABB) untuk simulasi Anda. Anda menentukan batas sebagai array berurutan 2 elemen yang menggambarkan nilai minimum dan maksimum (dalam urutan itu) untuk setiap sumbu (x and y).

Diperlukan: Bersyarat. Properti ini diperlukan (dan hanya dapat ditentukan) jika topologi diatur ke "Grid"

Jenis: Float array (untuk setiap sumbu)

Nilai yang valid: -3.4028235e38 - 3.4028235e38

grid_placement_groups

Menentukan jumlah kelompok penempatan sepanjang setiap sumbu (x dan y) dalam topologi grid. Grup penempatan adalah kumpulan partisi (dalam domain yang sama) yang berdekatan secara spasial.

Diperlukan: Bersyarat. Properti ini diperlukan (dan hanya dapat ditentukan) jika topologi diatur ke "Grid" Jika Anda tidak menentukan konfigurasi grup penempatan, SimSpace Weaver akan

menghitung satu untuk Anda. Setiap domain yang menggunakan strategi partisi tanpa konfigurasi grup penempatan harus menentukan `grid_partition` (lihat [Strategi partisi domain spasial](#)).

Jenis: Integer (untuk setiap sumbu)

Nilai yang valid: 1 -20. Kami merekomendasikan bahwa $x * y$ sama dengan jumlah pekerja yang diinginkan. Jika tidak, SimSpace Weaver akan mencoba menyeimbangkan grup penempatan Anda di seluruh pekerja yang tersedia.

Domain

`domains` Bagian (wajib) menentukan properti untuk setiap domain Anda. Semua simulasi harus memiliki setidaknya satu bagian untuk domain spasial. Anda dapat membuat beberapa bagian untuk domain tambahan. Setiap jenis domain memiliki format konfigurasinya sendiri.

Important

Versi 1.13 dan 1.12 tidak mendukung beberapa domain spasial.

Important

SimSpace Weaver mendukung hingga 5 domain untuk setiap simulasi. Ini mencakup semua domain spasial, kustom, dan layanan.

```
domains:  
  domain-name:  
    domain-configuration  
  domain-name:  
    domain-configuration  
  ...
```

Konfigurasi domain

- [Konfigurasi domain spasial](#)
- [Konfigurasi domain kustom](#)
- [Konfigurasi domain layanan](#)

Konfigurasi domain spasial

Untuk menentukan properti untuk domain spasial, ganti *spatial-domain-name* dengan nama pilihan Anda. Nama harus 3-64 karakter dan dapat berisi karakter A - Z, a - z, 0 - 9, dan _ (tanda hubung). Tentukan properti domain spasial setelah nama.

```
spatial-domain-name:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "partitioning-strategy-name"
    grid_partition:
      x: number-of-partitions-along-x-axis
      y: number-of-partitions-along-y-axis
    app_config:
      package: "app-package-s3-uri"
      launch_command: ["app-launch-command", "parameter1", ...]
      required_resource_units:
        compute: app-resource-units
      image: "ecr-repository-uri"
```

Strategi partisi domain spasial

`launch_apps_by_partitioning_strategy` Bagian (wajib) menentukan strategi partisi dan dimensi (dalam jumlah partisi) dari ruang simulasi.

```
launch_apps_by_partitioning_strategy:
  partitioning_strategy: "partitioning-strategy-name"
  grid_partition:
    x: number-of-partitions-along-x-axis
    y: number-of-partitions-along-y-axis
```

Sifat-sifat

`partitioning_strategy`

Menentukan strategi partisi untuk domain spasial ini.

Wajib: Ya

Tipe: String

Nilai yang valid: Nilai properti ini harus sesuai dengan nama strategi partisi yang ditentukan di bagian `partitioning_strategies`. Untuk informasi selengkapnya, lihat [Strategi partisi](#).

grid_partition

Menentukan jumlah partisi sepanjang setiap sumbu (x dan y) dalam topologi grid. Dimensi ini menggambarkan ruang simulasi total untuk domain ini.

Diperlukan: Bersyarat. Properti ini hanya dapat ditentukan jika topologi diatur ke "Grid" Properti ini bergantung pada `grid_placement_groups` properti strategi partisi yang ditentukan untuk domain ini:

- Properti ini diperlukan jika strategi partisi domain ini tidak menentukan konfigurasi `grid_placement_groups`
- Jika ada `grid_placement_groups` konfigurasi tetapi Anda tidak menentukan `grid_partition`, maka SimSpace Weaver akan menggunakan dimensi yang sama dengan `grid_placement_groups` konfigurasi.
- Jika Anda menentukan keduanya `grid_placement_groups` dan `grid_partition`, dimensi `grid_partition` harus kelipatan dari dimensi `grid_placement_groups` (misalnya, jika dimensi Anda 2x2, maka beberapa `grid_placement_groups` dimensi yang valid untuk adalah 2x2, 4x4, 6x6, 8x8, 10x10). `grid_partition`

Jenis: Integer (untuk setiap sumbu)

Nilai yang valid: 1 - 20

Konfigurasi aplikasi spasial

`app_config` Bagian (wajib) menentukan paket, konfigurasi peluncuran, dan persyaratan sumber daya untuk aplikasi di domain ini.

```
app_config:
  package: "app-package-s3-uri"
  launch_command: ["app-launch-command", "parameter1", ...]
  required_resource_units:
    compute: app-resource-units
```

Sifat-sifat

package

Menentukan paket (file zip) yang berisi aplikasi executable/binary. Paket harus disimpan dalam ember Amazon S3. Hanya format file zip yang didukung.

Wajib: Ya

Tipe: String

Nilai yang valid: URI Amazon S3 dari paket dalam bucket Amazon S3. Misalnya, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

`launch_command`

Menentukan nama file executable/binary dan parameter baris perintah untuk meluncurkan aplikasi. Setiap token string baris perintah adalah elemen dalam array.

Wajib: Ya

Jenis: Array string

`required_resource_units`

Menentukan jumlah unit sumber daya yang SimSpace Weaver harus dialokasikan untuk setiap instance aplikasi ini. Unit sumber daya adalah jumlah tetap unit pemrosesan pusat virtual (vCPUs) dan memori akses acak (RAM) pada seorang pekerja. Untuk informasi selengkapnya tentang unit sumber daya, lihat [Titik akhir dan kuota layanan](#). `computeProperty` menentukan alokasi unit sumber daya untuk `compute` keluarga pekerja, dan saat ini merupakan satu-satunya jenis alokasi yang valid.

Wajib: Ya

Tipe: Integer

Nilai yang valid: 1 - 4

Gambar kontainer kustom

`imageProperty` (opsional) menentukan lokasi gambar kontainer yang SimSpace Weaver digunakan untuk menjalankan aplikasi di domain ini (tidak didukung dalam versi 1.13 dan 1.12). Berikan URI ke repositori di Amazon Elastic Container Registry (Amazon ECR) yang berisi gambar. Jika properti ini tidak ditentukan tetapi `default_image` ditentukan di `simulation_properties` bagian tingkat atas, aplikasi di domain ini menggunakan `default_image`. Untuk informasi selengkapnya, lihat [Wadah khusus](#).

```
image: "ecr-repository-uri"
```

Sifat-sifat

image

Menentukan lokasi gambar kontainer untuk menjalankan aplikasi di domain ini.

