



Benutzerhandbuch für Version 1.17.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Benutzerhandbuch für Version 1.17.0

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

.....	ix
Was ist SimSpace Weaver?	1
Die wichtigsten Konzepte	1
Wie SimSpace Weaver funktioniert	2
Wie benutzt du SimSpace Weaver	5
Simulationsschema	6
Arbeiter und Ressourceneinheiten	6
Simulationsuhr	7
Partitionen	7
State Fabric	7
Entitäten	8
Apps	8
Beispielanwendungsfälle	11
AWS SimSpace Weaver Ende des Supports	13
Einrichtung	14
Richten Sie Ihr Konto ein	14
Melde dich an für eine AWS-Konto	14
Erstellen eines Benutzers mit Administratorzugriff	15
Fügen Sie zu verwendende Berechtigungen hinzu SimSpace Weaver	16
Richten Sie Ihre lokale Umgebung ein	18
AL2 in Docker	19
AL2 in WSL	20
Verwendung von lizenzierte Software	24
Erste Schritte	25
Schnellstart-Tutorial	25
Schritt 1: Protokollierung aktivieren (optional)	26
Schritt 2: Schnellstart mit dem Konsolen-Client (Option 1)	27
Schritt 2: Schnellstart mit dem Unreal Engine-Client (Option 2)	27
Stoppen und löschen Sie Ihre Simulation	27
Fehlerbehebung	28
Ausführliches Tutorial	28
Schritt 1: Protokollierung aktivieren (optional)	29
Schritt 2: Starten Sie Ihre Simulation	29
Schritt 3: Überprüfen Sie die Protokolle (optional)	36

Schritt 4: Sehen Sie sich Ihre Simulation an	38
Schritt 5: Stoppen und löschen Sie Ihre Simulation	39
Fehlerbehebung	40
Arbeitet mit SimSpace Weaver	41
Konfiguration Ihrer Simulation	41
Konfigurationsparameter für die Simulation	42
SDK-Version	43
Eigenschaften der Simulation	43
Worker	44
Uhr	45
Partitionierungsstrategien	48
Domains	49
Maximale Dauer	60
Maximaler Wert	60
Standardwert	60
Mindestwert	60
Starten einer Simulation mit der Konsole	60
Der Status einer Simulation, die ihre maximale Dauer erreicht	61
Apps entwickeln	61
Räumliche Apps	62
Benutzerdefinierte Apps	62
Entwicklung von Kundenanwendungen	63
Holen Sie sich die IP-Adresse und die Portnummer	64
Den Unreal Engine View Client starten	67
Fehlerbehebung	68
Lokale Entwicklung	69
Schritt 1: Starten Sie Ihre lokale Simulation	70
Schritt 2: Sehen Sie sich Ihre lokale Simulation an	71
Schritt 3: Beenden Sie Ihre lokale Simulation (optional unter Windows)	72
Problembehandlung bei der lokalen Entwicklung	73
SimSpace Weaver App-SDK	73
API-Methoden geben a zurück Result	74
Interaktion mit dem App-SDK auf oberster Ebene	75
Verwaltung der Simulation	75
Subscriptions (Abonnements)	78
Entitäten	79

Ereignisse der Entität	91
Result und Fehlerbehandlung	98
Generika und Domaintypen	100
Verschiedene App-SDK-Operationen	100
SimSpace Weaver Demo-Framework	103
Mit Kontingenten arbeiten	104
Holen Sie sich die Grenzwerte für eine App	104
Ruft die Menge der von einer App verwendeten Ressourcen ab	105
Metriken zurücksetzen	106
Überschreitung eines Limits	107
Der Speicher geht zur Neige	107
Bewährte Methoden	107
Simulationen debuggen	108
Verwenden Sie SimSpace Weaver Local und schauen Sie sich die Konsolenausgabe an	108
Sehen Sie sich Ihre Logs in Amazon CloudWatch Logs an	108
Verwenden Sie Beschreiben API-Aufrufe	109
Einen Client Connect	109
Debuggen lokaler Simulationen	110
Benutzerdefinierte Behälter	111
Erstellen Sie einen benutzerdefinierten Container	111
Ändern Sie ein Projekt, um einen benutzerdefinierten Container zu verwenden	113
Häufig gestellte Fragen	116
Fehlerbehebung	116
Verwenden von Python	117
Ein Python-Projekt erstellen	118
Eine Python-Simulation starten	119
Der Python-Beispielclient	120
Häufig gestellte Fragen	121
Fehlerbehebung	121
Support für andere Motoren	123
Unity	123
Unreal Engine	123
Verwendung von lizenzierter Software	123
Ressourcen verwalten mit AWS CloudFormation	124
Snapshots	126
Snapshots	127

Anwendungsfälle für Schnappschüsse	128
SimSpace Weaver Konsole	129
AWS CLI	131
Häufig gestellte Fragen	133
Messaging	134
Anwendungsfälle für Messaging	134
Das Messaging verwenden APIs	135
Wann sollte Messaging verwendet werden	143
Tipps für die Arbeit mit Nachrichten	146
Fehler bei der Nachrichtenübermittlung und Problembehandlung	148
Bewährte Methoden	151
Richten Sie Abrechnungsalarme ein	151
Verwenden von SimSpace Weaver Local	151
Beenden Sie Simulationen, die Sie nicht benötigen	152
Löschen Sie Ressourcen, die Sie nicht benötigen	152
Haben Sie Backups	152
Sicherheit	154
Datenschutz	155
Verschlüsselung im Ruhezustand	156
Verschlüsselung während der Übertragung	156
Datenschutz für den Datenverkehr zwischen Netzwerken	157
Identitäts- und Zugriffsverwaltung	157
Zielgruppe	158
Authentifizierung mit Identitäten	158
Verwalten des Zugriffs mit Richtlinien	162
Wie AWS SimSpace Weaver funktioniert mit IAM	165
Beispiele für identitätsbasierte Richtlinien	173
Berechtigungen, die für Sie SimSpace Weaver erstellt werden	177
Serviceübergreifende Confused-Deputy-Prävention	179
Fehlerbehebung	182
Protokollierung und Überwachung von Sicherheitsereignissen	185
Compliance-Validierung	186
Ausfallsicherheit	188
Sicherheit der Infrastruktur	188
Sicherheitsmodell für Netzwerkkonnektivität	188
Konfigurations- und Schwachstellenanalyse	189

Bewährte Methoden für die Gewährleistung der Sicherheit	190
Verschlüsseln Sie die Kommunikation zwischen Ihren Apps und ihren Clients	190
Erstellen Sie regelmäßig eine Sicherungskopie Ihres Simulationsstatus	190
Pflegen Sie Ihre Apps und SDKs	191
Protokollierung und Überwachung	192
Meldet sich an CloudWatch	193
Greife auf deine SimSpace Weaver Logs zu	193
SimSpace Weaver Logs	193
Überwachung mit CloudWatch	196
SimSpace Weaver Metriken auf Kontoebene	196
CloudTrail protokolliert	196
SimSpace Weaver Informationen in CloudTrail	197
SimSpace Weaver Logdateieinträge verstehen	198
Endpunkte und Servicekontingenten	200
Service-Endpunkte	200
Servicekontingente	201
Kontingente für Nachrichten	204
Taktraten	204
Servicekontingenten für SimSpace Weaver Local	205
Fehlerbehebung	207
AssumeRoleAccessDenied	207
InvalidBucketName	209
ServiceQuotaExceededException	210
TooManyBuckets	210
Die Erlaubnis wurde während des Simulationsstarts verweigert	211
Probleme im Zusammenhang mit der Zeit bei der Verwendung Docker	212
Der Konsolentclient kann keine Verbindung herstellen	212
Nein <code>simspaceweaver</code> in der AWS CLI	214
Schemareferenz	216
Beispiel für ein vollständiges Schema	216
Schemaformat	218
SDK-Version	219
Eigenschaften der Simulation	219
Worker	221
Uhr	222
Strategien zur Partitionierung	222

Domains	224
Einschränkungen bei der Platzierung	235
API-Referenzen	236
SimSpace Weaver Versionen	237
Aktuelle Version	237
So finden Sie Ihre aktuelle Version	237
Laden Sie die neueste Version herunter	237
Problembehandlung bei App-SDK-Downloads	238
Installieren Sie die neueste Version	239
Versionen der Dienste	239
1.17.0	254
Wesentliche Änderungen für 1.17.0	254
Ein Projekt auf Version 1.17.0 aktualisieren	255
Häufig gestellte Fragen zu Version 1.17.0	256
1.15.1	257
Aktualisieren Sie ein vorhandenes Python-Projekt auf 1.15.1	257
Fehlerbehebung für Version 1.15.1	258
Häufig gestellte Fragen zu Version 1.15.1	258
Dokumentverlauf	260
Glossar	270

Hinweis zum Ende des Supports: Am 20. Mai 2026 AWS endet der Support für AWS SimSpace Weaver. Nach dem 20. Mai 2026 können Sie nicht mehr auf die SimSpace Weaver Konsole oder SimSpace Weaver die Ressourcen zugreifen. Weitere Informationen finden Sie unter [AWS SimSpace Weaver Ende des Supports](#).

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Was ist AWS SimSpace Weaver?

AWS SimSpace Weaver ist ein Dienst, mit dem Sie groß angelegte räumliche Simulationen in der AWS Cloud erstellen und ausführen können. Sie können beispielsweise Menschengenerationen, große reale Umgebungen und immersive und interaktive Erlebnisse erstellen.

Mit SimSpace Weaver können Sie Simulationsworkloads auf mehrere Amazon Elastic Compute Cloud (Amazon EC2) -Instances verteilen. SimSpace Weaver stellt die zugrunde liegende AWS Infrastruktur für Sie bereit und kümmert sich um das Simulationsdatenmanagement und die Netzwerkkommunikation zwischen den EC2 Amazon-Instances, auf denen Ihre Simulation ausgeführt wird.

Schlüsselkonzepte für SimSpace Weaver

Eine Simulation oder ein Spiel wird durch den Computer begrenzt, auf dem sie ausgeführt werden. Je größer und komplexer Ihre virtuelle Welt wird, desto geringer wird die Verarbeitungsleistung. Berechnungen dauern länger, den Systemen geht der Arbeitsspeicher aus und die Client-Frameraten sinken. Bei Simulationen, die keine Echtzeitleistung benötigen, kann dies nur ärgerlich sein. Oder es könnte sich um eine geschäftskritische Situation handeln, in der erhöhte Verarbeitungsverzögerungen zu höheren Kosten führen. Wenn Ihre Simulation oder Ihr Spiel Echtzeitleistung benötigt, ist Leistungseinbußen definitiv ein Problem.

Eine gängige Lösung für eine Simulation, die eine Leistungsgrenze erreicht, besteht darin, die Simulation zu vereinfachen. Online-Spiele mit vielen Benutzern lösen Skalierungsprobleme häufig dadurch, dass sie Kopien ihrer virtuellen Welt auf verschiedenen Servern erstellen und die Benutzer auf diese verteilen.

SimSpace Weaver löst das Skalierungsproblem, indem es Ihre virtuelle Welt räumlich aufteilt und die Teile auf einen Cluster von Recheninstanzen verteilt, die in der AWS Cloud Die Recheninstanzen arbeiten zusammen, um die gesamte Simulationswelt parallel zu verarbeiten. Ihre Simulationswelt erscheint als ein einziger integrierter Raum für alles, was sich darin befindet, und für alle Clients, die sich mit ihr verbinden. Sie müssen eine Simulation nicht mehr aufgrund einer Hardwareleistungsgrenze vereinfachen. Sie können stattdessen mehr Rechenkapazität in der Cloud hinzufügen.

Themen

- [Wie SimSpace Weaver funktioniert](#)

- [Wie benutzt du SimSpace Weaver](#)
- [Simulationsschema](#)
- [Arbeiter und Ressourceneinheiten](#)
- [Simulationsuhr](#)
- [Partitionen](#)
- [State Fabric](#)
- [Entitäten](#)
- [Apps](#)

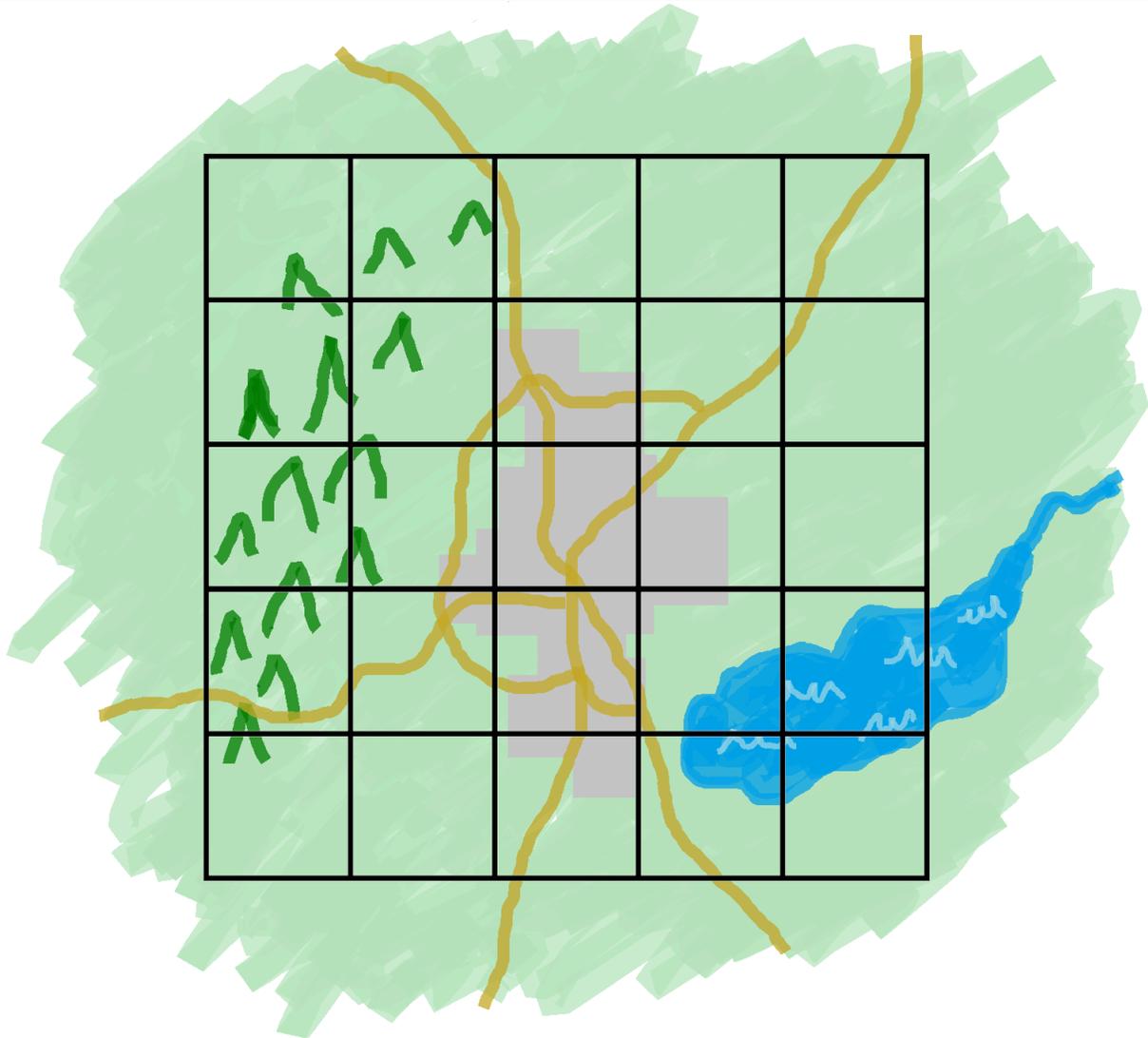
Wie SimSpace Weaver funktioniert

Ihre Simulation besteht aus einer Welt mit Objekten darin. Einige Objekte (wie Menschen und Fahrzeuge) bewegen sich und tun Dinge. Andere Objekte (wie Bäume und Gebäude) sind statisch. SimSpace Weaver In ist eine Entität ein Objekt in Ihrer Simulationswelt.

Sie definieren die Grenzen Ihrer Simulationswelt und unterteilen sie in ein Raster. Anstatt eine Simulationslogik zu erstellen, die auf dem gesamten Netz funktioniert, erstellen Sie eine Simulationslogik, die auf einer Zelle des Netzes funktioniert. Bei SimSpace Weaver einer Geo-App handelt es sich um ein von Ihnen geschriebenes Programm, das die Simulationslogik für eine Zelle Ihres Grids implementiert. Dies beinhaltet die Logik für alle Entitäten in dieser Zelle. Der Eigentümerbereich einer Geo-App ist die Rasterzelle, die von der Geo-App gesteuert wird.

Note

SimSpace Weaver In kann sich der Begriff „App“ auf den Code einer App oder eine laufende Instanz dieses Codes beziehen.



Ihre Simulationssimulation ist in ein Gitter unterteilt

Sie teilen Ihre Simulationssimulation in ein Gitter auf. Jede räumliche App implementiert Simulationslogik für eine einzelne Zelle in diesem Raster.

SimSpace Weaver führt eine Instanz Ihres Geo-App-Codes für jede Zelle Ihres Grids aus. Alle Spatial-App-Instanzen werden parallel ausgeführt. SimSpace Weaver Unterteilt Ihre Gesamtsimulation im Wesentlichen in mehrere kleinere Simulationen. Jede der kleineren Simulationen behandelt einen Teil der gesamten Simulationssimulation. SimSpace Weaver kann diese kleineren Simulationen auf mehrere Amazon Elastic Compute Cloud (Amazon EC2) -Instances

(sogenannte Worker) verteilen und ausführen AWS Cloud. Ein einzelner Worker kann mehrere räumliche Apps ausführen.

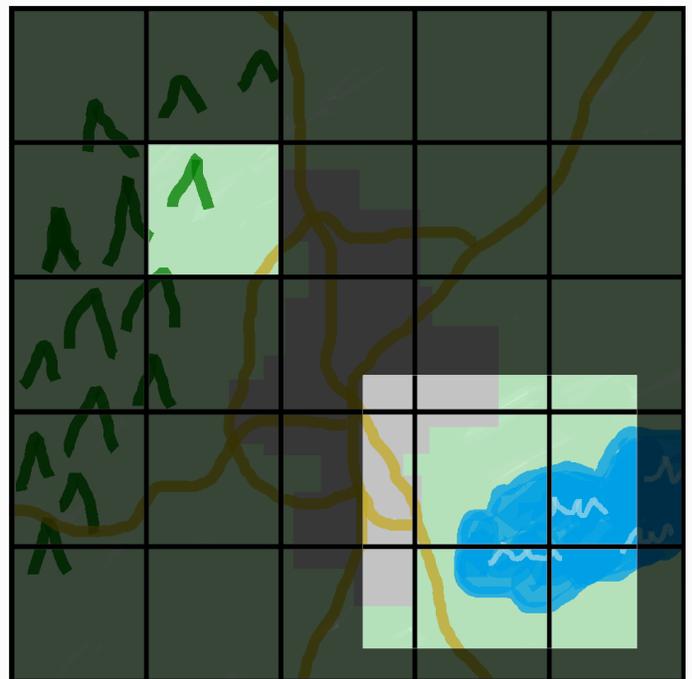
Entitäten können sich durch die Simulationswelt bewegen. Wenn eine Entität den Besitzbereich einer anderen Geo-App betritt (eine weitere Zelle im Raster), dann übernimmt der Besitzer der Geo-App die Kontrolle über die Entität. Wenn Ihre Simulation auf mehreren Workern ausgeführt wird, könnte eine Entität von der Kontrolle über eine Geo-App auf einem Worker zu einer Geo-App auf einem anderen Worker wechseln. Wenn eine Entität zu einem anderen Worker wechselt, SimSpace Weaver übernimmt sie die zugrunde liegende Netzwerkkommunikation.

Subscriptions (Abonnements)

Die Sicht einer Geo-App auf die Welt ist ihr eigener Zuständigkeitsbereich. Um herauszufinden, was in einem anderen Teil der Simulationswelt passiert, erstellt die Geo-App ein Abonnement. Der Abonnementbereich ist eine Teilmenge des gesamten Bereichs der Simulationswelt. Ein Abonnementbereich kann Teile mehrerer Eigentumsbereiche umfassen, einschließlich des Eigentumsbereichs der Geo-App. SimSpace Weaver benachrichtigt die Geodaten-App über alle Entitätsereignisse (z. B. Eingabe, Verlassen, Erstellen, Aktualisieren und Löschen), die innerhalb des Abonnementbereichs auftreten.



Die Sicht einer Geo-App auf die Welt



Die Ansicht einer räumlichen App mit einem zusätzlichen Abonnementbereich

Das Weltbild einer Geo-App ist ihr Eigentumsbereich, der eine Zelle im Weltraster darstellt.

Eine Geo-App verwendet ein Abonnement, um herauszufinden, was in einem anderen Teil der Simulationswelt passiert. Der Abonnementbereich kann mehrere Rasterzellen und Teile von Zellen enthalten.

Beispielsweise muss eine App, die Entitäten simuliert, die physisch interagieren, möglicherweise über Entitäten Bescheid wissen, die sich unmittelbar hinter den räumlichen Grenzen ihres Eigentumsbereichs befinden. Um dies zu erreichen, kann die App Bereiche abonnieren, die an ihren Eigentumsbereich grenzen. Nach der Erstellung des Abonnements erhält die App Benachrichtigungen über Entitätenereignisse in diesen Bereichen und kann die Entitäten lesen. Ein anderes Beispiel ist ein autonomes Fahrzeug, das alle Entitäten 200 Meter vor sich sehen muss, unabhängig davon, welcher App das Gebiet gehört. Die App für das Fahrzeug kann ein Abonnement mit einem Filter als nach der Achse ausgerichteter Begrenzungsrahmen erstellen (AABB) das deckt den sichtbaren Bereich ab.

Sie können Simulationslogik erstellen, die nicht für die Verwaltung der räumlichen Aspekte Ihrer Simulation verantwortlich ist. Eine benutzerdefinierte App ist ein ausführbares Programm, das auf einem einzelnen Worker ausgeführt wird. Sie steuern den Lebenszyklus (Start und Stopp) einer benutzerdefinierten App. Simulationsclients können eine Verbindung zu einer benutzerdefinierten App herstellen, um die Simulation anzusehen oder mit ihr zu interagieren. Sie können auch eine Service-App erstellen, die auf jedem Worker ausgeführt wird. SimSpace Weaver startet eine Instanz Ihrer Service-App auf jedem Worker, der Ihre Simulation ausführt.

Benutzerdefinierte Apps und Service-Apps erstellen Abonnements, um mehr über Entitätsereignisse zu erfahren und Entitäten zu lesen. Für diese Apps gibt es keine Eigentumsbereiche, da sie nicht räumlich begrenzt sind. Nur mit einem Abonnement können sie herausfinden, was in der Simulationswelt passiert.

Wie benutzt du SimSpace Weaver

Bei der Verwendung SimSpace Weaver sind dies die wichtigsten Schritte, die Sie befolgen müssen:

1. Schreiben und bauen C++ Apps, die das SimSpace Weaver App-SDK integrieren.
 - a. Ihre Apps führen API-Aufrufe durch, um mit dem Simulationsstatus zu interagieren.
2. Schreiben Sie Kunden, die sich Ihre Simulation über einige Apps ansehen und mit ihr interagieren.
3. Konfigurieren Sie Ihre Simulation in einer Textdatei.

4. Laden Sie Ihre App-Pakete und die Simulationskonfiguration in den Dienst hoch.
5. Starten Sie Ihre Simulation.
6. Starten und beenden Sie Ihre benutzerdefinierten Apps nach Bedarf.
7. Verbinden Sie Kunden mit Ihren benutzerdefinierten Apps oder Service-Apps, um die Simulation anzusehen oder mit ihr zu interagieren.
8. Überprüfen Sie Ihre Simulationsprotokolle in Amazon CloudWatch Logs.
9. Stoppen Sie Ihre Simulation.
10. Säubere deine Simulation.

Simulationsschema

Das Simulationsschema (oder Schema) ist eine YAML-formatierte Textdatei, die Konfigurationsinformationen für Ihre Simulation enthält. SimSpace Weaver verwendet Ihr Schema, wenn es eine Simulation startet. Das verteilbare SimSpace Weaver App-SDK-Paket enthält ein Schema für ein Beispielprojekt. Sie können dies als Ausgangspunkt für Ihr eigenes Schema verwenden. Weitere Informationen zum Simulationsschema finden Sie unter [SimSpace Weaver Referenz zum Simulationsschema](#).

Arbeiter und Ressourceneinheiten

Ein Worker ist eine EC2 Amazon-Instance, die Ihre Simulation ausführt. Sie geben in Ihrem Simulationsschema einen Worker-Typ an. SimSpace Weaver ordnet Ihren Worker-Typ einem bestimmten EC2 Amazon-Instance-Typ zu, den der Service verwendet. SimSpace Weaver startet und stoppt Ihre Mitarbeiter für Sie und verwaltet die Netzwerkkommunikation zwischen den Mitarbeitern. SimSpace Weaver startet für jede Simulation eine Gruppe von Arbeitern. Verschiedene Simulationen verwenden unterschiedliche Arbeiter.

Die verfügbare Rechenkapazität (Prozessor und Arbeitsspeicher) eines Workers ist in logische Einheiten unterteilt, die als Rechenressourceneinheiten (oder Ressourceneinheiten) bezeichnet werden. Eine Ressourceneinheit stellt eine feste Menge an Prozessor- und Speicherkapazität dar.

Note

Wir haben eine Rechenressourceneinheit zuvor als Steckplatz bezeichnet. Möglicherweise finden Sie diesen früheren Begriff immer noch in unserer Dokumentation.

Simulationsuhr

Jede Simulation hat ihre eigene Uhr. Sie starten und stoppen die Uhr mithilfe von API-Aufrufen oder der SimSpace Weaver Konsole. Die Simulation wird nur aktualisiert, wenn die Uhr läuft. Alle Operationen in der Simulation finden innerhalb von Zeitabschnitten statt, die als Ticks bezeichnet werden. Die Uhr kündigt allen Arbeitern die Startzeit jedes Ticks an.

Die Taktrate (oder Tickrate) ist die Anzahl der Ticks pro Sekunde (Hertz oder Hz), die die Uhr ankündigt. Die gewünschte Taktrate für eine Simulation ist Teil des Simulationsschemas. Alle Operationen für einen Tick müssen abgeschlossen sein, bevor der nächste Tick beginnt. Aus diesem Grund kann die effektive Taktrate niedriger als die gewünschte Taktrate sein. Die effektive Taktrate wird nicht höher als die gewünschte Taktrate sein.

Partitionen

Eine Partition ist ein Segment des gemeinsam genutzten Speichers auf einem Worker. Jede Partition enthält einen Teil der Simulationsstatusdaten.

Eine Partition für eine räumliche App (auch als räumliche App-Partition oder räumliche Partition bezeichnet) enthält alle Entitäten im Eigentumsbereich einer Geo-App. SimSpace Weaver ordnet Entitäten basierend auf der räumlichen Position der einzelnen Entitäten in räumliche App-Partitionen ein. Das bedeutet, dass SimSpace Weaver versucht wird, Entitäten, die räumlich nahe beieinander liegen, auf demselben Worker zu platzieren. Dadurch wird die Menge an Wissen, die eine App über Entitäten benötigt, die sie nicht besitzt, minimiert, um die Entitäten zu simulieren, die ihr gehören.

State Fabric

Die State Fabric ist das System des gemeinsamen Speichers (die Sammlung aller Partitionen) auf allen Workern. Sie enthält alle Zustandsdaten für Ihre Simulation.

Die State Fabric verwendet ein benutzerdefiniertes Binärformat, das eine Entität als Satz von Anfangsdaten und als Aktualisierungsprotokoll für jedes Datenfeld dieser Entität beschreibt. Mit diesem Format können Sie auf den Status einer Entität zu einem früheren Zeitpunkt in der Simulation zugreifen und ihn einem Zeitpunkt in der realen Zeit zuordnen. Der Puffer hat eine begrenzte Größe, und es ist nicht möglich, über das hinaus, was sich im Puffer befindet, in der Zeit zurückzugehen. SimSpace Weaver verwendet für jedes Feld einen Zeiger auf den aktuellen Offset im Aktualisierungsprotokoll und aktualisiert einen Zeiger als Teil einer Feldaktualisierung. SimSpace Weaver ordnet diese Aktualisierungsprotokolle mithilfe von Shared Memory dem Prozessraum einer App zu.

Dieses Objektformat führt zu einem geringen Overhead und zu keinen Serialisierungskosten. SimSpace Weaver verwendet dieses Objektformat auch, um Indexfelder (z. B. die Position der Entität) zu analysieren und zu identifizieren.

Entitäten

Eine Entität ist der kleinste Datenbaustein in Ihrer Simulation. Beispiele für Entitäten sind Akteure (wie Personen und Fahrzeuge) und statische Objekte (wie Gebäude und Hindernisse). Entitäten haben Eigenschaften (wie Position und Ausrichtung), in denen Sie persistente Daten speichern können SimSpace Weaver. Entitäten existieren innerhalb von Partitionen.

Apps

Eine SimSpace Weaver App ist eine von Ihnen geschriebene Software, die benutzerdefinierte Logik enthält, die jeden Simulationstrick ausführt. Der Zweck der meisten Apps besteht darin, Entitäten während der Ausführung der Simulation zu aktualisieren. Ihre Apps rufen APIs des SimSpace Weaver App-SDK auf, um Aktionen (wie Lesen und Aktualisieren) an Entitäten in Ihrer Simulation auszuführen.

Sie verpacken Ihre Apps und die benötigten Ressourcen (z. B. Bibliotheken) als ZIP-Dateien und laden sie hoch. SimSpace Weaver Eine App wird in einem Docker-Container auf einem Worker ausgeführt. SimSpace Weaver weist jeder App eine feste Anzahl von Ressourceneinheiten auf dem Worker zu.

SimSpace Weaver weist jeder App den Besitz einer (und nur einer) Partition zu. Eine App und ihre Partition befinden sich auf demselben Worker. Jede Partition hat nur einen App-Besitzer. Eine App kann Entitäten in ihrer Partition erstellen, lesen, aktualisieren und löschen. Eine App besitzt alle Entitäten in ihrer Partition.

Es gibt drei Arten von Apps: räumliche Apps, benutzerdefinierte Apps und Service-Apps. Sie unterscheiden sich je nach Anwendungsfällen und Lebenszyklen.

Note

In kann SimSpace Weaver sich der Begriff „App“ auf den Code für eine App oder eine laufende Instanz dieses Codes beziehen.

Räumliche Apps

Räumliche Apps aktualisieren den Status von Entitäten, die räumlich in Ihrer Simulation vorhanden sind. Sie könnten beispielsweise eine `Physics` App definieren, die für die Bewegung und Kollision von Entitäten bei jedem Tick verantwortlich ist, basierend auf ihrer Geschwindigkeit, Form und Größe. In diesem Fall werden mehrere Instanzen der `Physics` App parallel SimSpace Weaver ausgeführt, um die Größe der Arbeitslast zu bewältigen.