Wajib: Tidak

Tipe: String

Nilai yang valid:

- URI dari repositori di Amazon Elastic Container Registry (Amazon ECR) (misalnya, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Konfigurasi domain kustom

Untuk menentukan properti untuk domain kustom, ganti *custom-domain-name* dengan nama pilihan Anda. Nama harus 3-64 karakter dan dapat berisi karakter A - Z, a - z, 0 - 9, dan _ - (tanda hubung). Tentukan properti domain kustom setelah nama. Ulangi proses ini untuk setiap domain kustom.

```
custom-domain-name:
  launch_apps_via_start_app_call: {}
  app_config:
    package: "app-package-s3-uri"
    launch_command: ["app-launch-command", "parameter1", ...]
    required_resource_units:
      compute: app-resource-units
    endpoint_config:
      ingress_ports: [port1, port2, ...]
  image: "ecr-repository-uri"
```

Sifat-sifat

launch_apps_via_start_app_call

Properti ini diperlukan untuk meluncurkan aplikasi kustom Anda menggunakan StartApp API.

Wajib: Ya

Tipe: N/A

Nilai yang valid: {}

Konfigurasi aplikasi khusus

`app_config` section(wajib) menentukan paket, konfigurasi peluncuran, persyaratan sumber daya, dan port jaringan untuk aplikasi di domain kustom ini.

```
app_config:
  package: "app-package-s3-uri"
  launch_command: ["app-launch-command", "parameter1", ...]
  required_resource_units:
    compute: app-resource-units
  endpoint_config:
    ingress_ports: [port1, port2, ...]
```

Sifat-sifat

package

Menentukan paket (file zip) yang berisi aplikasi executable/binary. Paket harus disimpan dalam ember Amazon S3. Hanya format file zip yang didukung.

Wajib: Ya

Tipe: String

Nilai yang valid: URI Amazon S3 dari paket dalam bucket Amazon S3. Misalnya, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

launch_command

Menentukan nama file executable/binary dan parameter baris perintah untuk meluncurkan aplikasi. Setiap token string baris perintah adalah elemen dalam array.

Wajib: Ya

Jenis: Array string

required_resource_units

Menentukan jumlah unit sumber daya yang SimSpace Weaver harus dialokasikan untuk setiap instance aplikasi ini. Unit sumber daya adalah jumlah tetap unit pemrosesan pusat virtual (vCPUs) dan memori akses acak (RAM) pada seorang pekerja. Untuk informasi selengkapnya tentang

unit sumber daya, lihat [Titik akhir dan kuota layanan](#). `computeProperti` menentukan alokasi unit sumber daya untuk `compute` keluarga pekerja, dan saat ini merupakan satu-satunya jenis alokasi yang valid.

Wajib: Ya

Tipe: Integer

Nilai yang valid: 1 - 4

`endpoint_config`

Menentukan titik akhir jaringan untuk aplikasi di domain ini. Nilai `ingress_ports` menentukan port yang diikat aplikasi kustom Anda untuk koneksi klien yang masuk. SimSpace Weaver memetakan port yang dialokasikan secara dinamis ke port `ingress` yang Anda tentukan. Port `Ingress` adalah TCP dan UDP. Gunakan `DescribeApp` API untuk menemukan nomor port aktual untuk menghubungkan klien Anda.

Diperlukan: Tidak. Jika Anda tidak menentukan konfigurasi titik akhir maka aplikasi kustom Anda di domain ini tidak akan memiliki titik akhir jaringan.

Jenis: Array integer

Nilai yang valid: 1024 -49152. Nilai harus unik.

Gambar kontainer kustom

`imageProperti` (opsional) menentukan lokasi gambar kontainer yang SimSpace Weaver digunakan untuk menjalankan aplikasi di domain ini (tidak didukung dalam versi 1.13 dan 1.12). Berikan URI ke repositori di Amazon Elastic Container Registry (Amazon ECR) yang berisi gambar. Jika properti ini tidak ditentukan tetapi `default_image` ditentukan di `simulation_properties` bagian tingkat atas, aplikasi di domain ini menggunakan `default_image`. Untuk informasi selengkapnya, lihat [Wadah khusus](#).

```
image: "ecr-repository-uri"
```

Sifat-sifat

`image`

Menentukan lokasi gambar kontainer untuk menjalankan aplikasi di domain ini.

Wajib: Tidak

Tipe: String

Nilai yang valid:

- URI dari repositori di Amazon Elastic Container Registry (Amazon ECR) (misalnya, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Konfigurasi domain layanan

Untuk menentukan properti untuk domain layanan, ganti *service-domain-name* dengan nama pilihan Anda. Nama harus 3-64 karakter dan dapat berisi karakter A - Z, a - z, 0 - 9, dan `_` (tanda hubung). Tentukan properti domain layanan setelah nama. Ulangi proses ini untuk setiap domain layanan.

```
service-domain-name:
  launch_apps_per_worker:
    count: number-of-apps-to-launch
  app_config:
    package: "app-package-s3-uri"
    launch_command: [app-launch-command", "parameter1", ...]
    required_resource_units:
      compute: app-resource-units
    endpoint_config:
      ingress_ports: [port1, port2, ...]
  image: "ecr-repository-uri"
```

Luncurkan aplikasi per pekerja

`launch_apps_per_worker` Bagian (wajib) menunjukkan bahwa ini adalah konfigurasi domain layanan, dan menentukan jumlah aplikasi layanan yang akan diluncurkan per pekerja.

```
launch_apps_per_worker:
  count: number-of-apps-to-launch
```

Sifat-sifat

count

Properti ini menentukan jumlah aplikasi layanan yang akan diluncurkan per pekerja.

Wajib: Ya

Tipe: Integer

Nilai yang valid: {} | 1 | 2. Nilai {} menentukan nilai default dari. 1

Konfigurasi aplikasi layanan

`app_config` `section`(wajib) menentukan paket, konfigurasi peluncuran, persyaratan sumber daya, dan port jaringan untuk aplikasi dalam domain layanan ini.

```
app_config:
  package: "app-package-s3-uri"
  launch_command: ["app-launch-command", "parameter1", ...]
  required_resource_units:
    compute: app-resource-units
  endpoint_config:
    ingress_ports: [port1, port2, ...]
```

Sifat-sifat

package

Menentukan paket (file zip) yang berisi aplikasi executable/binary. Paket harus disimpan dalam ember Amazon S3. Hanya format file zip yang didukung.

Wajib: Ya

Tipe: String

Nilai yang valid: URI Amazon S3 dari paket dalam bucket Amazon S3. Misalnya, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

launch_command

Menentukan nama file executable/binary dan parameter baris perintah untuk meluncurkan aplikasi. Setiap token string baris perintah adalah elemen dalam array.

Wajib: Ya

Jenis: Array string

required_resource_units

Menentukan jumlah unit sumber daya yang SimSpace Weaver harus dialokasikan untuk setiap instance aplikasi ini. Unit sumber daya adalah jumlah tetap unit pemrosesan pusat virtual (vCPUs) dan memori akses acak (RAM) pada seorang pekerja. Untuk informasi selengkapnya tentang unit sumber daya, lihat [Titik akhir dan kuota layanan](#). computeProperti menentukan alokasi unit sumber daya untuk compute keluarga pekerja, dan saat ini merupakan satu-satunya jenis alokasi yang valid.

Wajib: Ya

Tipe: Integer

Nilai yang valid: 1 - 4

endpoint_config

Menentukan titik akhir jaringan untuk aplikasi di domain ini. Nilai `ingress_ports` menentukan port yang diikat aplikasi layanan Anda untuk koneksi klien yang masuk. SimSpace Weaver memetakan port yang dialokasikan secara dinamis ke port ingress yang Anda tentukan. Port Ingress adalah TCP dan UDP. Gunakan DescribeApp API untuk menemukan nomor port aktual untuk menghubungkan klien Anda.