SimSpace Weaver verwaltet den Lebenszyklus von Apps mit räumlichen Daten. Sie geben eine Anordnung von räumlichen App-Partitionen in Ihrem Simulationsschema an. Wenn Sie Ihre Simulation starten, SimSpace Weaver wird für jede Spatial-App-Partition eine Spatial-App gestartet. Wenn Sie die Simulation beenden, werden Ihre Geo-Apps SimSpace Weaver heruntergefahren.

Andere Arten von Apps können Entitäten erstellen, aber nur räumliche Apps können Entitäten aktualisieren. Andere Arten von Apps müssen Entitäten, die sie erstellen, in eine räumliche Domäne übertragen. SimSpace Weaver verwendet die räumliche Position einer Entität, um die Entität in die Partition einer Geo-App zu verschieben. Dadurch wird das Eigentum an der Entität auf die Geo-App übertragen.

Benutzerdefinierte Apps

Sie verwenden benutzerdefinierte Apps, um mit Ihrer Simulation zu interagieren. Eine benutzerdefinierte App liest Entitätsdaten mithilfe von Abonnements. Eine benutzerdefinierte App kann Entitäten erstellen. Die App muss jedoch eine Entität in eine Geo-App übertragen, um die Entität in die Simulation aufzunehmen und zu aktualisieren. Sie können einer benutzerdefinierten App einen Netzwerkendpunkt SimSpace Weaver zuweisen. Simulationsclients können eine Verbindung zum Netzwerkendpunkt herstellen, um mit der Simulation zu interagieren. Sie definieren Ihre benutzerdefinierten Apps in Ihrem Simulationsschema, aber Sie sind dafür verantwortlich, sie zu starten und zu beenden (mithilfe von SimSpace Weaver API-Aufrufen). Nachdem Sie eine benutzerdefinierte App-Instanz auf einem Worker gestartet haben, SimSpace Weaver wird die Instanz nicht auf einen anderen Worker übertragen.

Service-Apps

Sie können eine Service-App verwenden, wenn für jeden Worker ein schreibgeschützter Prozess ausgeführt werden muss. Sie können beispielsweise eine Service-App verwenden, wenn Sie eine umfangreiche Simulation haben und einen Viewer-Client benötigen, der die Simulation durchläuft und dem Benutzer nur die sichtbaren Entitäten anzeigt. In diesem Fall kann eine einzelne

benutzerdefinierte App-Instanz nicht alle Entitäten in der Simulation verarbeiten. Sie können eine Service-App so konfigurieren, dass sie auf jedem Worker gestartet wird. Jede dieser Service-Apps kann dann die Entitäten für den ihr zugewiesenen Worker filtern und nur die relevanten Entitäten an die verbundenen Clients senden. Ihr Viewing Client kann sich dann mit verschiedenen Service-Apps verbinden, während er sich durch den Simulationsraum bewegt. Sie konfigurieren Service-Apps in Ihrem Simulationsschema. SimSpace Weaver startet und stoppt Ihre Service-Apps für Sie.

Zusammenfassung der App

In der folgenden Tabelle sind die Eigenschaften der verschiedenen Arten von SimSpace Weaver Apps zusammengefasst.

	Räumliche Apps	Benutzerdefinierte Apps	Service-Apps
Entitäten lesen	Ja	Ja	Ja
Entitäten aktualisieren	Ja	Nein	Nein
Entitäten erstellen	Ja	Ja*	Ja*
Lebenszyklus	Verwaltet (SimSpace Weaver kontrolliert es.)	Nicht verwaltet (Sie kontrollieren es.)	Verwaltet (SimSpace Weaver kontrolliert es.)
Methode starten	SimSpace Weaver startet eine App-Instanz für jede räumliche Partition, wie in Ihrem Schema angegeben.	Sie starten jede App-Instanz.	SimSpace Weaver startet eine oder mehrere App-Instanzen auf jedem Worker, wie in Ihrem Schema angegeben.
Clients können eine Verbindung herstellen	Nein	Ja	Ja

* Wenn eine benutzerdefinierte App oder Service-App eine Entität erstellt, muss die App das Eigentum an der Entität auf eine Geo-App übertragen, damit die Geo-App den Status der Entität aktualisieren kann.

Domains

Eine SimSpace Weaver Domain ist eine Sammlung von App-Instanzen, die denselben ausführbaren App-Code ausführen und dieselben Startoptionen und Befehle haben. Wir bezeichnen Domänen anhand der Arten von Apps, die sie enthalten: räumliche Domänen, benutzerdefinierte Domänen und Dienstdomänen. Sie konfigurieren Ihre Apps innerhalb von Domänen.

Abonnements und Replikation

Eine App erstellt ein Abonnement für eine räumliche Region, um Entitätsereignisse (z. B. Eingabe, Verlassen, Erstellen, Aktualisieren und Löschen) in dieser Region zu ermitteln. Eine App verarbeitet Entitätsereignisse aus einem Abonnement, bevor sie Daten für Entitäten in Partitionen liest, deren Eigentümer sie nicht ist.

Eine Partition kann auf demselben Worker wie die App existieren (dies wird als lokale Partition bezeichnet), aber eine andere App kann Eigentümer der Partition sein. Eine Partition kann auch auf einem anderen Worker existieren (dies wird als Remote-Partition bezeichnet). Wenn das Abonnement für eine Remote-Partition gilt, erstellt der Worker mithilfe eines Prozesses, der als Replikation bezeichnet wird, eine lokale Kopie der Remote-Partition. Der Worker liest dann die lokale Kopie (replizierte Remote-Partition). Wenn eine andere App auf dem Worker auf demselben Tick von dieser Partition lesen muss, liest der Worker dieselbe lokale Kopie.

Beispielhafte Anwendungsfälle für SimSpace Weaver

Sie können es SimSpace Weaver für agentenbasierte Modelle und diskrete Zeitschrittsimulationen mit einer räumlichen Komponente verwenden.

Erstellen Sie Simulationen mit großen Menschenmengen

Sie können SimSpace Weaver damit Menschenmassen in realen Umgebungen simulieren. SimSpace Weaver ermöglicht es Ihnen, Ihre Simulationen auf Millionen von dynamischen Objekten mit eigenem Verhalten zu skalieren.

Erstellen Sie Umgebungen im Stadtmaßstab

Wird verwendet SimSpace Weaver , um einen digitalen Zwilling einer ganzen Stadt zu erstellen. Erstellen Sie Simulationen für die Stadtplanung, die Planung der Verkehrsführung und die Planung der Reaktion auf Umweltgefahren. Sie können Ihre eigenen Geodatenquellen als Bausteine für Ihre Umgebungen verwenden.

Schaffen Sie immersive und interaktive Erlebnisse

Erstellen Sie Simulationserlebnisse, an denen mehrere Benutzer teilnehmen und interagieren können. Verwenden Sie beliebige Entwicklungstools wie Unreal Engine und Unity, um dreidimensionale (3D) virtuelle Welten zu erstellen. Passen Sie Ihr 3D-Erlebnis mit Ihren eigenen Inhalten und Verhaltensweisen an.

AWS SimSpace Weaver Ende des Supports

Nach reiflicher Überlegung haben wir beschlossen AWS SimSpace Weaver, den Support mit Wirkung zum 20. Mai 2026 einzustellen. AWS SimSpace Weaver wird ab dem 20. Mai 2025 keine neuen Kunden mehr annehmen. Als Bestandskunde mit einem Konto, das vor dem 20. Mai 2025 für den Service registriert wurde, können Sie die AWS SimSpace Weaver Funktionen weiterhin nutzen. Nach dem 20. Mai 2026 können Sie ihn nicht mehr nutzen AWS SimSpace Weaver.

[Weitere Informationen zur Umstellung auf containerisierte Simulationen AWS Batch zur Unterstützung bei der Ausführung finden Sie in diesem Blogbeitrag.](#)

Einrichtung für SimSpace Weaver

Um SimSpace Weaver für die erste Nutzung eingerichtet zu werden, müssen Sie Ihre AWS-Konto und Ihre lokale Umgebung einrichten. Wenn Sie diese Aufgaben abgeschlossen haben, sind Sie bereit für die [Tutorials „Erste Schritte“](#).

Einrichtungsaufgaben

1. [Richten Sie Ihre AWS-Konto zur Verwendung ein SimSpace Weaver.](#)
2. [Richten Sie Ihre lokale Umgebung ein für SimSpace Weaver.](#)

Richten Sie Ihre AWS-Konto zur Verwendung ein SimSpace Weaver

Führen Sie die folgenden Aufgaben aus, AWS-Konto um Ihre Nutzung einzurichten SimSpace Weaver.

Melde dich an für eine AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie [https://portal.aws.amazon.com/billing/die Anmeldung](https://portal.aws.amazon.com/billing/die-Anmeldung).
2. Folgen Sie den Online-Anweisungen.

Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscode auf der Telefontastatur.

Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontos erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter [Benutzerzugriff mit der Standardeinstellung konfigurieren](#).AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal](#).

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

Anweisungen hierzu finden Sie unter [Berechtigungssatz erstellen](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Eine genaue Anleitung finden Sie unter [Gruppen hinzufügen](#) im AWS IAM Identity Center Benutzerhandbuch.

Fügen Sie zu verwendende Berechtigungen hinzu SimSpace Weaver

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter [Eine Rolle für einen externen Identitätsanbieter \(Verbund\) erstellen](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter [Eine Rolle für einen IAM-Benutzer erstellen](#) im IAM-Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Example IAM-Richtlinie zur Gewährung von Nutzungsberechtigungen 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"
      }
    }
  }
]
```

Richten Sie Ihre lokale Umgebung ein für SimSpace Weaver

SimSpace Weaver Simulationen werden in Containern ausgeführt Amazon Linux 2 (AL2) Umgebungen. Sie benötigen eine AL2 Umgebung, in der Sie Ihre Apps kompilieren und mit dem SimSpace Weaver App-SDK verknüpfen können. Die standardmäßige lokale Entwicklungsumgebung ist ein AL2 Container in Docker. Wenn Sie sich dafür entscheiden, es nicht zu verwenden Docker, bieten wir alternative Anweisungen zum Ausführen einer AL2 Umgebung in Windows Subsystem for Linux (WSL). Sie können auch Ihre eigene Methode verwenden, um eine lokale AL2 Umgebung zu erstellen. Weitere Möglichkeiten zur AL2 lokalen Ausführung finden Sie in der [EC2 Amazon-Dokumentation](#).

Important

Docker on Microsoft Windows ist die Standard-Entwicklungsumgebung. Der Einfachheit halber schlagen wir andere Methoden zur Einrichtung Ihrer lokalen Entwicklungsumgebung vor, diese sind jedoch kein Standard und werden nicht unterstützt.

Themen

- [Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 \(AL2\) in Docker](#)
- [Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 \(AL2\) in Windows Subsystem for Linux \(WSL\)](#)

Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 (AL2) in Docker

Dieser Abschnitt enthält Anweisungen zum Einrichten Ihrer lokalen SimSpace Weaver Distribution-Zip-Datei mit einer AL2 Umgebung in Docker. Für Anweisungen zur Einrichtung mit AL2 in Windows Subsystem for Linux (WSL), finden Sie unter [Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 \(AL2\) in Windows Subsystem for Linux \(WSL\)](#).

Voraussetzungen

- Microsoft Windows 10 oder höher oder ein kompatibles Linux-System
- [Microsoft Visual Studio 2019](#) oder später, mit dem [Desktop development with C++](#) installierter Workload
- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)
- [Python 3.9](#)

Um die SimSpace Weaver Distribution-Zip-Datei mit AL2 in einzurichten Docker

1. Wenn Sie Ihre AWS Anmeldeinformationen für noch nicht konfiguriert haben AWS CLI, folgen Sie diesen Anweisungen: [Konfiguration der AWS-CLI](#).
2. [Laden Sie das verteilbare SimSpace Weaver App-SDK-Paket](#) herunter. Sie enthält Folgendes:
 - Binärdateien und Bibliotheken für die App-Entwicklung SimSpace Weaver
 - Hilfsskripte, die Teile des Entwicklungsworkflows automatisieren
 - Beispielanwendungen, die SimSpace Weaver Konzepte demonstrieren
3. Entpacken Sie die Datei in eine Datei *sdk-folder* Ihrer Wahl.
4. Rufen Sie das *sdk-folder* auf.
5. Geben Sie den folgenden Befehl ein, um die erforderlichen Python-Pakete zu installieren:

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

6. Geben Sie den folgenden Befehl ein, um die SimSpace Weaver Distribution mit einem Docker-Image einzurichten.

```
python setup.py
```

Der Befehl hat folgende Auswirkungen:

- Erstellt ein AL2 Docker-Image, auf dem alle Anforderungen für SimSpace Weaver Bauprojekte installiert sind.
- Erstellt die CloudFormation Ressourcen, die zum Starten einer Simulation erforderlich sind.
 - Die Beispielvorgabe für einen CloudFormation Stack finden Sie unter *sdk-folder/* `PackagingTools/sample-stack-template.yaml`
- Konfiguriert die bereitgestellten Beispielprojekte mit den richtigen Pfaden für Ihr lokales System.

Fehlerbehebung

- Docker scheint festzustecken
 - Wenn die Konsolenausgabe nach dem Aufrufen von Docker-Befehlen hängen zu bleiben scheint, versuchen Sie, die Docker-Engine neu zu starten. Wenn das nicht funktioniert, starten Sie Ihren Computer neu.

Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL)

Dieser Abschnitt enthält Anweisungen zum Einrichten Ihrer SimSpace Weaver Distribution-Zip-Datei mit einer AL2 Umgebung in Windows Subsystem for Linux (WSL). Für Anweisungen zum Einrichten AL2 in Docker, finden Sie unter [Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 \(AL2\) in Docker](#).

Important

In diesem Abschnitt wird eine Lösung beschrieben, die eine Version verwendet AL2 , die nicht Amazon gehört, nicht von Amazon entwickelt oder unterstützt wird. Diese Lösung wird nur zu

Ihrer Bequemlichkeit bereitgestellt, falls Sie sie nicht verwenden möchten Docker. Amazon und AWS übernehmen keine Haftung, wenn Sie sich für diese Lösung entscheiden.

Voraussetzungen

- [Hyper-V on Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- AL2 Open-Source-Distribution von Drittanbietern für WSL ([Version 2.0.20200722.0-update.2 herunterladen](#)) ([siehe Anweisungen](#))

Important

Unsere WSL Anweisungen verwenden die [Version 2.0.20200722.0-update.2](#) der Distribution für AL2 WSL. Wenn Sie eine andere Version verwenden, können Fehler auftreten.

Um die SimSpace Weaver Distribution-Postleitzahl mit in einzurichten AL2 WSL

1. Starten Sie an einer Windows-Befehlszeile Ihre AL2 Umgebung in WSL.

```
ws1 -d Amazon2
```

Important

Während Sie in WSL, schließen Sie die `--al2` Option ein, wenn Sie eines der `quick-start.py` Python-Hilfsskripte ausführen, die sich unter `befindensdkey-folder/Samples/sample-name/tools/cloud/quick-start.py`.

2. Aktualisieren Sie an einer Linux-Shell-Eingabeaufforderung Ihren Yum-Paketmanager.

```
yum update -y
```

⚠ Important

Wenn bei diesem Schritt eine Zeitüberschreitung eintritt, müssen Sie möglicherweise zu wechseln WSL1 und wiederholen Sie diese Verfahren. Beenden Sie Ihre WSL AL2 Sitzung und geben Sie an der Windows-Eingabeaufforderung Folgendes ein:

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

3. Installieren Sie das Tool zum Entpacken.