Diperlukan: Tidak. Jika Anda tidak menentukan konfigurasi titik akhir maka aplikasi layanan Anda di domain ini tidak akan memiliki titik akhir jaringan.

Jenis: Array integer

Nilai yang valid: 1024 -49152. Nilai harus unik.

Gambar kontainer kustom

`imageProperti` (opsional) menentukan lokasi gambar kontainer yang SimSpace Weaver digunakan untuk menjalankan aplikasi di domain ini (tidak didukung dalam versi 1.13 dan 1.12). Berikan URI ke repositori di Amazon Elastic Container Registry (Amazon ECR) yang berisi gambar. Jika properti ini tidak ditentukan tetapi `default_image` ditentukan di `simulation_properties` bagian tingkat atas, aplikasi di domain ini menggunakan `default_image`. Untuk informasi selengkapnya, lihat [Wadah khusus](#).

```
image: "ecr-repository-uri"
```

Sifat-sifat

image

Menentukan lokasi gambar kontainer untuk menjalankan aplikasi di domain ini.

Wajib: Tidak

Tipe: String

Nilai yang valid:

- URI dari repositori di Amazon Elastic Container Registry (Amazon ECR) (misalnya, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Kendala penempatan

`placement_constraints` Bagian (opsional) menentukan domain spasial mana yang SimSpace Weaver harus ditempatkan bersama pada pekerja yang sama. Untuk informasi selengkapnya, lihat [Mengkonfigurasi domain spasial](#).

Important

Versi 1.13 dan 1.12 tidak mendukung `placement_constraints`.

```
placement_constraints:  
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]  
    on_workers: ["worker-group-name"]
```

Sifat-sifat

placed_together

Menentukan domain spasial yang SimSpace Weaver harus menempatkan bersama-sama.

Wajib: Ya

Jenis: Array string

Nilai yang valid: Nama domain spasial yang ditentukan dalam skema

`on_workers`

Menentukan kelompok pekerja yang SimSpace Weaver harus menempatkan domain pada.

Wajib: Ya

Jenis: Array string 1-elemen

Nilai yang valid: Nama kelompok pekerja yang ditentukan dalam skema

SimSpace Weaver Referensi API

SimSpace Weaver memiliki 2 set antarmuka pemrograman aplikasi yang berbeda (APIs):

- Layanan APIs — Ini APIs mengontrol sumber daya layanan dan layanan, seperti simulasi, jam, dan aplikasi Anda. Mereka adalah bagian dari kit pengembangan AWS perangkat lunak utama (SDK) dan Anda dapat menggunakan AWS Command Line Interface (CLI) untuk memanggil mereka. Untuk informasi selengkapnya tentang service APIs, lihat referensi [SimSpace Weaver API](#).
- App SDK APIs — Ini APIs mengontrol data dalam simulasi Anda. Anda menggunakannya dalam kode aplikasi untuk melakukan hal-hal seperti membaca dan menulis data bidang entitas, bekerja dengan langganan, dan memantau peristiwa dalam simulasi. Untuk informasi selengkapnya, lihat dokumentasi SDK SimSpace Weaver aplikasi di folder SDK aplikasi Anda: `sdk-folder\SimSpaceWeaverAppSdk\documentation`

Note

`sdk-folder` adalah folder tempat Anda membuka ritseting paket.
SimSpaceWeaverAppSdkDistributable

AWS SimSpace Weaver versi

Kami terus meningkatkan AWS SimSpace Weaver. Anda harus mengunduh SDK SimSpace Weaver aplikasi terbaru saat kami merilis versi baru jika Anda ingin memanfaatkan fitur dan pembaruan fitur baru. Untuk menjalankan simulasi yang ada dengan versi yang lebih baru, Anda mungkin harus memperbarui skema dan kodenya, lalu memulai contoh simulasi baru. Anda tidak perlu memutakhirkan, dan dapat terus menjalankan simulasi yang ada dengan versi sebelumnya. Anda dapat memeriksa halaman ini untuk melihat apa yang berbeda di antara versi. Semua versi saat ini didukung.

Important

Versi terbaru dari [Panduan AWS SimSpace Weaver Pengguna](#) hanya mencakup versi terbaru dari layanan. Anda dapat menemukan dokumentasi untuk versi sebelumnya di [Katalog AWS SimSpace Weaver Panduan](#), tersedia dari [halaman arahan dokumentasi utama](#).

Versi terbaru

Versi terbaru adalah: 1.17.0

Cara menemukan versi Anda saat ini

Jika Anda membuat simulasi dengan SDK SimSpace Weaver aplikasi, `create-project` skrip akan mengunduh versi pustaka SDK ke dalam subdirectory di file Anda. *sdk-folder* Subdirektori yang berisi pustaka SDK memiliki nama yang menyertakan nomor versi SDK: `SimSpaceWeaverAppSdk-sdk-version` Misalnya, pustaka untuk versi 1.16.0 ada di `SimSpaceWeaverAppSdk-1.16.0`

Anda juga dapat menemukan versi paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan di file teks `app_sdk_distributable_version.txt` di file Anda. *sdk-folder*

Unduh versi terbaru

Gunakan salah satu tautan berikut untuk mengunduh versi terbaru.

- [Paket terdistribusi SDK aplikasi lengkap](#)
- [Hanya pustaka SDK aplikasi](#)

Anda juga dapat mengunduh paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan lengkap dari [SimSpace Weaver konsol](#) di Konsol Manajemen AWS Pilih Unduh SDK aplikasi dari panel navigasi.

Warning

Jangan gunakan AWS CLI untuk mengunduh apa pun yang tampaknya merupakan paket SimSpace Weaver aplikasi yang dapat didistribusikan SDK. Hanya gunakan tautan unduhan di halaman ini atau tautan unduhan di konsol. Metode atau lokasi unduhan lainnya tidak didukung dan mungkin berisi kode usang, salah, atau berbahaya.

Memecahkan masalah unduhan SDK aplikasi

Kami menggunakan Amazon CloudFront (CloudFront) untuk mendistribusikan file SDK. Zip aplikasi. Anda mungkin mengalami beberapa situasi berikut.

- Paket yang diunduh bukan versi terbaru
 - Jika file.zip yang Anda unduh tidak berisi versi terbaru, ada kemungkinan cache di lokasi CloudFront tepi Anda belum diperbarui. Unduh lagi setelah 24 jam.
- Anda mendapatkan kesalahan HTTP 4xx atau 5xx menggunakan tautan unduhan
 - Coba lagi setelah 24 jam. Jika Anda mendapatkan kesalahan yang sama, gunakan tautan Umpan Balik di bagian bawah [SimSpace Weaver konsol](#) untuk melaporkan masalah. Pilih Laporkan masalah sebagai Jenis umpan balik.
- Browser Anda melaporkan bahwa itu tidak dapat memuat halaman
 - Anda mungkin memiliki masalah konfigurasi jaringan atau browser lokal. Verifikasi bahwa Anda dapat memuat halaman lain. Bersihkan cache browser Anda dan coba lagi. Pastikan Anda tidak memiliki aturan firewall yang mungkin memblokir URL unduhan.
- Anda mendapatkan kesalahan ketika Anda mencoba untuk menyimpan file
 - Periksa izin sistem file lokal Anda untuk memastikan bahwa Anda memiliki izin yang benar untuk menyimpan file.
- Browser Anda ditampilkan AccessDenied

- Jika Anda memasukkan URL secara manual ke browser Anda, periksa apakah itu benar. Jika Anda menggunakan tautan unduhan, pastikan ada sesuatu yang tidak mengganggu URL di browser Anda; gunakan tautan lagi.