```
yum install -y unzip
```

4. Entfernen Sie alle yum installierten AWS CLI Dateien. Probieren Sie die beiden folgenden Befehle aus, wenn Sie sich nicht sicher sind, ob ein yum installiert ist AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Erstellen Sie ein temporäres Verzeichnis und gehen Sie dorthin.

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

6. Downloaden und installieren Sie das AWS CLI:

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

7. Sie können das temporäre Verzeichnis entfernen.

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

8. Starten Sie die Shell-Sitzung neu, um den Pfad in der Umgebung zu aktualisieren.

```
exec
```

9. Konfigurieren Sie Ihre AWS Anmeldeinformationen für die AWS CLI in Ihrer AL2 Umgebung. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI](#). Falls Sie dies verwenden AWS IAM Identity Center, finden Sie weitere Informationen unter [Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center](#) im AWS Command Line Interface Benutzerhandbuch.

```
aws configure
```

10. Installieren Sie Git.

```
yum install -y git
```

11. Installieren wget.

```
yum install -y wget
```

12. Erstellen Sie einen Ordner für das SimSpace Weaver App-SDK.

```
mkdir sdk-folder
```

13. Gehen Sie zu Ihrem SDK-Ordner.

```
cd sdk-folder
```

14. Laden Sie das verteilbare SimSpace Weaver App-SDK-Paket herunter. Sie enthält Folgendes:

- Binärdateien und Bibliotheken für die App-Entwicklung SimSpace Weaver
- Hilfsskripte, die Teile des Entwicklungsworkflows automatisieren
- Beispielanwendungen, die SimSpace Weaver Konzepte demonstrieren

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

15. Entpacken Sie die Datei.

```
unzip *.zip
```

16. Führen Sie das WSL-Setup-Skript aus.

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

17. Geben Sie den folgenden Befehl ein, um die erforderlichen Python-Pakete zu installieren:

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

18. Führen Sie das Installationsskript für die SimSpace Weaver Distribution ZIP aus:

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

Der Befehl hat folgende Auswirkungen:

- Erzeugt die CloudFormation Ressourcen, die zum Starten einer Simulation erforderlich sind.
 - Die Beispielvorlage für einen CloudFormation Stack finden Sie unter *sdk-folder/* `PackagingTools/sample-stack-template.yaml`
- Konfiguriert die bereitgestellten Beispielprojekte mit den richtigen Pfaden für Ihr lokales System.

Note

Sie müssen dies nur einmal für Ihre AL2 Umgebung in WSL tun.

Verwendung von lizenzierter Software mit AWS SimSpace Weaver

AWS SimSpace Weaver ermöglicht es Ihnen, Simulationen mit der Simulationsengine und den Inhalten Ihrer Wahl zu erstellen. Im Zusammenhang mit Ihrer Nutzung von sind Sie dafür verantwortlich SimSpace Weaver, die Lizenzbedingungen aller Software oder Inhalte, die Sie in Ihren Simulationen verwenden, zu beschaffen, zu pflegen und einzuhalten. Stellen Sie sicher, dass Ihr Lizenzvertrag Ihnen die Bereitstellung Ihrer Software und Inhalte in einer virtuellen, gehosteten Umgebung ermöglicht.

Erste Schritte mit SimSpace Weaver

In diesem Abschnitt finden Sie Tutorials, die Ihnen den Einstieg erleichtern SimSpace Weaver. Diese Tutorials führen Sie in den allgemeinen Arbeitsablauf für Gebäudesimulationen mit ein SimSpace Weaver. Diese Tutorials zeigen, wie Sie Simulationen in erstellen, bereitstellen und ausführen SimSpace Weaver. Wir empfehlen, mit dem Schnellstart-Tutorial zu beginnen, um eine Simulation in wenigen Minuten zum Laufen zu bringen. Sehen Sie sich danach die anderen Tutorials an, um mehr zu erfahren.

In diesen Tutorials wird eine Beispielanwendung (PathfindingSample) verwendet, die in der .zip-Datei des SimSpace Weaver App-SDK enthalten ist, die Sie während der [Einrichtung heruntergeladen haben](#). Die Beispielanwendung demonstriert die Konzepte, die allen SimSpace Weaver Simulationen gemeinsam sind, einschließlich räumlicher Partitionierung, partitionsübergreifender Entitätsübergabe, Apps und Abonnements.

In den Tutorials erstellen Sie eine Simulation mit vier räumlichen Partitionen. Eine separate Instanz der PathfindingSample Spatial-App verwaltet jede einzelne Partition. Die räumlichen Apps erstellen Entitäten in ihren eigenen Partitionen. Die Entitäten bewegen sich an eine bestimmte Position in der Simulationswelt und weichen dabei Hindernissen aus. Sie können eine separate Client-Anwendung (im SimSpace Weaver App-SDK enthalten) verwenden, um die Simulation anzusehen.

Themen

- [Schnellstart-Tutorial für SimSpace Weaver](#)
- [Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung](#)

Schnellstart-Tutorial für SimSpace Weaver

Dieses Tutorial führt Sie durch den Prozess zum Erstellen und Ausführen einer Simulation SimSpace Weaver in wenigen Minuten. Wir empfehlen, dass Sie mit diesem Tutorial beginnen und anschließend das [ausführliche Tutorial](#) durchgehen.

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die Schritte unter abgeschlossen haben. [Einrichtung für SimSpace Weaver](#)

Note

Die hier verwendeten Skripte dienen der besseren Übersicht und sind NICHT erforderlich. In der [ausführlichen Anleitung erfahren](#) Sie, wie diese Schritte manuell ausgeführt werden können.

Schritt 1: Protokollierung aktivieren (optional)

Um die Protokollierung zu aktivieren

1. Navigieren Sie zu:

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

2. Öffnen Sie die Schemadatei in einem Texteditor:

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

3. Suchen Sie den `simulation_properties`: Abschnitt am Anfang der Datei:

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

4. Fügen Sie nach der Zeile die folgenden 2 Zeilen ein `simulation_properties::`

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

5. Vergewissern Sie sich, dass Ihr `simulation_properties`: Abschnitt dem folgenden entspricht:

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

6. Speichern Sie die Datei und beenden Sie den Text-Editor.

Schritt 2: Schnellstart mit dem Konsolen-Client (Option 1)

Navigieren Sie zu:

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

Führen Sie einen der folgenden Befehle aus:

- Docker: `python quick-start.py --consoleclient`
- WSL: `python quick-start.py --consoleclient --al2`

Standardmäßig wird dadurch eine Simulation mit einer einzelnen Partition auf einem einzelnen Worker gestartet. Andere Konfigurationen können gestartet werden, indem das `--schema {file name}.yaml` aus dem `/Samples/PathfindingSample/tools/` Ordner übergeben wird.

Note

Eine [Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung](#) ausführliche Erläuterung der Funktionsweise dieses Skripts finden Sie unter.

Schritt 2: Schnellstart mit dem Unreal Engine-Client (Option 2)

Siehe [Den Unreal Engine View Client starten](#).

Stoppen und löschen Sie Ihre Simulation

Navigieren Sie zu:

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

Finden Sie die Namen Ihrer Simulationen:

```
aws simspaceweaver list-simulations
```

Stoppen und löschen Sie die Simulation

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

Fehlerbehebung

- `FileNotFoundError: cmake`

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

- Lösung: Das Skript kann den Befehl `cmake` nicht finden. Bitte stellen Sie sicher, dass Sie die empfohlene CMake Mindestversion installiert haben und dass sie mit dem `cmake` Befehl im `PATH` aufgerufen werden kann. Verwenden Sie den Befehl `cmake --version` zur Überprüfung.
- `ImportError`: Das Laden der DLL ist beim Import von `libweaver_app_sdk_python_v1` fehlgeschlagen: Das angegebene Modul konnte nicht gefunden werden.
 - Lösung: Dieser Fehler tritt auf, wenn Python 3.9 nicht zum Starten des Weaver Python SDK verwendet wird. Bitte stellen Sie sicher, dass die mit dem Befehl „Python“ verknüpfte Python-Version Python 3.9 ist. Sie können dies überprüfen, indem Sie den `python --version` Befehl ausführen.
- Das Schnellstart-Skript scheint nach dem Start von Docker Build hängen zu bleiben.
 - Lösung: Manchmal benötigt Docker ein paar Minuten, um sich aufzuwärmen. Wenn dieses Problem länger als ~5 Minuten andauert, starten Sie Docker oder Ihr System neu.
- `target_compile_features` Keine bekannten Funktionen für den CXX-Compiler „GNU“:
 - Lösung: Leeren Sie Ihren Docker-Cache, löschen Sie das Weaverappbuilder-Docker-Image, löschen Sie die Build-Artefakte Ihres Projekts und führen Sie es erneut aus. `setup.py` Dadurch sollte Ihre Docker-Umgebung zurückgesetzt und der Fehler behoben werden.

Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung

Das [Schnellstart-Tutorial](#) befasste sich mit dem Erstellen, Starten, Stoppen und Löschen einer Beispielsimulation mit `quick-start.py` und `stop-and-delete.py`. In diesem Tutorial wird detailliert beschrieben, wie diese Skripten funktionieren und welche zusätzlichen Parameter diese Skripte verwenden können, um die Flexibilität für benutzerdefinierte Weaver-Simulationen zu maximieren.

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die Schritte unter abgeschlossen haben. [Einrichtung für SimSpace Weaver](#)

Schritt 1: Protokollierung aktivieren (optional)

Um die Protokollierung zu aktivieren

1. Navigieren Sie zu:

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

2. Öffnen Sie die Schemadatei in einem Texteditor:

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

3. Suchen Sie den `simulation_properties`: Abschnitt am Anfang der Datei:

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

4. Fügen Sie nach der Zeile die folgenden 2 Zeilen ein `simulation_properties::`

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

5. Vergewissern Sie sich, dass Ihr `simulation_properties`: Abschnitt dem folgenden entspricht:

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

6. Speichern Sie die Datei und beenden Sie den Text-Editor.

Schritt 2: Starten Sie Ihre Simulation

Wie im [Schnellstart-Tutorial](#) gezeigt, sind die grundlegendsten Schritte zum Starten einer Beispielsimulation:

1. Navigieren Sie zu:

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

2. Führen Sie einen der folgenden Befehle aus:

- Docker: `python quick-start.py`
- WSL: `python quick-start.py --a12`

Dieses Skript automatisiert allgemeine Terminalbefehle, die alle manuell mit dem ausgeführt werden können. AWS CLI Diese Schritte sind:

1. Laden Sie das Weaver-Schema auf S3 hoch.

- SimSpace Weaver verwendet ein Schema, um Ihre Simulation zu konfigurieren. Das Schema ist eine Klartextdatei im YAML-Format. Weitere Informationen finden Sie unter [Konfiguration Ihrer Simulation](#).

2. Erstellen Sie einen benutzerdefinierten Container und laden Sie ihn hoch (optional).

- Wenn Ihr Schema einen benutzerdefinierten Container definiert, erstellt das Schnellstart-Skript das Docker-Image und lädt es auf Amazon ECR hoch. Weitere Informationen finden Sie unter [Maßgeschneiderte Behälter](#). Ein Beispiel für diese Funktion finden Sie im `PythonBubblesSample` Schema.

3. Erstellen Sie das Projekt.

- `quick-start.py` ruft die in definierte `build_project` Funktion auf `build.py`. Dieser Schritt variiert je nach Projekt. Für das `PathfindingSample`, CMake wird verwendet. Der CMake und Docker-Befehl, für den Sie unter finden, finden Sie unter `build.py`.

4. Laden Sie die Build-Artefakte auf S3 hoch.

- Sie können Ihre S3-Buckets überprüfen, um sicherzustellen, dass alle Uploads erfolgreich waren. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unter [Erstellen, Konfigurieren und Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.
- Die Zips und der S3-Bucket der Beispielanwendung verwenden das folgende Namensformat:
 - `weaver-sample-bucket-account-number-region`
 - Räumliche App: `ProjectNameSpatial.zip`
 - (benutzerdefinierte) App anzeigen: `ProjectNameView.zip`

5. Starten Sie die Simulation.

- Dies ist eine Zusammenfassung des `aws simspaceweaver start-simulation` AWS CLI Anrufs. Weitere Informationen finden Sie in der [AWS CLI Befehlsreferenz für SimSpace Weaver](#).
- Das Skript wird solange wiederholt, bis der Simulationsstatus entweder `STARTED` oder `laudetFAILED`. Es kann einige Minuten dauern, bis eine Simulation gestartet wird.

6. Holen Sie sich die Simulationsdetails.

- Die `DescribeSimulation` API bietet Details zu Ihrer Simulation, einschließlich ihres Status. Eine Simulation kann sich in einem der folgenden Zustände befinden:

Status des Simulationslebenszyklus

1. **STARTING**— Ausgangszustand nach dem Anruf `StartSimulation`
2. **STARTED**— Alle räumlichen Apps sind gestartet und funktionieren einwandfrei
3. **STOPPING**— Ausgangszustand nach dem Anruf `StopSimulation`
4. **STOPPED**— Alle Rechenressourcen sind gestoppt
5. **DELETING**— Ausgangszustand nach dem Anruf `DeleteSimulation`
6. **DELETED**— Alle der Simulation zugewiesenen Ressourcen werden gelöscht
7. **FAILED**— Die Simulation hatte einen kritischen Fehler/Ausfall und wurde gestoppt
8. **SNAPSHOT_IN_PROGRESS**— Ein [Snapshot](#) ist in Bearbeitung

Um Ihre Simulationsdetails zu erhalten

1. Rufen Sie die `ListSimulations`-API auf.

```
aws simspaceweaver list-simulations
```

Das Skript sollte Details zu jeder Ihrer Simulationen anzeigen, ähnlich den folgenden:

```
{
  "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. Rufen Sie `DescribeSimulation` an, um Ihre Simulationsdetails zu erhalten. *simulation-name* Ersetzen Sie es durch das Name Ergebnis Ihrer Simulation aus der Ausgabe des vorherigen Schritts.

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

Das Skript sollte mehr Details zu der von Ihnen angegebenen Simulation anzeigen, ähnlich der folgenden:

```
{
  "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. Starten Sie benutzerdefinierte Apps.

- SimSpace Weaver verwaltet nicht den Lebenszyklus von benutzerdefinierten Apps. Sie müssen Ihre benutzerdefinierten Apps starten. Es empfiehlt sich, Ihre benutzerdefinierten Apps zu starten, bevor Sie Ihre Simulationsuhr starten. Sie können benutzerdefinierte Apps jedoch auch starten, nachdem Sie die Uhr gestartet haben.

Sie können die `StartApp` API aufrufen, um Ihre benutzerdefinierten Apps zu starten.

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

Durch den `StartApp` API-Aufruf wird eine neue Instanz der benutzerdefinierten App unter dem von Ihnen angegebenen Namen erstellt und gestartet. Wenn Sie den Namen einer App angeben, die bereits existiert, erhalten Sie eine Fehlermeldung. Wenn Sie eine bestimmte App (Instanz) neu starten möchten, müssen Sie diese App zuerst beenden und löschen.

Note

Der Status Ihrer Simulation muss vorliegen, `STARTED` bevor Sie benutzerdefinierte Apps starten können.

Die Beispielanwendung bietet die `ViewApp` benutzerdefinierte App zum Anzeigen Ihrer Simulation. Diese App stellt Ihnen eine statische IP-Adresse und Portnummer zur Verfügung, um die Simulationsclients zu verbinden (Sie werden dies in einem späteren Schritt in diesem Tutorial tun). Sie können sich a domain als eine Klasse von Apps vorstellen, die denselben ausführbaren Code und dieselben Startoptionen haben. Das `app name` identifiziert die Instanz der App. Weitere Informationen zu SimSpace Weaver Konzepten finden Sie unter [Schlüsselkonzepte für SimSpace Weaver](#).

Sie können die `DescribeApp` API verwenden, um den Status einer benutzerdefinierten App zu überprüfen, nachdem Sie sie gestartet haben.

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

Um die `View-App` in diesem Tutorial zu starten

1. Rufen Sie an `StartApp` für `ViewApp`.

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

2. Rufen Sie `DescribeApp` an, um den Status Ihrer benutzerdefinierten App zu überprüfen.

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

Sobald der Status Ihrer benutzerdefinierten App (Instanz) lautet `STARTED`, enthält die Ausgabe von `DescribeApp` die IP-Adresse und die Portnummer für diese benutzerdefinierte App (Instanz). In der folgenden Beispielausgabe ist die IP-Adresse der Wert von `Address` und die Portnummer der Wert von `Actual` im `EndpointInfo` Block.