Instal versi terbaru

Pasang versi terbaru

1. [Unduh versi terbaru.](#)
2. Buka SimSpaceWeaverAppSdkDistributable zip.zip ke folder.
3. Jalankan `python setup.py` dari folder SDK SimSpace Weaver aplikasi versi terbaru yang tidak di-zip.
4. Gunakan folder SDK SimSpace Weaver aplikasi versi terbaru yang tidak di-zip, bukan versi sebelumnya.

Versi layanan

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.17.0	<p>Perubahan besar pada paket SimSpace Weaver aplikasi yang dapat didistribusikan SDK</p> <ul style="list-style-type: none">• Kami mengganti skrip Windows batch dan Linux Bash dengan skrip berbasis Python. Oleh karena itu Python 3.9 sekarang	April 17, 2024	Panduan ini	<ul style="list-style-type: none">• Paket lengkap• Perpustakaan saja <p>Lihat Pemecahan Masalah.</p>

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	<p>diperlukan untuk menggunakan skrip dan sampel, bahkan jika Anda tidak menggunakan (atau bermaksud menggunakan) Python SDK.</p> <ul style="list-style-type: none">• Rilis ini meningkatkan dukungan untuk Amazon Linux 2.• Kami memperbaiki beberapa bug di SimSpace Weaver Local. <p>Untuk informasi selengkapnya, lihat catatan rilis.</p> <p>Perbaikan bug</p> <ul style="list-style-type: none">• Kami memperbaiki bug yang menyebabkan entitas menjadi tidak dimiliki jika mereka tidak menyelesaikan transfer antar			

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	pekerja jarak jauh.			
1.16.0	<p>Fitur baru:</p> <ul style="list-style-type: none">• Sekarang Anda dapat menggunakan pesan APIs di SDK SimSpace Weaver aplikasi untuk mengirim dan menerima pesan antar aplikasi Anda. Fitur ini tersedia untuk integrasi C++, Python, dan Unity and Unreal Engine 5.	Februari 12, 2024	Lihat Katalog AWS SimSpace Weaver Panduan .	<ul style="list-style-type: none">• Paket lengkap• Perpustakaan saja <p>Lihat Pemecahan Masalah.</p>

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.15.3	<p>SimSpace Weaver Local memperbar ui:</p> <ul style="list-style-type: none">Kami berubah SimSpace Weaver Local untuk lebih menyelara skannya dengan pengembangan untuk AWS Cloud. Perubahan ini berdampak pada proyek dan alur kerja C ++, Python, Unity, dan Unreal Engine SimSpace Weaver Local.	Desember 4, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.15.2	<p>Pembaruan paket yang dapat didistribusikan SDK Aplikasi:</p> <ul style="list-style-type: none">Kami memperbarui <code>Dockerfile</code> untuk menggunakan versi khusus yang diperlukan <code>cmake</code>. Build kontainer Docker mungkin gagal tanpa perubahan ini.	November 2, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh
1.15.1	<p>Pembaruan fitur:</p> <ul style="list-style-type: none">Python SDK: Rilis ini memperbaiki masalah yang menyebabkan simulasi berbasis Python gagal di AWS Cloud	September 22, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.15.0	<p>Fitur baru:</p> <ul style="list-style-type: none">• Python SDK: Anda sekarang dapat mengembankan simulasi Anda dengan Python. Paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan mencakup template untuk proyek Python sampel dan klien tampilan Python.	31 Agustus 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.14.0	<p>Fitur baru</p> <ul style="list-style-type: none">• Kontainer khusus: Buat image container berbasis Amazon Linux 2 (AL2) Anda sendiri, simpan di Amazon Elastic Container Registry (Amazon ECR), dan gunakan untuk menjalankan aplikasi SimSpace Weaver Anda di file. AWS Cloud• Beberapa domain spasial: Buat lebih dari 1 domain spasial dalam simulasi. Pisahkan logika simulasi alih-alih menggabungkan semuanya menjadi satu aplikasi spasial. Alokasikan sumber daya yang berbeda ke domain spasial	26 Juli 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	<p>berdasarkan persyaratannya.</p> <ul style="list-style-type: none">• Tingkat centang tak terbatas: Aktifkan simulasi Anda berjalan secepat kode Anda dapat dijalankan. Setel jam simulasi Anda sehingga mengirimkan centang berikutnya segera setelah semua aplikasi menyelesaikan operasi komit mereka untuk centang saat ini. <p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none">• Nilai sekarang <code>tick_rate</code> adalah string. Nilai harus menyertakan tanda kutip ganda ("). Tingkat centang untuk versi sebelumnya			

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	a masih merupakan bilangan bulat.			
1.13.1	<p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none">• Pembaruan fitur: Pembuatan proyek sekarang bekerja dengan benar dengan PathfindingSampleUnreal template.	Juni 7, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.13.0	<p>SimSpace Weaver layanan APIs:</p> <ul style="list-style-type: none">• CreateSnapshot Aksi baru• Perubahan pada StartSimulation tindakan• Menambahkan <code>SnapshotS3Location</code> parameter untuk memulai dari snapshot.• <code>SchemaS3Location</code> Parameter sekarang opsional.• Perubahan DescribeSimulationOutput:<ul style="list-style-type: none">• <code>SchemaError</code> sudah usang.• Menambahkan	April 29, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	<p><code>StartError</code> bidang.</p> <ul style="list-style-type: none">• Menambahkan <code>SnapshotS3Location</code> bidang.• Menambahkan status <code>SNAPSHOT_IN_PROGRESS</code> simulasi.• Tipe S3Destination data baru <p>SimSpace Weaver konsol:</p> <ul style="list-style-type: none">• Fungsionalitas baru untuk membuat snapshot.• Fungsionalitas baru untuk memulai simulasi dari snapshot.			

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
	<p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none"> • Skrip baru untuk mendukung snapshot <ul style="list-style-type: none"> • <code>create-snapshot- <i>project-name</i> .bat</code> • <code>start-from-snapshot- <i>project-name</i> .bat</code> • <code>quick-start-from-snapshot- <i>project-name</i> -cli.bat</code> • <code>list-snapshots- <i>project-name</i> .bat</code> • Proyek sekarang menggunakan satu ember Amazon S3 per proyek: <code>weaver-<i>lowercase-project-name</i>-<i>account-number</i> <i>region</i></code> 			

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.12.3	<p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none">• Skrip berikut sekarang mendukung <code>--maximum-duration</code> parameter:• <code>quick-start-<i>project-name</i> - cli.bat</code>• <code>quick-start-<i>project-name</i> - cli.sh</code>• <code>start-simulation-<i>project-name</i> .bat</code>• <code>start-simulation-<i>project-name</i> .sh</code>• <code>run-<i>project-name</i> .bat</code>• <code>run-<i>project-name</i> .sh</code>	Maret 27, 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.12.2	<p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none">• Perbaiki bug: <code>docker-create-image.bat</code> sekarang berjalan dengan benar.	1 Maret 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

Versi	Catatan	Tanggal rilis	Dokumentasi	Unduhan SDK Aplikasi
1.12.1	<p>SimSpace Weaver aplikasi SDK:</p> <ul style="list-style-type: none">• Skrip sekarang menerima AWS CLI profil untuk digunakan untuk AWS otentikasi.• Skrip sekarang mendukung AWS IAM Identity Center AWS otentikasi. <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none">• Perbaikan bug: <code>Api::BeginUpdateWillBlock</code> sekarang kembali dengan benar <code>true</code> jika semua aplikasi spasial belum bergabung dengan simulasi.	28 Februari 2023	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh
1.12.0	Rilis untuk ketersediaan umum (GA)	29 November 2022	Lihat Katalog AWS SimSpace Weaver Panduan .	Tidak tersedia untuk diunduh

AWS SimSpace Weaver versi 1.17.0

Rilis ini merupakan perombakan paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan. Kami mengganti skrip Windows batch dan Linux Bash yang sudah ketinggalan zaman dengan skrip berbasis Python.

Important

Python 3.9 sekarang menjadi persyaratan untuk menggunakan skrip dan sampel, tidak hanya untuk Python SDK.

Daftar Isi

- [Perubahan besar untuk 1.17.0](#)
- [Perbarui proyek ke 1.17.0](#)
- [Pertanyaan yang sering diajukan tentang versi 1.17.0](#)

Perubahan besar untuk 1.17.0

- Pembuatan proyek yang disederhanakan
 - Setelah `berjalansetup.py`, Anda dapat membuat proyek Anda sendiri hanya dengan menyalin-menempelkan sampel.
- Sampel 1-klik
 - File zip distribusi sekarang berisi ready-to-use sampel yang berfungsi setelah mengatur distribusi.
- Setiap SDK sekarang ada di direktorinya sendiri: `cpp`, `pythonunreal`, dan `unity`. Anda mungkin harus memperbarui jalur tergantung pada SDK yang Anda gunakan.
- Perbaiki skrip pembantu.
 - Skrip sekarang berisi beberapa AWS CLI opsi untuk memaksimalkan fleksibilitasnya.
 - Peluncuran dan koneksi klien konsol terintegrasi sebagai bagian dari mulai cepat.
 - Output konsol yang ditingkatkan.
 - Pembuatan sampel Unreal dan Unity sekarang berfungsi `quick-start`, tidak ada lagi langkah manual yang diperlukan.

- SimSpace Weaver Local sekarang bekerja hanya dengan menelepon `quick-start`, tidak ada lagi pembuatan dan peluncuran manual.
- SimSpace Weaver Local `quick-start` memiliki dukungan terintegrasi untuk mencatat output aplikasi.
- SimSpace Weaver Local sekarang dapat diluncurkan di lingkungan non-GUI, seperti dalam sesi ssh.
- Fitur “wadah khusus” sekarang terintegrasi ke dalam `quick-start` skrip.
- Peningkatan dukungan Amazon Linux 2 (AL2): alur kerja skrip untuk Windows dan AL2 sekarang sebanding. Sebelumnya, AL2 proyek membutuhkan lebih banyak langkah manual dan SimSpace Weaver Local tidak didukung untuk AL2.
- Plugin Unreal Engine dan Unity sekarang disertakan sebagai bagian dari paket SimSpace Weaver aplikasi SDK yang dapat didistribusikan.
- Perbaiki bug untuk SimSpace Weaver Local
 - Memperbaiki bug di mana entitas dapat diberi ID entitas yang sama.
 - Memperbaiki bug di mana dua partisi dapat diberi ID partisi yang sama.
 - Memperbaiki bug yang terkait dengan aplikasi yang mencoba menulis ke entitas yang tidak mereka miliki.
 - Menyelesaikan masalah kebocoran memori.

Perbarui proyek ke 1.17.0

1. Siapkan distribusi 1.17.0: Pergi melalui prosedur pengaturan lagi karena kami mengubahnya untuk 1.17.0. Untuk informasi selengkapnya, lihat [Menyiapkan untuk SimSpace Weaver](#).
2. Setiap Weaver App SDK sekarang ada di direktorinya sendiri. Perbarui jalur build Anda untuk mencerminkan hal ini.
 - a. Direktori C++: `SimSpaceWeaverAppSdk/cpp`
 - SDK aplikasi SimSpace Weaver C ++ sekarang menggunakan file. `FindSimSpaceWeaverAppSdk.cmake` File ini menetapkan `weaver` target yang ditautkan, dan menyertakan perbaikan bug penting saat membangun Weaver di file. AWS Cloud Anda harus menggunakan ini alih-alih menautkan ke binari secara langsung.
 - b. Direktori Python: `SimSpaceWeaverAppSdk/python`
 - c. Plugin persatuan: `SimSpaceWeaverAppSdk/unity`

- d. Plugin Unreal Engine: `SimSpaceWeaverAppSdk/unreal`
3. `tools` skrip sebelumnya tidak akan berfungsi dengan SimSpace Weaver distribusi baru. Untuk menggunakan `tools` skrip baru dengan proyek Anda:
 - a. Hapus direktori lama `tools/windowstools/linux`, dan `tools/local` direktori Anda.
 - b. Salin `tools` direktori proyek contoh yang menggunakan SDK SimSpace Weaver aplikasi yang sama dengan project Anda. Pastikan Anda telah menjalankan `setup.py` sebelum menyalin direktori ini.

Important

Skrip alat hanya dijamin untuk bekerja dengan proyek sampel. Anda mungkin harus mengedit skrip ini, terutama `build.py` skrip, untuk bekerja dengan proyek Anda. Setiap pengeditan akan unik untuk proyek Anda, oleh karena itu kami tidak dapat memberikan panduan apa pun.

Pertanyaan yang sering diajukan tentang versi 1.17.0

Apakah saya harus memperbarui ke versi 1.17.0?

Ini bukan pembaruan yang diperlukan karena tidak ada perubahan pada SimSpace Weaver API atau SDK SimSpace Weaver aplikasi. Anda harus memperbarui ke 1.17.0 jika Anda ingin menggunakan 1.17.0 SimSpace Weaver Local, yang berisi beberapa perbaikan bug.

Berapa versi Python minimum yang diperlukan?

Python 3.9 adalah versi minimum.

Apa CMake versi minimum yang diperlukan?

CMake versi 3.13 adalah minimum.

Berapa versi minimum Unreal Engine yang diperlukan?

Unreal Engine 5.0 adalah minimum.

Apa versi minimum Unity yang diperlukan?

Unity versi 2022.3.19.F1 adalah minimum.

AWS SimSpace Weaver versi 1.15.1

Rilis ini adalah pembaruan yang diperlukan untuk Python SDK yang awalnya dirilis dalam SimSpace Weaver versi 1.15.0. Ini memperbaiki masalah ketidakcocokan versi yang menyebabkan simulasi berbasis Python gagal di. AWS Cloud Gunakan versi ini alih-alih 1.15.0.

Perbarui proyek Python yang ada ke 1.15.1

Jika Anda memiliki proyek Python yang sudah ada yang Anda buat dengan SDK Python versi 1.15.0, Anda harus melakukan langkah-langkah berikut untuk memperbaruinya ke 1.15.1 sehingga dapat berjalan di file. AWS Cloud

Alih-alih mengikuti prosedur ini, Anda juga dapat membuat proyek Python baru dengan 1.15.1 Python SDK dan memindahkan kode kustom Anda ke proyek baru.