```
{
  "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

Der Wert von `Declared` ist die Portnummer, an die Ihr App-Code gebunden werden soll. Der Wert von `Actual` ist die Portnummer, über SimSpace Weaver die Clients eine Verbindung zu Ihrer App herstellen können. SimSpace Weaver ordnet den `Declared` Port dem Port zu. `Actual`

Note

Sie können das unter beschriebene Verfahren verwenden [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#), um die IP-Adresse und die Portnummer jeder gestarteten benutzerdefinierten App abzurufen.

8. Starten Sie die Uhr.

- Wenn Sie Ihre Simulation zum ersten Mal erstellen, hat sie eine Uhr, aber die Uhr läuft nicht. Wenn Ihre Uhr nicht läuft, aktualisiert Ihre Simulation ihren Status nicht. Nachdem Sie die Uhr gestartet haben, beginnt sie, Ticks an Ihre Apps zu senden. Bei jedem Tick durchgehen Ihre Geo-Apps die Entitäten, deren Eigentümer sie sind, und übertragen die Ergebnisse an SimSpace Weaver

 Note

Es kann 30-60 Sekunden dauern, bis die Uhr gestartet wird.

Rufen Sie die StartClock-API auf.

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

 Note

Die StartClock API verwendet Ihre *simulation-name*, die Sie mithilfe der ListSimulations API finden können:

```
aws simspaceweaver list-simulations
```

Schnellstart-Parameter

- -h, --help
 - Führen Sie diese Parameter auf.
- --sauber
 - Löscht den Inhalt des Build-Verzeichnisses vor dem Erstellen.
- --al2
 - Baut direkt auf dem nativen Computer statt auf Docker. Verwenden Sie dies nur, wenn Sie in einer Amazon Linux 2-Umgebung wie WSL ausgeführt werden.
- --nur hochladen
 - Laden Sie nur das Schema und die App-Zip-Dateien auf Amazon S3 hoch, starten Sie die Simulation nicht.

- `--kein Build`
 - Überspringe den Neuaufbau des Projekts.
- `-- kein Behälter`
 - Überspringt die Neuerstellung des Simulationscontainers, der im Schema aufgeführt ist.
- `--consoleclient`
 - Der in `config.py` aufgeführte Konsolen-Client wird automatisch erstellt und eine Verbindung hergestellt.
- `--schema SCHEMA`
 - Welches Schema dieser Aufruf verwenden wird. Standardmäßig ist der Wert 'SCHEMA' in `config.py`.
- `--Name NAME`
 - Welchen Namen die Simulation haben wird. Standardmäßig ist der Wert von 'Project_Name'-Date-Time in `config.py` voreingestellt.

Schritt 3: Überprüfen Sie die Protokolle (optional)

SimSpace Weaver schreibt Simulationsverwaltungsnachrichten und die Konsolenausgabe Ihrer Apps in Amazon CloudWatch Logs. Weitere Informationen zur Arbeit mit Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollströmen](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Jede Simulation, die Sie erstellen, hat ihre eigene Protokollgruppe in CloudWatch Logs. Der Name der Protokollgruppe ist im Simulationsschema angegeben. Im folgenden Schemaausschnitt ist der Wert von `log_destination_service` `logs`. Das bedeutet, dass der Wert von der Name einer Protokollgruppe `log_destination_resource_name` ist. In diesem Fall ist die Protokollgruppe `MySimulationLogs`.

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

Sie können die DescribeSimulation API auch verwenden, um den Namen der Protokollgruppe für die Simulation zu ermitteln, nachdem Sie sie gestartet haben.

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

Das folgende Beispiel zeigt den Teil der AusgabeDescribeSimulation, der die Protokollkonfiguration beschreibt. Der Name der Protokollgruppe wird am Ende von `angezeigtLogGroupArn`.

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

Jede Simulationsprotokollgruppe enthält mehrere Protokollströme:

- Management-Log-Stream — vom SimSpace Weaver Dienst erzeugte Simulationsverwaltungsnachrichten.

```
/sim/management
```

- Fehlerprotokollstream — vom SimSpace Weaver Dienst erzeugte Fehlermeldungen. Dieser Protokollstream ist nur vorhanden, wenn Fehler vorliegen. SimSpace Weaver speichert Fehler, die von Ihren Apps geschrieben wurden, in ihren eigenen App-Protokollstreams (siehe die folgenden Protokollstreams).

```
/sim/errors
```

- Spatial App Log Streams (1 für jede Spatial App auf jedem Worker) — Konsolenausgabe, die von Spatial Apps erzeugt wird. Jede Spatial-App schreibt in ihren eigenen Log-Stream. *spatial-app-id* Das sind alle Zeichen nach dem abschließenden Schrägstrich am Ende von. *worker-id*

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

- Benutzerdefinierte App-Protokollstreams (1 für jede benutzerdefinierte App-Instanz) — Konsolenausgabe, die von benutzerdefinierten Apps erzeugt wird. Jede benutzerdefinierte Anwendungsinstanz schreibt in ihren eigenen Protokollstream.

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

- Protokollstreams von Service-Apps (1 für jede Service-App-Instanz) — Konsolenausgabe, die von Service-Apps erzeugt wird. Jede Service-App schreibt in ihren eigenen Protokollstream. *service-app-id* Das sind alle Zeichen nach dem abschließenden Schrägstrich am Ende von. *service-app-name*

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

Note

Die Beispielanwendung hat keine Service-Apps.

Schritt 4: Sehen Sie sich Ihre Simulation an

Das SimSpace Weaver App-SDK bietet verschiedene Optionen zum Anzeigen der Beispielanwendung. Sie können den Beispiel-Konsolenclient verwenden, wenn Sie keine lokale Unterstützung für die Unreal Engine-Entwicklung haben. In den Anweisungen für den Unreal Engine-Client wird davon ausgegangen, dass Sie Windows verwenden.

Der Konsolenclient zeigt eine Liste von Entitätsereignissen an, sobald sie auftreten. Der Client erhält die Informationen zum Entitätsereignis von `ViewApp`. Wenn Ihr Konsolenclient die Liste der Ereignisse anzeigt, bestätigt er die Netzwerkkonnektivität anhand der Aktivität `ViewApp` und in Ihrer Simulation.

Die `PathfindingSample` Simulation erzeugt stationäre und sich bewegende Objekte auf einer zweidimensionalen Ebene. Die beweglichen Objekte bewegen sich um die stationären Objekte herum. Der Unreal Engine-Client bietet eine Visualisierung der Entitätsereignisse.

Konsolen-Client

Der Konsolen-Client kann beim Starten eines Beispiels automatisch erstellt und verbunden werden, `quick-start.py` wenn Sie die `--consoleclient` Option angeben. Gehen Sie wie folgt vor, um den Konsolen-Client zu erstellen und `quick-start.py` eine Verbindung herzustellen, nachdem er bereits aufgerufen wurde:

Navigieren Sie zu:

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

Führen Sie das Skript aus, um den Client zu erstellen und eine Verbindung herzustellen:

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

Das Skript wird Folgendes tun:

1. Erstellen Sie den Konsolen-Client mit CMake.
2. Startet die erstellte ausführbare Datei mit der angegebenen IP-Adresse und Portnummer.

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

Unreal Engine-Client

Siehe [Den Unreal Engine View Client starten](#).

Schritt 5: Stoppen und löschen Sie Ihre Simulation

Navigiere zu:

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

Finden Sie die Namen Ihrer Simulationen:

```
aws simspaceweaver list-simulations
```

Stoppen und löschen Sie die Simulation:

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

Das Skript `stop-and-delete.py` wird Folgendes tun:

1. Rufen Sie den AWS CLI Befehl auf, um eine Simulation zu beenden.
 - `aws simspaceweaver stop-simulation`
 - Weitere Informationen finden Sie unter [AWS CLI Befehlsreferenz](#) für SimSpace Weaver.
2. Rufen Sie den AWS CLI Befehl auf, um eine Simulation zu löschen.
 - `aws simspaceweaver delete-simulation`
 - Weitere Informationen finden Sie unter [AWS CLI Befehlsreferenz](#) für SimSpace Weaver.

stop-and-delete Parameter

- `-h, --help`
 - Führen Sie diese Parameter auf.
- `--Simulation SIMULATION`
 - Der Name der Simulation für stop-and-delete
- `--stop`
 - Stoppen Sie einfach die Simulation. Löscht es nicht.
- `--löschen`
 - Löscht nur eine Simulation. Funktioniert nur, wenn die Simulation entweder STOPPED oder ist FAILED.

Fehlerbehebung

Siehe [Fehlerbehebung](#) im Schnellstart-Tutorial.

Arbeiten mit SimSpace Weaver

Dieses Kapitel enthält Informationen und Anleitungen, die Ihnen beim Erstellen Ihrer eigenen Anwendungen helfen sollen SimSpace Weaver.

Themen

- [Konfiguration Ihrer Simulation](#)
- [Maximale Dauer einer Simulation](#)
- [Apps entwickeln](#)
- [Entwicklung von Kundenanwendungen](#)
- [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#)
- [Den Unreal Engine View Client starten](#)
- [Lokale Entwicklung in SimSpace Weaver](#)
- [AWS SimSpace Weaver App-SDK](#)
- [AWS SimSpace Weaver Demo-Framework](#)
- [Mit Servicekontingenten arbeiten](#)
- [Simulationen debuggen](#)
- [Maßgeschneiderte Behälter](#)
- [Verwenden von Python](#)
- [Support für andere Motoren](#)
- [Verwendung von lizenzierter Software mit AWS SimSpace Weaver](#)
- [Verwaltung Ihrer Ressourcen mit AWS CloudFormation](#)
- [Snapshots](#)
- [Messaging](#)

Konfiguration Ihrer Simulation

Ein Simulationsschema (oder Schema) ist ein YAML-formatierte Textdatei, die die Konfiguration für eine Simulation spezifiziert. Sie können dasselbe Schema verwenden, um mehrere Simulationen zu starten. Die Schemadatei befindet sich im Projektordner für Ihre Simulation. Sie können einen beliebigen Texteditor verwenden, um die Datei zu bearbeiten. SimSpace Weaver liest Ihr Schema

nur, wenn die Simulation gestartet wird. Alle Änderungen, die Sie an einer Schemadatei vornehmen, wirken sich nur auf neue Simulationen aus, die Sie nach den Änderungen starten.

Um Ihre Simulation zu konfigurieren, bearbeiten Sie Ihre Simulationsschemadatei (verwenden Sie das entsprechende Pfadtrennzeichen für Ihr Betriebssystem):

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

Sie laden das Simulationsschema hoch, wenn Sie eine neue Simulation erstellen. Das Schnellstart-Hilfsskript für Ihr Projekt lädt das Schema als Teil des Prozesses zur Erstellung Ihrer Simulation hoch:

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

Weitere Informationen zum Ausführen des Schnellstart-Skripts finden Sie [Ausführliches Tutorial](#) im [Erste Schritte](#) Kapitel dieses Handbuchs.

Konfigurationsparameter für die Simulation

Das Simulationsschema enthält Bootstrapping-Informationen, darunter:

- Simulationseigenschaften — [SDK-Version und Rechenkonfiguration \(Typ und Anzahl der Worker\)](#)
- Uhren — Taktrate und Toleranzen
- Strategien zur räumlichen Aufteilung — räumliche Topologie (z. B. ein Raster), Grenzen und Platzierungsgruppen (räumliche Unterteilung nach Arbeitskräften)
- Domänen und ihre Apps — App-Bucket, Pfad und Startbefehl (e)

SimSpace Weaver verwendet Ihre Schemakonfiguration, um räumliche Partitionen zu konfigurieren und anzuordnen, Apps zu starten und die Simulation mit der angegebenen Tick-Rate voranzutreiben.

Note

Das Skript zum Erstellen eines Projekts im SimSpace Weaver App-SDK generiert automatisch ein Simulationsschema für Sie, das auf der Beispielanwendung basiert.

In den folgenden Themen werden die Parameter im Simulationsschema beschrieben. Eine vollständige Beschreibung des Simulationsschemas finden Sie unter [SimSpace Weaver Referenz zum Simulationsschema](#).

Themen

- [SDK-Version](#)
- [Eigenschaften der Simulation](#)
- [Worker](#)
- [Uhr](#)
- [Partitionierungsstrategien](#)
- [Domains](#)

SDK-Version

Das `sdk_version` Feld gibt die Version an SimSpace Weaver , für die das Schema formatiert ist. Zulässige Werte: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

Der Wert von `sdk_version` beinhaltet nur die Hauptversionsnummer und die erste Nebenversionsnummer. Der Wert 1.12 gibt beispielsweise alle Versionen an 1.12.x, z. B. 1.12.01.12.1, und 1.12.2.

Eigenschaften der Simulation

Der `simulation_properties` Abschnitt Ihres Schemas spezifiziert die Protokollierungskonfiguration und einen Datentyp für das Indexfeld (normalerweise die räumliche Position) von Entitäten.

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

Der Wert von `log_destination_service` bestimmt die Interpretation des Werts von `log_destination_resource_name`. Derzeit wird als einziger Wert unterstützt `logs`. Das bedeutet, dass der `log_destination_resource_name` Wert von der Name einer Protokollgruppe in Amazon CloudWatch Logs ist.

Note

Die Protokollierung ist optional. Wenn Sie die Eigenschaften des Protokollziels nicht konfigurieren, erzeugt Ihre Simulation keine Protokolle.

Nur die `default_entity_index_key_type`-Eigenschaft ist erforderlich. Der einzige gültige Wert ist `Vector3<f32>`.

Worker

`workers`In diesem Abschnitt werden die Art und Anzahl der Arbeiter angegeben, die Sie für Ihre Simulation verwenden möchten. SimSpace Weaver verwendet eigene Worker-Typen, die EC2 Amazon-Instance-Typen zugeordnet sind.

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

Aktivierung von Simulationen mit mehreren Mitarbeitern

Sie können eine Simulation erstellen, die mehr als einen Worker verwendet. Standardmäßig verwenden Simulationen 1 Worker. Sie müssen Ihr Simulationsschema ändern, bevor Sie die Simulation starten.

Note

Sie können eine Simulation, die bereits gestartet wurde, nicht ändern. Wenn Sie Multi-Worker für eine laufende Simulation aktivieren möchten, müssen Sie die Simulation zuerst beenden und löschen.

Um mehr als einen Worker zu verwenden, setzen Sie die `desired` Anzahl der Recheninstanzen auf einen Wert größer als 1. Es gibt eine maximale Anzahl von Apps für jeden Worker. Weitere Informationen finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#). SimSpace Weaver verwendet nur dann mehr als einen Worker, wenn die Anzahl der Apps auf einem Worker diesen

Grenzwert überschreitet. SimSpace Weaver kann eine App auf jedem der verfügbaren Mitarbeiter platzieren. Die Platzierung einer App bei einem bestimmten Mitarbeiter ist nicht garantiert.