Untuk memperbarui proyek Python 1.15.0 ke 1.15.1

1. Buka folder proyek Python Anda.
2. `src/PythonBubblesSample/bin/run-python`Ubah baris berikut:

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

Ke hal berikut:

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. Dalam `CMakeLists.txt` menghapus baris berikut:

- ```
file(COPY "${SDK_PATH}/libweaver_app_sdk_python_v1_${ENV{PYTHON_VERSION}}.so"
 DESTINATION "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1")
```
- ```
file(RENAME "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_  
  ${ENV{PYTHON_VERSION}}.so" "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1/  
  libweaver_app_sdk_python_v1.so")
```
- ```
message(" * COPYING WEAVER PYTHON SDK TO BUILD DIR ${ZIP_FILES_DIR}....")
```

- `file(COPY ${SDK_DIR} DESTINATION ${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1)`

## Pemecahan masalah untuk versi 1.15.1

Setelah memperbarui simulasi Python 1.15.0, gagal memulai di AWS Cloud

Gejala: Setelah sekitar 5-10 menit setelah Anda memulai simulasi, log manajemen simulasi melaporkan `internal error` dan status simulasi adalah. `FAILED`

Ini dapat terjadi jika file pustaka dari SDK Python 1.15.0 disertakan dalam file zip aplikasi. Pastikan Anda menyelesaikan langkah-langkah untuk memperbarui proyek Anda, dan pastikan itu `libweaver_app_sdk_python_v1.so` tidak ada dalam file zip Anda atau direferensikan dengan cara apa pun.

## Pertanyaan yang sering diajukan tentang versi 1.15.1

Apakah rilis ini memengaruhi apa pun selain SDK Python?

Tidak.

Apakah saya harus memperbarui ke versi 1.15.1?

Anda tidak perlu memperbarui ke 1.15.1 jika Anda tidak bermaksud menggunakan Python untuk aplikasi spasial Anda. Jika Anda memperbarui ke 1.15.0, simulasi berbasis Python Anda tidak akan berjalan di. AWS Cloud Kami menyarankan Anda memperbarui ke 1.15.1 jika Anda menggunakan 1.15.0.

Apa `$LD_LIBRARY_PATH`?

Ini adalah lokasi SDK Python saat simulasi Anda berjalan di file. AWS Cloud Ini baru untuk 1.15.1. Kami membuat perubahan ini untuk menghindari masalah versi Python di masa depan. Menautkan ke direktori itu secara fungsional sama dengan menautkan ke `libweaver_app_sdk_python_v1.so` 1.15.0.

# Riwayat dokumen untuk AWS SimSpace Weaver

Tabel berikut menjelaskan perubahan penting pada SimSpace Weaver dokumentasi.

| Tanggal           | Perubahan                    | Pembaruan dokumentasi                                                                                                                                                                                                                                                                                               | Versi API diperbarui |
|-------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Mei 20, 2025      | Pemberitahuan akhir dukungan | Pemberitahuan akhir dukungan: Pada 20 Mei 2026, AWS akan mengakhiri dukungan untuk AWS SimSpace Weaver. Setelah 20 Mei 2026, Anda tidak akan lagi dapat mengakses SimSpace Weaver konsol atau SimSpace Weaver sumber daya. Untuk informasi lebih lanjut, lihat <a href="#">AWS SimSpace Weaver akhir dukungan</a> . | N/A                  |
| April 17, 2024    | Konten diperbarui            | Diperbarui di seluruh panduan pengguna untuk rilis versi 1.17.0. Perubahan besar pada <a href="#">Pengaturan</a> chapter dan <a href="#">Memulai</a> tutorial. Lihat <a href="#">catatan rilis</a> untuk informasi lebih lanjut.                                                                                    | N/A                  |
| Februari 12, 2024 | Konten diperbarui            | Memperbarui <a href="#">AWS SimSpace Weaver versi</a> chapter untuk rilis versi 1.16.0.                                                                                                                                                                                                                             | N/A                  |
| Februari 12, 2024 | Konten baru                  | Ditambahkan <a href="#">Perpesanan</a> bagian sebagai bagian                                                                                                                                                                                                                                                        | N/A                  |

| Tanggal           | Perubahan         | Pembaruan dokumentasi                                                                                                                                                                                                                                 | Versi API diperbarui |
|-------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
|                   |                   | dari rilis versi 1.16.0. Bagian ini menjelaskan an pesan yang APIs ditambahkan ke SDK SimSpace Weaver aplikasi. Anda dapat menggunakannya APIs untuk mengirim dan menerima pesan antar aplikasi Anda.                                                 |                      |
| Februari 12, 2024 | Konten diperbarui | Diperbarui <a href="#">SimSpace Weaver referensi skema simulasi</a> chapter untuk versi 1.16.0.                                                                                                                                                       | N/A                  |
| Februari 12, 2024 | Konten diperbarui | Menambahkan kuota layanan untuk pengiriman pesan ke <a href="#">SimSpace Titik akhir dan kuota Weaver</a> chapter.                                                                                                                                    | N/A                  |
| Februari 12, 2024 | Panduan baru      | Pisahkan konten untuk versi sebelum 1.16.0 menjadi panduan terpisah. Ditambahk an <a href="#">AWS SimSpace Weaver Guide Catalog</a> (tersedia dari <a href="#">halaman arahan dokumentasi utama</a> ) untuk mengakses panduan untuk versi sebelumnya. | N/A                  |

| Tanggal          | Perubahan                  | Pembaruan dokumentasi                                                                                                                                                                                                             | Versi API diperbarui |
|------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Desember 4, 2023 | Konten diperbarui          | Memperbarui <a href="#">AWS SimSpace Weaver versi chapter</a> untuk rilis versi 1.15.3.                                                                                                                                           | N/A                  |
| Desember 4, 2023 | Konten diperbarui          | Memperbarui <a href="#">AWS SimSpace Weaver versi chapter</a> untuk menyertakan petunjuk instalasi untuk versi terbaru.                                                                                                           | N/A                  |
| Desember 4, 2023 | Konten diperbarui          | Diperbarui <a href="#">Kuota layanan untuk SimSpace Weaver Local</a> .                                                                                                                                                            | N/A                  |
| Desember 4, 2023 | Konten baru dan diperbarui | Merestrukturisasi <a href="#">Pembangunan lokal di SimSpace Weaver</a> bagian dan menambahkan halaman baru yang menjelaskan perbedaan untuk SimSpace Weaver Local diperkenalkan di versi 1.15.3.                                  | N/A                  |
| 7 November 2023  | Konten diperbarui          | Memperbarui instruksi untuk menyiapkan Docker dan WSL untuk menggunakan tautan unduhan langsung/URL untuk SDK aplikasi. Untuk informasi selengkapnya, lihat <a href="#">Siapkan lingkungan lokal Anda untuk SimSpace Weaver</a> . | N/A                  |

| Tanggal            | Perubahan         | Pembaruan dokumentasi                                                                                                                                                                                                                                                                                                                                                                           | Versi API diperbarui |
|--------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| November 2, 2023   | Konten diperbarui | Memperbarui halaman versi layanan untuk rilis 1.15.2. Untuk informasi selengkapnya, lihat <a href="#">Versi layanan</a> .                                                                                                                                                                                                                                                                       | N/A                  |
| 23 Oktober 2023    | Konten diperbarui | Memperbarui halaman versi layanan dengan instruksi baru untuk mengunduh paket aplikasi yang dapat didistribusikan SDK. Pelanggan sekarang hanya boleh menggunakan salah satu tautan unduhan langsung kami yang disetujui dan tidak menggunakan AWS CLI untuk mengunduh paket yang dapat didistribusikan SDK aplikasi. Untuk informasi selengkapnya, lihat <a href="#">Unduh versi terbaru</a> . | N/A                  |
| September 22, 2023 | Konten baru       | Menambahkan halaman catatan versi dengan petunjuk pembaruan untuk rilis 1.15.1. Untuk informasi selengkapnya, lihat <a href="#">AWS SimSpace Weaver versi 1.15.1</a> .                                                                                                                                                                                                                          | N/A                  |

| Tanggal            | Perubahan         | Pembaruan dokumentasi                                                                                                                                                                                                    | Versi API diperbarui |
|--------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| September 10, 2023 | Konten baru       | Menambahkan bagian pemecahan masalah untuk situasi di mana AWS CLI tidak dikenali. SimSpace Weaver Untuk informasi selengkapnya, lihat <a href="#">Yang AWS CLI tidak mengenali simspaceweaver</a> .                     | N/A                  |
| September 10, 2023 | Konten diperbarui | Instruksi instalasi yang diperbarui untuk AWS CLI di WSL. Untuk informasi selengkapnya, lihat <a href="#">Siapkan paket SimSpace Weaver distribusi untuk Amazon Linux 2 (AL2) di Windows Subsystem for Linux (WSL)</a> . | N/A                  |
| 7 September 2023   | Pembaruan API     | BucketName dan sekarang ObjectKey diperlukan untuk tipe data <a href="#">S3Location</a> . BucketName sekarang diperlukan untuk tipe data <a href="#">S3Destination</a> .                                                 | AWS SDK: 2023-09-07  |
| 31 Agustus 2023    | Konten baru       | Menambahkan bagian baru untuk rilis 1.15.0.: <a href="#">Bekerja menggunakan Python</a>                                                                                                                                  | N/A                  |

| Tanggal         | Perubahan         | Pembaruan dokumentasi                                                                                                                                                                                                             | Versi API diperbarui |
|-----------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| 15 Agustus 2023 | Konten diperbarui | Instruksi unduhan yang diperbarui <a href="#">AWS SimSpace Weaver versi</a> untuk hanya mencantumkan ember SimSpace Weaver Amazon S3 resmi. Lokasi unduhan lainnya tidak dikendalikan oleh AWS dan mungkin berisi kode berbahaya. | N/A                  |
| 26 Juli 2023    | Konten diperbarui | Diperbarui <a href="#">Jam</a> .                                                                                                                                                                                                  | N/A                  |
| 26 Juli 2023    | Konten diperbarui | Diperbarui <a href="#">Mengkonfigurasi domain spasial</a> .                                                                                                                                                                       | N/A                  |
| 26 Juli 2023    | Konten baru       | Menambahkan bagian baru: <a href="#">Wadah khusus</a> .                                                                                                                                                                           | N/A                  |
| 26 Juli 2023    | Konten diperbarui | Diperbarui <a href="#">AWS SimSpace Weaver versi</a> untuk rilis 1.14.0.                                                                                                                                                          | N/A                  |
| 6 Juli 2023     | Konten baru       | Menambahkan bagian baru: <a href="#">PathfindingSample klien konsol gagal terhubung</a> .                                                                                                                                         | N/A                  |
| Juni 7, 2023    | Konten diperbarui | Diperbarui <a href="#">AWS SimSpace Weaver versi</a> untuk rilis 1.13.1.                                                                                                                                                          | N/A                  |

| Tanggal        | Perubahan          | Pembaruan dokumentasi                                                                                                                                                                                                    | Versi API diperbarui |
|----------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| 15 Mei 2023    | Konten baru        | Menambahkan bagian baru: <a href="#">Menggunakan snapshot dengan AWS CloudFormation</a> .                                                                                                                                | N/A                  |
| April 29, 2023 | Konten baru        | Menambahkan konten untuk rilis 1.13.0. Untuk informasi selengkapnya, lihat <a href="#">AWS SimSpace Weaver versi</a> .                                                                                                   | AWS SDK: 2023-04-28  |
| Maret 27, 2023 | Konten baru        | Menambahkan bagian yang menjelaskan durasi maksimum simulasi. Menambahkan catatan dalam tutorial untuk rilis 1.12.3, yang menambahkan dukungan untuk --maximum-duration parameter ke skrip SDK SimSpace Weaver aplikasi. | N/A                  |
| 9 Maret 2023   | Konten yang diubah | Mengklarifikasi bahwa kami hanya memberikan instruksi untuk Docker on Windows dan for Windows Subsystem for Linux (WSL), dan bahwa WSL (dan lingkungan Linux lainnya) tidak didukung.                                    | N/A                  |

| Tanggal          | Perubahan          | Pembaruan dokumentasi                                                                                                                                  | Versi API diperbarui |
|------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| 28 Februari 2023 | Konten baru        | Ditambahkan sebuah chapter yang menjelaskan SimSpace Weaver versi.                                                                                     | N/A                  |
| 28 Februari 2023 | Konten yang diubah | Mengubah konten tentang otentikasi untuk menyertakan penggunaan AWS IAM Identity Center dan profil bernama untuk AWS Command Line Interface (AWS CLI). | N/A                  |
| 17 Februari 2023 | Konten baru        | Menambahkan bagian tentang mengelola sumber daya Anda dengan AWS CloudFormation.                                                                       | N/A                  |
| 23 Januari 2023  | Konten baru        | Menambahkan instruksi untuk men-debug simulasi lokal.                                                                                                  | N/A                  |
| 29 November 2022 | Peluncuran layanan | Merilis Panduan Pengguna dan Referensi API untuk SimSpace Weaver.                                                                                      | AWS SDK: 2022-11-29  |

# Glosarium

---

Glosarium ini mendefinisikan istilah-istilah yang khusus untuk AWS SimSpace Weaver

Untuk AWS terminologi terbaru, lihat [AWS glosarium di Referensi Umum.AWS](#)

## A

---

|              |                                                                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| aplikasi     | Kode yang dapat dieksekusi (juga disebut binari) yang Anda buat. Istilah aplikasi dapat merujuk ke kode atau instance kode yang sedang berjalan. Aplikasi merangkum perilaku simulasi. Aplikasi membuat, menghapus, membaca, dan memperbarui <a href="#">entitas</a> .                               |
| aplikasi SDK | Kit pengembangan perangkat lunak (SDK) yang Anda gunakan untuk mengintegrasikan aplikasi. SimSpace Weaver SDK menyediakan APIs untuk membaca dan menulis data <a href="#">entitas</a> dan melacak waktu simulasi. Untuk informasi selengkapnya, lihat <a href="#">SimSpace Weaver aplikasi SDK</a> . |

## C

---

|                             |                                                                                                                                                                                                                                                                     |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| klien                       | Proses (atau definisinya) yang ada di luar SimSpace Weaver dan berinteraksi dengan simulasi melalui <a href="#">aplikasi atau aplikasi layanan khusus</a> . Anda dapat menggunakan klien untuk melihat atau mengubah status simulasi.                               |
| jam                         | Abstraksi dari proses SimSpace Weaver penjadwalan internal. Jam menerbitkan <a href="#">kutu</a> ke <a href="#">aplikasi</a> untuk mempertahankan sinkronisasi waktu. Setiap simulasi memiliki jam sendiri.                                                         |
| tingkat jam                 | <a href="#">Jumlah kutu per detik yang diterbitkan jam ke aplikasi</a> . Untuk informasi selengkapnya tentang tarif jam yang didukung, lihat <a href="#">SimSpace Titik akhir dan kuota Weaver</a> .                                                                |
| tingkat centang jam         | Lihat <a href="#">kecepatan jam</a> .                                                                                                                                                                                                                               |
| menghitung unit sumber daya | Unit sumber daya komputasi (prosesor dan memori) pada <a href="#">pekerja</a> . Satu instance <a href="#">aplikasi</a> biasanya dialokasikan 1 unit sumber daya komputasi. Anda dapat mengalokasikan lebih dari 1 unit sumber daya komputasi untuk setiap aplikasi. |

aplikasi kustom Jenis [aplikasi](#) yang Anda gunakan untuk membaca dan berinteraksi dengan keadaan simulasi. Aplikasi khusus dapat membuat entitas dalam simulasi tetapi tidak memilikinya. Saat aplikasi kustom membuat entitas, aplikasi tersebut harus mentransfer entitas ke [domain spasial](#). Anda mengontrol siklus hidup aplikasi kustom menggunakan aplikasi APIs. Untuk informasi lebih lanjut tentang SimSpace Weaver APIs, lihat [SimSpace Weaver Referensi API](#).

domain kustom [Domain](#) yang berisi [aplikasi khusus](#).

partisi kustom [Partisi aplikasi kustom](#).