Der folgende Schemaausschnitt zeigt eine Konfiguration für eine Simulation, die 2 Worker anfordert. SimSpace Weaver versucht, den zweiten Worker zuzuweisen, wenn die Anzahl der Apps die maximale Anzahl von Apps für einen Worker überschreitet.

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

Uhr

Der `clock` Abschnitt spezifiziert die Eigenschaften der Simulationsuhr. Derzeit können Sie nur die Tick-Rate konfigurieren (die Anzahl der Ticks pro Sekunde, die die Uhr an Apps sendet). Die Tickrate ist eine maximale Rate. Die effektive Tickrate könnte niedriger sein, da alle Operationen (wie Entitätsaktualisierungen) für einen Tick abgeschlossen sein müssen, bevor der nächste Tick beginnen kann. Die Tickrate wird auch als Taktrate bezeichnet.

Die gültigen Werte für `tick_rate` hängen von den in Ihrem Schema `sdk_version` angegebenen Werten ab.

Gültige Werte für die Tickrate

- Versionen vor "1.14":
 - 10
 - 15
 - 30
- Version "1.14" oder später:
 - "10"
 - "15"
 - "30"
 - "unlimited"

Weitere Informationen finden Sie unter [Unbegrenzte Tickrate](#).

Important

- Bei Schemas, bei denen `sdk_version` ein Wert vor "1.14" dem Wert von eine Ganzzahl `tick_rate` ist, z. B. 30
- Bei Schemas mit einem `sdk_version` "1.14" oder einem späteren Wert `tick_rate` ist der Wert von eine Zeichenfolge, z. B. "30" Der Wert muss die doppelten Anführungszeichen enthalten.

Wenn Sie eine Version "1.12" oder ein "1.13" Schema in Version "1.14" oder höher konvertieren, müssen Sie den Wert von `tick_rate` in doppelte Anführungszeichen setzen.

Unbegrenzte Tickrate

Sie können den Wert `tick_rate` auf `unlimited` setzen, damit Ihre Simulation so schnell ausgeführt werden kann, wie Ihr Code ausgeführt werden kann. Bei einer unbegrenzten Tick-Rate wird der nächste Tick sofort SimSpace Weaver gesendet, nachdem alle Apps die Commits für den aktuellen Tick abgeschlossen haben.

Important

Eine unbegrenzte Tick-Rate wird in SimSpace Weaver Versionen vor 1.14.0 nicht unterstützt. Der Mindestwert von `sdk_version` im Schema ist. "1.14"

Unbegrenzte Tick-Rate in SimSpace Weaver Local

SimSpace Weaver Local implementiert, "unlimited" als ob das Schema eine Tick-Rate von 10 kHz (10000) spezifiziert hätte. Der Effekt ist derselbe wie eine unbegrenzte Tick-Rate in der AWS Cloud. Sie geben immer noch `tick_rate: "unlimited"` in Ihrem Schema an. Weitere Informationen zur SimSpace Weaver Local, finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).

Häufig gestellte Fragen zur Uhr

F1. Kann ich eine STARTET-Simulation so ändern, dass sie eine andere Tickrate verwendet?

Sie können die Tick-Rate einer Simulation, die bereits in einer Phase ihres Lebenszyklus vorhanden ist, nicht ändern. AWS Cloud Sie können auch nicht die Tick-Rate einer Simulation ändern, die in läuft SimSpace Weaver Local. Sie können das `tick_rate` im Schema festlegen und von diesem Schema aus eine neue Simulation starten.

F2. Kann ich meine Simulation mit einer unbegrenzten Tick-Rate in einer Version vor 1.14 ausführen?

Nein, eine unbegrenzte Tickrate wird in Versionen vor 1.14.0 nicht unterstützt.

Behebung von Uhrfehlern

Wenn Ihre Simulation nicht gestartet werden kann, können Sie den Wert von `StartError` in der Ausgabe der `DescribeSimulationAPI` überprüfen. Ein ungültiger `tick_rate` Wert in Ihrem Schema führt zu den folgenden Fehlern.

Note

Die hier gezeigte Fehlerausgabe wird zur besseren Lesbarkeit in mehreren Zeilen angezeigt. Die tatsächliche Fehlerausgabe ist einzeilig.

- Das `sdk_version` ist früher als "1.14" und der Wert von `tick_rate` ist eine ungültige Ganzzahl. Zulässige Werte: 10, 15, 30

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

- Der `sdk_version` ist früher als "1.14" und der Wert von `tick_rate` ist eine Zeichenfolge. Zulässige Werte: 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\"}]"
```

- Das `sdk_version` ist "1.14" oder später und der Wert von `tick_rate` ist eine ungültige Zeichenfolge. Zulässige Werte: "10", "15", "30", "unlimited"

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

- Das `sdk_version` ist "1.14" oder später und der Wert von `tick_rate` ist eine Ganzzahl. Zulässige Werte: "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"}]"
```

Partitionierungsstrategien

`partitioning_strategies` In diesem Abschnitt werden die Konfigurationseigenschaften für die Partitionen von räumlichen Apps angegeben. Sie geben Ihren eigenen Namen für eine Partitionierungsstrategie an (eine Reihe von Eigenschaften in diesem Abschnitt) und verwenden ihn in Ihrer Spatial-App-Konfiguration.

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

Die `topology` Eigenschaft gibt den Typ des Koordinatensystems an, das in Ihrer Simulation verwendet wird. Der Wert `Grid` gibt ein 2-dimensionales (2D) Gitter an.

Bei einer `Grid` Topologie wird der Simulationsraum als achsenausgerichteter Begrenzungsrahmen modelliert (AABB). Sie geben die Koordinatengrenzen für jede Achse Ihres AABB in der `aabb_bounds` Eigenschaft an. Alle Entitäten, die in Ihrer Simulation räumlich existieren, müssen eine Position innerhalb des AABB haben.

Gruppen zur Rasterplatzierung

Eine Platzierungsgruppe ist eine Sammlung von räumlichen App-Partitionen, die Sie auf demselben Worker platzieren möchten SimSpace Weaver . Sie geben die Anzahl und Anordnung der Platzierungsgruppen (in einem Raster) in der `grid_placement_groups` Eigenschaft an. SimSpace Weaver versucht, die Partitionen gleichmäßig auf die Platzierungsgruppen zu verteilen. Die Eigentumsbereiche von Apps mit räumlichen Daten und Partitionen in derselben Platzierungsgruppe werden räumlich nebeneinander liegen.

Wir empfehlen, dass $x * y$ der gewünschten Anzahl von Mitarbeitern entspricht. Wenn es nicht gleich ist, SimSpace Weaver wird versucht, Ihre Platzierungsgruppen auf die verfügbaren Arbeitskräfte aufzuteilen.

Wenn Sie keine Platzierungsgruppenkonfiguration angeben, SimSpace Weaver wird eine für Sie berechnet.

Domains

Sie geben einen Namen für eine Reihe von Konfigurationseigenschaften für eine Domäne an. Die Starteinstellung für Apps in einer Domain bestimmt den Typ der Domain:

- **launch_apps_via_start_app_call**— benutzerdefinierte Domain
- **launch_apps_by_partitioning_strategy**— räumliche Domäne
- **launch_apps_per_worker**(nicht in der Beispielanwendung enthalten) — Dienstdomäne

Important

SimSpace Weaver unterstützt bis zu 5 Domänen für jede Simulation. Dies umfasst alle räumlichen, benutzerdefinierten und Dienstdomänen.

```
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 App SDK-Projekte der Version 1.12.x verwenden separate Buckets für die ZIP-Dateien der App und das Schema:

- Weber- -app-zips- *lowercase-project-name account-number region*
- Weber- -Schemas- *lowercase-project-name account-number region*

Themen

- [Konfiguration der App](#)
- [Konfiguration von räumlichen Domänen](#)
- [Netzwerk-Endpunkte](#)
- [Konfiguration von Dienstdomänen](#)

Konfiguration der App

Sie geben die Konfiguration einer App (`app_config`) als Teil der Konfiguration für ihre Domain an. Alle Arten von Domänen verwenden dieselben App-Konfigurationseigenschaften.

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

Note

SimSpace Weaver App SDK-Projekte der Version 1.12.x verwenden separate Buckets für die ZIP-Dateien der App und das Schema:

- Weber- --app-zips- *lowercase-project-name account-number region*
- Weber- --Schemas- *lowercase-project-name account-number region*

Die `package` Eigenschaft spezifiziert den S3-URI einer Zip-Datei in einem S3-Bucket. Die ZIP-Datei enthält die ausführbare Datei der App (auch als Binärdatei bezeichnet) und alle anderen Ressourcen, die sie benötigt (z. B. Bibliotheken). Jede Instanz der ausführbaren App läuft in einem Docker Container auf einem Worker.

Die `launch_command` Eigenschaft gibt den Namen der ausführbaren Datei und alle Befehlszeilenoptionen zum Ausführen der App an. Der Wert von `launch_command` ist ein Array. Jedes Token der gesamten Startbefehlszeichenfolge ist ein Element im Array.

Beispiel

- Für den Startbefehl: `MyTestApp --option1 value1`
- Geben Sie an: `launch_command: ["MyTestApp", "--option1", "value1"]`

Die `required_resource_units` Eigenschaft gibt die Anzahl der Rechenressourceneinheiten an, die dieser App zugewiesen werden SimSpace Weaver sollen. Eine Rechenressourceneinheit ist eine feste Menge an Verarbeitungskapazität (vCPU) und Speicher (RAM) an einem Arbeiter. Sie können diesen Wert erhöhen, um die Rechenleistung zu erhöhen, die der App zur Verfügung steht, wenn sie auf einem Worker ausgeführt wird. Jeder Worker verfügt über eine begrenzte Anzahl von Rechenressourceneinheiten. Weitere Informationen finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#).

Konfiguration von räumlichen Domänen

Für räumliche Domänen müssen Sie eine `partitioning_strategy` angeben. Der Wert dieser Eigenschaft ist der Name, den Sie einer Partitionierungsstrategie gegeben haben, die Sie in einem anderen Teil des Schemas definiert haben.

```
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 App SDK-Projekte der Version 1.12.x verwenden separate Buckets für die ZIP-Dateien der App und das Schema:

- Weber- -app-zips- *lowercase-project-name account-number region*
- Weber- -Schemas- *lowercase-project-name account-number region*

Eine Partitionierungsstrategie mit einer Grid Topologie (die einzige Topologie, die in dieser Version unterstützt wird) zielt darauf ab, räumliche App-Partitionen dieser Domain in einem Raster SimSpace Weaver anzuordnen. Die `grid_partition` Eigenschaft gibt die Anzahl der Zeilen und Spalten des Partitionsrasters an.

SimSpace Weaver startet für jede Zelle im Partitionsraster eine Instanz der Spatial-App. Wenn eine räumliche Domäne beispielsweise `grid_partition` Werte hat `x: 2` und `y: 2` es $2 * 2 = 4$ Partitionen in der räumlichen Domäne gibt. SimSpace Weaver startet 4 Instanzen der App, die in der räumlichen Domäne konfiguriert sind, und weist jeder App-Instanz eine Partition zu.

Themen

- [Ressourcenanforderungen für räumliche Domänen](#)
- [Mehrere räumliche Domänen](#)
- [Häufig gestellte Fragen zu räumlichen Domänen](#)
- [Problembehandlung bei räumlichen Domänen](#)

Ressourcenanforderungen für räumliche Domänen

Sie können jedem Mitarbeiter bis zu 17 Rechenressourceneinheiten zuweisen. Sie geben im `app_config` Abschnitt Ihrer räumlichen Domäne die Anzahl der Rechenressourceneinheiten an, die jede Geodaten-App verwendet.

Example Schemaausschnitt, der die Rechenressourceneinheiten für eine Geodaten-App zeigt

```
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
```

Um die Anzahl der Rechenressourceneinheiten zu berechnen, die eine Domain benötigt, multiplizieren Sie die Anzahl der Zellen in Ihrem Grid (in Ihrem `grid_partition`, $x * y$) mit der Anzahl der Rechenressourceneinheiten, die den Geo-Apps zugewiesen sind.

Für das vorherige Beispiel `MySpatialDomain` gibt die Domäne Folgendes an:

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

Das Raster für `MySpatialDomain` hat $2 * 2 = 4$ Zellen. Für die räumliche Domäne sind $4 * 1 = 4$ Rechenressourceneinheiten erforderlich.

Die Gesamtzahl der Rechenressourceneinheiten für alle in Ihrem Schema angegebenen Domänen muss kleiner oder gleich der `desired` Anzahl der Worker multipliziert mit der maximalen Anzahl von Rechenressourceneinheiten für jeden Worker (17) sein.

Mehrere räumliche Domänen

Sie können Ihre Simulation so konfigurieren, dass mehr als eine räumliche Domäne verwendet wird. Sie können beispielsweise einen räumlichen Bereich verwenden, um die Hauptakteure einer Simulation (z. B. Menschen und Autos) zu steuern, und einen anderen räumlichen Bereich, um die Umgebung zu kontrollieren.

Sie können auch mehrere räumliche Domänen verwenden, um verschiedenen Teilen Ihrer Simulation unterschiedliche Ressourcen zuzuweisen. Wenn Ihre Simulation beispielsweise einen Entitätstyp hat, der zehnmal mehr Entitätsinstanzen als ein anderer Typ hat, können Sie verschiedene Domänen erstellen, um jeden Entitätstyp zu behandeln, und der Domäne mit mehr Entitäten mehr Ressourcen zuweisen.

Important

SimSpace Weaver Versionen vor 1.14.0 unterstützen nicht mehrere räumliche Domänen.

Important

AWS SimSpace Weaver Local unterstützt derzeit nicht mehrere räumliche Domänen. Weitere Informationen zur SimSpace Weaver Local, finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).

Important

SimSpace Weaver unterstützt bis zu 5 Domänen für jede Simulation. Dies umfasst alle räumlichen, benutzerdefinierten und Dienstdomänen.

Konfigurieren Sie mehrere räumliche Domänen

Um mehr als eine räumliche Domäne zu konfigurieren, fügen Sie die anderen räumlichen Domänendefinitionen als separate benannte Abschnitte in Ihrem Schema hinzu. Jede Domäne muss den `launch_apps_by_partitioning_strategy` Schlüssel angeben. Sehen Sie sich das folgende Beispielschema an.

```
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
```

Räumliche Domänen zusammenfügen

In einigen Szenarien möchten Sie möglicherweise Partitionen für eine räumliche Domäne auf Workern neben Partitionen aus einer anderen Domäne platzieren. Dies kann die Leistungsmerkmale verbessern, wenn diese Partitionen domänenübergreifende Abonnements für einander erstellen.

Fügen Sie Ihrem Schema den Schlüssel `placement_constraints` der obersten Ebene hinzu, um anzugeben, welche Domänen zusammen platziert SimSpace Weaver werden sollen. Der erforderliche `on_workers` Schlüssel muss sich auf eine benannte `workers` Konfiguration im Schema beziehen.

Example Schemaausschnitt, der zusammengesetzte räumliche Domänen zeigt

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

Important

- Wenn Sie Platzierungsgruppen verwenden:
 - Stellen Sie sicher, dass $x * y$ ein Vielfaches der Anzahl der Mitarbeiter ist.
 - Stellen Sie sicher, dass die Werte der Platzierungsgruppen gemeinsame Teiler für die Rastermaße der Domänen sind, die Sie zusammen platzieren.
- Wenn Sie keine Platzierungsgruppen verwenden:
 - Stellen Sie sicher, dass eine Achse Ihrer räumlichen Domänengitter einen gemeinsamen Teiler hat, der der Anzahl der Arbeiter entspricht.

Weitere Informationen zu Platzierungsgruppen finden Sie unter [Partitionierungsstrategien](#)

Häufig gestellte Fragen zu räumlichen Domänen

F1. Wie kann ich einer vorhandenen Simulation eine weitere räumliche Domäne hinzufügen?

- Für eine laufende Simulation — Sie können die Konfiguration für eine laufende Simulation nicht ändern. Ändern Sie die Domänenkonfiguration im Schema, laden Sie das Schema und die App-Zip-Dateien hoch und starten Sie eine neue Simulation.
- Für eine neue Simulation — Fügen Sie die Domänenkonfiguration zum Schema hinzu, laden Sie das Schema und die App-Zip-Dateien hoch und starten Sie die neue Simulation.

Problembehandlung bei räumlichen Domänen

Möglicherweise wird der folgende Fehler angezeigt, wenn Sie versuchen, Ihre Simulation zu starten, Ihre Domänenkonfiguration jedoch ungültig ist.

```
"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  
  [\u0027grid_partition\u0027] values, or adjusting the  
  dimensions of your [\u0027grid_placement_groups\u0027].\u0027}"]"
```

Mögliche Ursachen

- Das Schema weist Apps mehr Rechenressourceneinheiten zu, als Workern zur Verfügung stehen.
- SimSpace Weaver kann keine Vereinbarung treffen, um Domänen auf Workern zusammen zu platzieren. Dies ist der Fall, wenn Sie mehrere räumliche Domänen angeben, es aber keinen gemeinsamen Teiler oder kein Vielfaches zwischen den Domänenrastern gibt (z. B. zwischen einem 2x4-Raster und einem 3x5-Raster).

Netzwerk-Endpunkte

Benutzerdefinierte Apps und Service-Apps können über Netzwerkendpunkte verfügen, mit denen externe Clients eine Verbindung herstellen können. Sie geben eine Liste von Portnummern als Wert für `ingress_ports` in `anendpoint_config`. Bei diesen Portnummern handelt es sich sowohl um TCP als auch um UDP. Die benutzerdefinierte App oder die Service-App sollte sich an die Portnummern binden, die Sie in `anendpoint_config.ingress_ports` angeben. SimSpace Weaver weist zur Laufzeit dynamisch Portnummern zu und ordnet diese Ports den dynamischen Ports zu. Sie können den

anrufen Describe-App API, nachdem Ihre Apps begonnen haben, die dynamischen (tatsächlichen) Portnummern zu finden. Weitere Informationen finden Sie [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#) im Schnellstart-Tutorial.

```
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 App SDK-Projekte der Version 1.12.x verwenden separate Buckets für die .zip-Dateien der App und das Schema:

- Weber- -app-zips- *lowercase-project-name account-number region*
- Weber- -Schemas- *lowercase-project-name account-number region*

Note

`endpoint_config` ist eine optionale Eigenschaft für benutzerdefinierte Apps und Service-Apps. Wenn Sie keinen angeben, hat die App keinen Netzwerkendpunkt. `endpoint_config`

Konfiguration von Dienstdomänen

Das Vorhandensein von `launch_apps_per_worker`: in einer Domänenkonfiguration weist darauf hin, dass es sich um eine Dienstdomäne mit Service-Apps handelt. SimSpace Weaver startet und stoppt Service-Apps für Sie. Beim SimSpace Weaver Starten und Beenden einer App wird davon ausgegangen, dass die App über einen verwalteten Lebenszyklus verfügt. SimSpace Weaver unterstützt derzeit das Starten von 1 oder 2 Service-Apps für jeden einzelnen Mitarbeiter.

Example Beispiel für eine Domain, die so konfiguriert ist, dass sie auf jedem Worker 1 Service-App startet

```
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 Beispiel für eine Domain, die so konfiguriert ist, dass sie 2 Service-Apps auf jedem Worker startet

```
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
```

Maximale Dauer einer Simulation

Jede Simulation in AWS SimSpace Weaver hat eine Einstellung für die maximale Dauer, die angibt, wie lange die Simulation maximal ausgeführt werden kann. Sie geben die maximale Dauer als Parameter an, wenn Sie eine Simulation starten. Die [StartSimulationAnwendungsprogrammierschnittstelle \(API\)](#) hat einen optionalen Parameter `MaximumDuration`. Der Wert des Parameters ist eine Anzahl von Minuten (m oder M), Stunden (h oder H) oder Tagen (d oder D). Zum Beispiel 1H bedeutet 1h oder 1 Stunde. SimSpace Weaver stoppt Ihre Simulation, wenn sie dieses Limit erreicht.

Maximaler Wert

Der höchste gültige Wert für `MaximumDuration` ist 14D oder sein Äquivalent in Stunden (336H) oder Minuten (20160M).

Standardwert

Der Parameter `MaximumDuration` ist optional. Wenn Sie keinen Wert angeben, wird ein Wert von SimSpace Weaver verwendet 14D.

Mindestwert

Der niedrigste gültige Wert für `MaximumDuration` ist ein Wert, der numerisch entspricht 0. Beispielsweise sind die Werte 0M, und 0H0D, alle numerisch äquivalent zu 0.

Wenn Sie den Mindestwert für die maximale Dauer angeben, geht Ihre Simulation sofort in den STOPPING Status über, sobald sie den STARTED Status erreicht.

Starten einer Simulation mit der Konsole

Sie können einen Wert für die maximale Dauer angeben, wenn Sie eine Simulation in der [SimSpace Weaver Konsole](#) starten. Geben Sie den Wert in das Feld Maximale Dauer des Formulars Simulationseinstellungen ein, wenn Sie Simulation starten wählen.

Important

Wenn Sie keinen Wert für Maximale Dauer angeben, SimSpace Weaver wird der [Standardwert](#) (14D) verwendet.

Der Status einer Simulation, die ihre maximale Dauer erreicht

Wenn eine Simulation, die ihre maximale Dauer erreicht hat, SimSpace Weaver automatisch beendet wird, lautet der Status der Simulation STOPPING (falls in Bearbeitung) oder STOPPED. In der [SimSpace Weaver Konsole](#) wird der Zielstatus der Simulation immer noch angezeigt STARTED, da dies der letzte von einem Benutzer angeforderte Status war.

Apps entwickeln

SimSpace Weaver Entwicklung erfordert eine Amazon Linux 2 (AL2) Umgebung zum Erstellen von Apps, auf der Ihre Simulationen ausgeführt werden Amazon Linux in der AWS Cloud. Wenn du verwendest Windows, können Sie Skripts im SimSpace Weaver App-SDK verwenden, um eine zu erstellen und zu starten Docker Container, der läuft AL2 mit den Abhängigkeiten, die Sie zum Erstellen von SimSpace Weaver Apps benötigen. Sie können auch eine starten AL2 Umgebung unter Verwendung Windows Subsystem for Linux (WSL), oder verwende ein natives AL2 System. Weitere Informationen finden Sie unter [Richten Sie Ihre lokale Umgebung ein für SimSpace Weaver](#).

Note

Unabhängig davon, wie Sie Ihre lokale Entwicklungsumgebung konfigurieren, laufen Ihre Apps in Docker Container, wenn Sie sie hochladen, um sie in der auszuführen AWS Cloud. Ihre Apps haben keinen direkten Zugriff auf das Host-Betriebssystem.

Allgemeiner Ablauf einer SimSpace Weaver App

1. Erstellen Sie eine -Anwendung.
2. Schleife:
 - a. Beginnen Sie das Update, indem Sie eine erstellen `Transaction`.
 - Beenden Sie die Schleife, wenn die Simulation beendet wird.
 - b. Verarbeiten Sie Ereignisse im Zusammenhang mit Abonnements und Eigentümerschaften.
 - c. Aktualisieren Sie die Simulation.
 - d. Bestätigen Sie die `Transaction`, um das Update zu beenden.
3. Zerstören Sie die Anwendung.

Räumliche Apps

Jede Geo-App hat einen Eigentumsbereich, der eine räumliche Region der Simulationswelt darstellt. Entitäten, die sich im Besitzbereich einer Geo-App befinden, werden in der der App zugewiesenen Partition gespeichert. Die einzelne Spatial-App besitzt die volle Eigentümerschaft (Lese- und Schreibberechtigungen) für alle Entitäten innerhalb der ihr zugewiesenen Partition. Keine anderen Apps können in diese Entitäten schreiben. Die Spatial-App verbessert den Status ihrer Entitäten. Jede Geo-App besitzt nur eine Partition. SimSpace Weaver verwendet den räumlichen Standort einer Entität, um sie zu indizieren und einer räumlichen App-Partition zuzuweisen.

Das SimSpace Weaver App-SDK stellt eine Beispielanwendung bereit. Sie finden den Quellcode für die Spatial-App der Beispielanwendung im folgenden Ordner (verwenden Sie das richtige Pfadtrennzeichen für Ihr Betriebssystem):

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

Benutzerdefinierte Apps

Sie erstellen und verwenden benutzerdefinierte Apps, um mit der Simulation zu interagieren.

Benutzerdefinierte Apps können

- Entitäten erstellen
- Abonnieren Sie andere Partitionen
- Änderungen übernehmen

Allgemeiner Ablauf einer benutzerdefinierten App

1. Erstellen Sie eine -Anwendung.
2. Abonnieren Sie eine bestimmte Region in der Simulation:
 - a. Erstellen Sie eine `Transaction`, um mit dem ersten Update zu beginnen.
 - b. Erstellen Sie ein Abonnement für die spezifische Region.
 - c. Bestätigen Sie die `Transaction`, um das erste Update zu beenden.
3. Schleife:
 - a. Erstellen Sie eine `Transaction`, um mit dem Update zu beginnen.

- Beenden Sie die Schleife, wenn die Simulation beendet wird.
 - b. Änderungen des Prozessstatus.
 - c. Bestätigen Sie die `Transaction`, um das Update zu beenden.
4. Zerstören Sie die Anwendung.

Nachdem eine benutzerdefinierte App eine Entität erstellt hat, muss sie die Entität in eine räumliche Domäne übertragen, damit die Entität räumlich innerhalb der Simulation existiert. SimSpace Weaver verwendet die räumliche Position der Entität, um die Entität in der entsprechenden räumlichen App-Partition zu platzieren. Die benutzerdefinierte App, mit der die Entität erstellt wurde, kann die Entität nicht aktualisieren oder löschen, nachdem sie in eine räumliche Domäne übertragen wurde.

Das SimSpace Weaver App-SDK stellt eine Beispielanwendung bereit. Sie können die in der Beispielanwendung enthaltenen benutzerdefinierten Apps als Modelle für Ihre eigenen benutzerdefinierten Apps verwenden. Sie finden den Quellcode für die View-App (eine benutzerdefinierte App) der Beispielanwendung im folgenden Ordner (verwenden Sie das richtige Pfadtrennzeichen für Ihr Betriebssystem):

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

Entwicklung von Kundenanwendungen

Zu den Gründen, aus denen Sie einen Client mit einer Simulation verbinden möchten, gehören:

- Integrieren Sie Verkehrsinformationen in Echtzeit in eine Simulation im Stadtmaßstab.
- Erstellen Sie human-in-the-loop-Simulationen, bei denen ein menschlicher Bediener einen Teil der Simulation kontrolliert.
- Ermöglichen Sie Benutzern die Interaktion mit der Simulation, z. B. bei einer Trainingssimulation.

Die benutzerdefinierten Apps in diesen Beispielen dienen als Schnittstelle zwischen dem Simulationsstatus und der Außenwelt. Die Clients stellen eine Verbindung zu den benutzerdefinierten Apps her, um mit der Simulation zu interagieren.

SimSpace Weaver kümmert sich nicht um die Client-Anwendungen und deren Kommunikation mit Ihren benutzerdefinierten Apps. Sie sind verantwortlich für das Design, die Erstellung, den Betrieb und die Sicherheit Ihrer Client-Anwendungen und deren Kommunikation mit Ihren

benutzerdefinierten Apps. SimSpace Weaver gibt nur eine IP-Adresse und eine Portnummer für jede Ihrer benutzerdefinierten Apps bekannt, sodass Clients eine Verbindung zu ihnen herstellen können.

Das SimSpace Weaver App-SDK stellt Clients für seine Beispielanwendung bereit. Sie können diese Clients als Modelle für Ihre eigenen Client-Anwendungen verwenden. Sie finden den Quellcode für die Beispielanwendungsclients im folgenden Ordner:

Docker

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

WSL

Important

Wir stellen Ihnen diese Anweisungen der Einfachheit halber zur Verfügung. Sie sind zur Verwendung mit Windows Subsystem for Linux (WSL), und werden nicht unterstützt.

Weitere Informationen finden Sie unter [Richten Sie Ihre lokale Umgebung ein für SimSpace Weaver](#).

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

Weitere Informationen zum Erstellen und Verwenden der Beispielanwendungsclients finden Sie in [Erste Schritte mit SimSpace Weaver](#) den Tutorials unter.

Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab

Um Ihre Simulation anzusehen, erstellen Sie eine benutzerdefinierte App und stellen mit einem Client eine Verbindung zu ihr her. Weitere Informationen finden Sie in den Tutorials unter [Erste Schritte mit SimSpace Weaver](#). Sie können das folgende Verfahren verwenden, um die IP-Adresse und Portnummer Ihrer benutzerdefinierten App abzurufen. Verwenden Sie das entsprechende Pfadtrennzeichen für Ihr Betriebssystem (z. B. \ unter Windows und / unter Linux).

Um Ihre IP-Adresse und Portnummer zu erhalten

1. Verwenden Sie die `ListSimulationsAPI`, um den Namen Ihrer Simulation abzurufen.

```
aws simspaceweaver list-simulations
```

Beispielausgabe:

```
{
  "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. Verwenden Sie die `DescribeSimulationAPI`, um eine Liste der Domänen in Ihrer Simulation abzurufen.

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

Suchen Sie im `Domains LiveSimulationState` Abschnitt der Ausgabe nach dem Abschnitt.

Beispielausgabe:

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

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

3. Verwenden Sie die `ListAppsAPI`, um eine Liste der benutzerdefinierten Apps in einer Domain abzurufen. Der Domainname für die View-App (benutzerdefiniert) im Beispielprojekt lautet beispielsweise `MyViewDomain`. Suchen Sie in der Ausgabe nach dem Namen der App.

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

Beispielausgabe:

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

4. Verwenden Sie die `DescribeAppAPI`, um die IP-Adresse und die Portnummer abzurufen. Für das Beispielprojekt lautet der Domainname `MyViewDomain` und der App-Name `ViewApp`.

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

Die IP-Adresse und die Portnummer befinden sich im `EndpointInfo` Block in der Ausgabe. Die IP-Adresse ist der Wert von `Address` und die Portnummer ist der Wert von `Actual`.

Beispielausgabe:

```
{
  "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

Der Wert von `Declared` ist die Portnummer, an die Ihr App-Code gebunden werden soll. Der Wert von `Actual` ist die Portnummer, über SimSpace Weaver die Clients eine Verbindung zu Ihrer App herstellen können. SimSpace Weaver ordnet den `Declared` Port dem Port zu. `Actual`

Den Unreal Engine View Client starten

Navigiere zu:

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

1. Führen Sie einen der folgenden Befehle aus:
 - Docker: `python quick-start.py`
 - WSL: `python quick-start.py --a12`
2. Ermitteln Sie die IP-Adresse und die „tatsächliche“ Portnummer. Diese befinden sich in der Konsolenausgabe, wenn Sie `quick-start.py` ausführen, oder rufen Sie sie ab, indem Sie den Anweisungen unter folgen [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#).