## D

---

tenggat waktu [Waktu aktual](#) dimana operasi (seperti pemrosesan untuk tanda [centang](#)) harus selesai.

domain Sekelompok instance [aplikasi](#) yang menjalankan kode yang dapat dieksekusi yang sama (biner aplikasi) dan memiliki opsi peluncuran yang sama.

## E

---

titik akhir (layanan) Nama domain yang sepenuhnya memenuhi syarat (FQDN) yang digunakan program (seperti AWS Command Line Interface) untuk terhubung ke layanan. SimSpace Weaver

titik akhir (simulasi) Alamat IP dan nomor port yang digunakan klien untuk terhubung ke simulasi. Anda dapat mengonfigurasi titik akhir pada [aplikasi khusus dan aplikasi layanan](#).

entitas Objek data pelanggan (atau definisinya). Entitas dapat statis (tetap di satu lokasi) atau dinamis (bergerak melalui ruang simulasi). Misalnya, orang dan bangunan dalam simulasi.

## I

---

indeks (simulasi) Deskripsi sifat spasial simulasi, termasuk batas spasial dan sistem koordinatnya.

---

## L

|                              |                                                                                                                                                                                                                                                |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| siklus hidup (dari aplikasi) | Deskripsi langkah-langkah logis yang diharapkan yang dilalui <a href="#">aplikasi</a> selama simulasi. Siklus hidup dikelola (SimSpace Weaver memulai dan menghentikan aplikasi) atau tidak dikelola (Anda memulai dan menghentikan aplikasi). |
| beban (data bidang entitas)  | Baca data bidang <a href="#">entitas</a> dari <a href="#">State Fabric</a> .                                                                                                                                                                   |

---

## P

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| partisi | Segmen memori bersama pada <a href="#">pekerja</a> . <a href="#">Setiap partisi berisi subset diskrit entitas dalam domain</a> . Setiap <a href="#">aplikasi</a> memiliki partisi yang ditetapkan. Aplikasi memiliki semua entitas di partisinya. Ketika sebuah aplikasi membuat entitas, itu membuatnya di partisinya. Ketika entitas berpindah dari satu partisi ke partisi lain, kepemilikan akan ditransfer dari aplikasi partisi sumber ke aplikasi partisi tujuan. |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## R

|                  |                             |
|------------------|-----------------------------|
| unit sumber daya | Lihat <a href="#">???</a> . |
|------------------|-----------------------------|

---

## D

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| skema            | Dokumen YAMAL atau JSON yang menjelaskan konfigurasi simulasi. SimSpace Weaver menggunakan skema untuk membuat sumber daya <a href="#">simulasi</a> .                                                                                                                                                                                                                                                                                                         |
| aplikasi layanan | Jenis <a href="#">aplikasi</a> yang Anda gunakan untuk membaca dan berinteraksi dengan keadaan simulasi. Aplikasi layanan dapat membuat entitas dalam simulasi tetapi harus mentransfernya ke <a href="#">domain</a> spasial. SimSpace Weaver mengelola <a href="#">siklus hidup</a> aplikasi layanan, dan memulai 1 (atau lebih, seperti yang ditentukan dalam <a href="#">skema</a> simulasi Anda) pada setiap <a href="#">pekerja</a> dalam simulasi Anda. |
| domain layanan   | <a href="#">Domain</a> yang berisi <a href="#">aplikasi layanan</a> .                                                                                                                                                                                                                                                                                                                                                                                         |
| partisi layanan  | <a href="#">Partisi aplikasi layanan</a> .                                                                                                                                                                                                                                                                                                                                                                                                                    |

|                                 |                                                                                                                                                                                                                                                                           |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| simulasi (sumber daya)          | Abstraksi cluster komputasi yang menjalankan ruang virtual simulasi. Anda dapat memiliki beberapa simulasi. Anda mengkonfigurasi simulasi menggunakan <a href="#">skema</a> .                                                                                             |
| aplikasi spasial                | Jenis <a href="#">aplikasi</a> yang merangkum logika simulasi inti. Setiap aplikasi spasial memiliki 1 (dan hanya 1) <a href="#">partisi</a> .                                                                                                                            |
| domain spasial                  | <a href="#">Domain</a> yang berisi <a href="#">aplikasi spasial</a> .                                                                                                                                                                                                     |
| partisi spasial                 | <a href="#">Partisi aplikasi spasial</a> .                                                                                                                                                                                                                                |
| State Fabric                    | SimSpace Weaver Database dalam memori. State Fabric Menyimpan keadaan simulasi, termasuk entitas dan SimSpace Weaver data internal.                                                                                                                                       |
| menyimpan (data bidang entitas) | Tulis data bidang entitas ke file <a href="#">State Fabric</a> .                                                                                                                                                                                                          |
| berlangganan                    | Permintaan jangka panjang untuk instance <a href="#">aplikasi</a> tertentu untuk menerima data dari <a href="#">area langganan</a> . Aplikasi berlangganan menggunakan langganan untuk menemukan perubahan pada <a href="#">entitas</a> di dalam area langganan.          |
| area berlangganan               | Wilayah 2 dimensi dari ruang simulasi. <a href="#">Langganan</a> mengacu pada area berlangganan. Area berlangganan dapat menjangkau lebih dari 1 <a href="#">partisi</a> , dan juga mencakup bagian partisi. Area berlangganan berkelanjutan dalam batas yang ditentukan. |

## T

|                 |                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| centang         | Nilai diskrit untuk waktu (baik waktu jam dinding atau waktu simulasi). <a href="#">Aplikasi</a> dapat melakukan iterasi lebih cepat dari durasi centang, tetapi diharapkan untuk menulis kutu tertentu dalam tenggat waktu tertentu. Semua operasi untuk semua aplikasi untuk tanda centang tertentu harus selesai sebelum centang berikutnya dapat dimulai. |
| tingkat centang | Lihat kecepatan jam.                                                                                                                                                                                                                                                                                                                                          |
| waktu (aktual)  | Waktu saat ini dari perspektif realitas. SimSpace Weaver menggunakan stempel waktu POSIX 64-bit yang merupakan jumlah nanodetik sejak zaman Unix (January 1, 1970, 00:00:00 UTC).                                                                                                                                                                             |

waktu (simulasi)

Waktu saat ini dari perspektif simulasi. SimSpace Weaver menggunakan penghitung centang logis integer 64-bit, yang mungkin tidak secara langsung sesuai dengan waktu sebenarnya.

## W

---

pekerja

Instans Amazon Elastic Compute Cloud (Amazon EC2) yang menjalankan kode simulasi.