3. Navigieren Sie zu:

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

4. Führen Sie die folgenden Befehle aus, um die NNG-Bibliothek zu erstellen:

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

5. Öffnen `view_app_url.txt` Sie in einem Texteditor.
6. Aktualisieren Sie die URL mit der IP-Adresse und der Portnummer für Ihre View-App: `tcp://ip-address:actual-port-number` (sie sollte so aussehentcp://198.51.100.135:1234).
7. Wählen Sie im Unreal Editor die Option Abspielen.

Fehlerbehebung

- Der CMake NNG-Installationsschritt schlägt mit der Meldung „Möglicherweise benötige ich Administratorrechte“ fehl:

```
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)
```

- Lösung: Falls `nng.lib` er im Verzeichnis `UnrealClient/lib/nng` so existiert, kann dieser Fehler getrost ignoriert werden. Falls nicht, versuchen Sie, die Befehle `cmake build` in einem Terminal mit Administratorrechten auszuführen.
- „CMake um eine von `nng` bereitgestellte Paketkonfigurationsdatei zu finden“:

```
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.
```

- Lösung: CMake hat Probleme, die `Findnng.cmake` Datei zu finden. Wenn Sie mit `cmake build` CMake, fügen Sie das Argument hinzu `-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty`. Stellen Sie sicher, dass sich die `Findnng.cmake` Datei noch im `ThirdParty` Verzeichnis befindet, bevor Sie den CMake Build erneut ausführen.

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

Lokale Entwicklung in SimSpace Weaver

Sie können Ihre SimSpace Weaver Anwendungen lokal bereitstellen, um sie schnell zu testen und zu debuggen.

Voraussetzungen

- Führen Sie die Schritte unter [Einrichtung für SimSpace Weaver](#) aus.

Themen

- [Schritt 1: Starten Sie Ihre lokale Simulation](#)
- [Schritt 2: Sehen Sie sich Ihre lokale Simulation an](#)
- [Schritt 3: Beenden Sie Ihre lokale Simulation \(optional unter Windows\)](#)
- [Problembhebung bei der lokalen Entwicklung in SimSpace Weaver](#)

Schritt 1: Starten Sie Ihre lokale Simulation

1. Navigieren Sie zu

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

2. Führen Sie den folgenden Befehl aus, um Ihre Simulation lokal zu erstellen und zu starten.

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

Dieses Skript führt Folgendes aus:

1. Erstellen Sie das Projekt.

- `quick-start.py` ruft die in `build.py` definierte `build_project` Funktion auf. Dieser Schritt variiert je nach Projekt. Für das `PathfindingSample`, CMake wird verwendet. Den CMake und Docker-Befehl finden Sie in der Datei `build.py`.

2. Starten Sie Ihre lokale Simulation

- Das Skript startet einen lokalen Prozess für jede im Schema definierte räumliche Partition.
- Das Skript startet einen Prozess für jede im Schema definierte benutzerdefinierte App.
- Die räumlichen Apps werden zuerst gestartet, gefolgt von den benutzerdefinierten Apps — jeweils in der Reihenfolge, in der sie im Schema erscheinen.

Important

Verwenden Sie beim Starten in einer Umgebung, die keine grafische Benutzeroberfläche unterstützt, z. B. in einer SSH-Sitzung auf der Konsole, die `--noappwindow` Option, um die gesamte Ausgabe an das aktuelle Terminal umzuleiten.

Important

Für Linux-Benutzer geht das Skript davon aus, dass Ihr System über den `xterm` Befehl verfügt. Wenn Ihre Linux-Distribution den `xterm` Befehl nicht hat, verwenden Sie die `--noappwindow` Option, um die gesamte Ausgabe an das aktuelle Terminal umzuleiten.

- `-h, --help`
 - Führen Sie diese Parameter auf.
- `--sauber`
 - Löscht den Inhalt des Build-Verzeichnisses vor dem Erstellen.
- `--nobuild`
 - Überspringe den Neuaufbau des Projekts.
- `--kein App-Fenster`
 - Öffne nicht für jede App ein neues Fenster. Leite stattdessen den Stdout zum aktuellen Terminal um.
- `--logdatei`
 - Schreibt die Konsolenausgabe in eine Protokolldatei.
- `--consoleclient`
 - Stellt automatisch eine Verbindung zum in der Konfiguration aufgelisteten Konsolen-Client her.
- `--schema SCHEMA`
 - Welches Schema dieser Aufruf verwenden wird. Standardmäßig ist 'SCHEMA' in `config.py`.

Schritt 2: Sehen Sie sich Ihre lokale Simulation an

Um Ihre lokale Simulation anzusehen, können Sie jeden der Clients verwenden, die im Lieferumfang von `SimSpaceWeaverAppSdkDistributable` enthalten sind. Weitere Informationen zum Erstellen und Verwenden der Beispielclients finden Sie in den Tutorials unter [Erste Schritte mit SimSpace Weaver](#).

Sie müssen die IP-Adresse und die Portnummer im Client aktualisieren, um eine Verbindung zur View-App für Ihre lokale Simulation herzustellen. Verwenden Sie immer die folgenden Werte mit `SimSpace Weaver Local`:

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

Je nachdem, welchen Client Sie auswählen, können Sie die IP-Adresse und die Portnummer wie folgt aktualisieren:

- **Unreal** — Ändern Sie die URL in Zeile 1 von `view_app_url.txt`
- **Konsole** — Starten Sie den Client mit der IP-Adresse und der Portnummer (URL) als Parameter

Schritt 3: Beenden Sie Ihre lokale Simulation (optional unter Windows)

Note

Dieser Schritt ist unter Linux erforderlich, unter Windows jedoch optional.

1. Navigiere zu:

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

2. Führen Sie den folgenden Befehl aus, um Ihre lokale Simulation zu beenden und alle gemeinsam genutzten Speicherressourcen zu löschen.

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

Dieses Skript führt Folgendes aus:

- Stoppen Sie die lokalen Prozesse.
- Löschen Sie das gemeinsam genutzte Speicherobjekt (nur unter Linux erforderlich).

stop-and-delete.py-Parameter

- -h, --help
 - Führen Sie diese Parameter auf.
- --stop
 - Versuche nur, die Prozesse zu stoppen.
- --löschen
 - Versuchen Sie nur, die gemeinsam genutzten Speicherressourcen zu löschen.
- --process
 - Der Name des Prozesses, der gestoppt werden soll. Verwenden Sie diesen Wert, wenn Ihr Prozessname nicht mit dem Paketnamen im Schema übereinstimmt.
- --schema SCHEMA
 - Welches Schema dieser Aufruf verwendet wird. Standardmäßig ist der Wert 'SCHEMA' in config.py.

Problembhebung bei der lokalen Entwicklung in SimSpace Weaver

- Linux: Xterm-Befehl nicht gefunden/kann nicht geöffnet werden
 - Die lokalen Skripts gehen davon aus, dass der Befehl xterm existiert, wenn er unter Linux ausgeführt wird. Wenn Sie nicht über den Befehl xterm verfügen oder in einer Umgebung arbeiten, die die grafische Benutzeroberfläche nicht unterstützt, verwenden Sie die `--noappwindow` Option, wenn Sie das Schnellstartskript ausführen.
- Es werden keine App-Fenster geöffnet!
 - Dies passiert, wenn die lokale Simulation sofort abstürzt. Um die Konsolenausgabe nach dem Absturz zu sehen, verwenden Sie beim Ausführen des Schnellstart-Skripts die `--logfile` Optionen `--noappwindow` oder.
- Die Simulation tickt nicht mehr, nachdem die View-App gestartet oder der View-Client eine Verbindung hergestellt hat!
 - Wenn Sie diese `--noappwindow` Option verwenden, werden diese Probleme in der Regel behoben. Andernfalls ist ein mehrmaliger Neustart ebenfalls erfolgreich (wenn auch mit einer viel geringeren Geschwindigkeit).

AWS SimSpace Weaver App-SDK

Das SimSpace Weaver App-SDK bietet APIs die Möglichkeit, die Entitäten in Ihrer Simulation zu steuern und auf SimSpace Weaver Ereignisse zu reagieren. Es umfasst den folgenden Namespace:

- API — Kerndefinitionen der API und ihrer Verwendung

Link zur folgenden Bibliothek:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

Die Bibliothek ist für dynamisches Verknüpfen verfügbar, wenn Sie Ihre Apps in der ausführen AWS Cloud. Sie müssen es nicht zusammen mit Ihren Apps hochladen.

Note

Das SimSpace Weaver App-SDK APIs steuert die Daten in Ihrer Simulation. Diese APIs sind unabhängig vom SimSpace Weaver Dienst APIs, der Ihre SimSpace Weaver Servicere Ressourcen (wie Simulationen, Apps und Uhren) steuert. AWS Weitere Informationen finden Sie unter [SimSpace Weaver API-Referenzen](#).

Themen

- [API-Methoden geben a zurück Result](#)
- [Interaktion mit dem App-SDK auf oberster Ebene](#)
- [Verwaltung der Simulation](#)
- [Subscriptions \(Abonnements\)](#)
- [Entitäten](#)
- [Ereignisse der Entität](#)
- [Result und Fehlerbehandlung](#)
- [Generika und Domaintypen](#)
- [Verschiedene App-SDK-Operationen](#)

API-Methoden geben a zurück Result

Die meisten SimSpace Weaver API-Funktionen haben einen Rückgabety `Aws::WeaverRuntime::Result<T>`. Wenn die Funktion erfolgreich ausgeführt wurde, `Result` enthält der `T`. Andernfalls `Result` enthält der einen `Aws::WeaverRuntime::ErrorCode`, der einen Fehlercode aus dem Rust App SDK.

Example Beispiel

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

Diese Methode:

- Gibt zurück `Transaction`, wenn die `BeginUpdate()` Ausführung erfolgreich war.
- Gibt zurück, `Aws::WeaverRuntime::ErrorCode` wenn ein `BeginUpdate()` Fehler auftritt.

Interaktion mit dem App-SDK auf oberster Ebene

Lebenszyklus

- Das SimSpace Weaver App-SDK verwaltet den App-Lebenszyklus. Sie müssen den Lebenszyklusstatus einer App nicht lesen oder schreiben.

Partitionen

- Wird verwendet `Result <PartitionSet> AssignedPartitions(Transaction& txn);`, um eigene Partitionen abzurufen.
- Wird verwendet `Result <PartitionSet> AllPartitions(Transaction& txn);`, um alle Partitionen in der Simulation abzurufen.

Verwaltung der Simulation

In diesem Abschnitt werden Lösungen für häufig auftretende Aufgaben des Simulationsmanagements beschrieben.

Themen

- [Starten Sie eine Simulation](#)
- [Aktualisieren Sie eine Simulation](#)
- [Beendet eine Simulation](#)

Starten Sie eine Simulation

Verwenden Sie `CreateApplication()`, um eine App zu erstellen.

Example Beispiel

```
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()));

```

Aktualisieren Sie eine Simulation

Verwenden Sie die folgenden BeginUpdate Funktionen, um die App zu aktualisieren:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`— sagt dir, ob blockiert `BeginUpdate()` wird oder nicht.

Wird verwendet `Result<void> Commit(Transaction& txn)`, um die Änderungen zu übernehmen.

Example Beispiel

```

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();
}

```

Beendet eine Simulation

Wird verwendet `Result<void> DestroyApplication(Application&& app)`, um die App und die Simulation zu beenden.

Andere Apps stellen fest, dass die Simulation beendet wird, wenn sie `ErrorCode::ShuttingDown` von ihren Anrufen an `BeginUpdateWillBlock()` oder `BeginUpdate()` empfangen. Wenn eine App empfängt `ErrorCode::ShuttingDown`, kann sie anrufen, um sich selbst `Result<void> DestroyApplication(Application&& app)` zu beenden.

Example Beispiel

```

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

Rufen Sie erst `Result<void> DestroyApplication(Application&& app)` danach `anApi::Commit()`. Das Zerstören einer Anwendung während eines Updates kann zu undefiniertem Verhalten führen.

⚠ Important

Sie müssen `DestroyApplication()` vor dem Beenden des Programms aufrufen, um sicherzustellen, dass die Anwendung als erfolgreich beendet gemeldet wird.

`DestroyApplication()` Wenn Sie beim Beenden des Programms nicht aufrufen, wird der Status als betrachtet. FATAL

Subscriptions (Abonnements)

Sie erstellen ein Abonnement mit einem Abonnementbereich und einer Domain-ID. Die Domain-ID steht für die Domain, der dieser Abonnementbereich gehört. A `BoundingBox2F32` beschreibt den Abonnementbereich. Verwenden Sie die folgende Funktion, um ein Abonnement zu erstellen:

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

Example Beispiel

```
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();
}

```

Sie können das `Api::SubscriptionHandle` zurückgegebene von `useCreateSubscriptionBoundingBox2F32()`, um das Abonnement zu ändern. Sie übergeben es als Argument an die folgenden Funktionen:

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

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

Entitäten

Wenn eine Entität den Abonnementbereich `Api::Entity` der App betritt `CreateEntity()`, rufen Sie das Ereignis auf Store und Load APIs verwenden dabei das Ereignis, das von einem Eigentümerwechsel `Result<Api::Entity>` zurückgegeben wurde (weitere Informationen finden Sie unter [Ereignisse der Entität](#)). Wir empfehlen, dass Sie Ihre `Api::Entity` Objekte nachverfolgen, damit Sie sie zusammen mit ihnen verwenden können APIs.

Themen

- [Entitäten erstellen](#)
- [Eine Entität in eine räumliche Domäne übertragen](#)

- [Daten aus Entitätsfeldern schreiben und lesen](#)
- [Speichert die Position einer Entität](#)
- [Lädt die Position einer Entität](#)

Entitäten erstellen

Wird verwendet `CreateEntity()`, um eine Entität zu erstellen. Sie definieren die Bedeutung dessen `Api::TypeId`, was Sie an diese Funktion übergeben.

```
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

Die Werte 0-511 für `Api::BuiltinTypeId` sind reserviert. Ihre Entität `TypeId` (`k_entityTypeId` in diesem Beispiel) muss einen Wert von 512 oder höher haben.

Eine Entität in eine räumliche Domäne übertragen

Nachdem eine benutzerdefinierte App oder Service-App eine Entität erstellt hat, muss die App die Entität in eine räumliche Domäne übertragen, damit die Entität räumlich in der Simulation existiert. Entitäten in einer räumlichen Domäne können von anderen Apps gelesen und von einer räumlichen App aktualisiert werden. Verwenden Sie die `ModifyEntityDomain()` API, um eine Entität in eine räumliche Domäne zu übertragen.

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

Wenn das `DomainId` nicht mit dem der aufrufenden App zugewiesenen `Partition` übereinstimmt, `DomainId` muss das für `a` sein `DomainType::SpatialDomain`. Die Eigentumsübertragung auf die neue Version `Domain` erfolgt während `derCommit(Transaction&&)`.

Parameter

`txn`

Das aktuelle `Transaction`.

`entity`

Das Ziel `Entity` für die Änderung von `Domain`.

`domainId`

Das `DomainId` Ziel `Domain` für die `Entity`.

Diese API gibt zurück `Success`, ob die Entitätsdomäne erfolgreich geändert wurde.

Daten aus Entitätsfeldern schreiben und lesen

Alle Entitätsdatenfelder sind `Blob`-Typen. Sie können bis zu 1.024 Byte an Daten in eine Entität schreiben. Wir empfehlen, `Blobs` so klein wie möglich zu halten, da größere Größen die Leistung beeinträchtigen. Wenn Sie in ein `Blob` schreiben, übergeben Sie `SimSpace Weaver` einen Zeiger auf die Daten und deren Länge. Wenn Sie aus einem `Blob` lesen, `SimSpace Weaver` erhalten Sie einen Zeiger und eine Länge zum Lesen. Alle Lesevorgänge müssen abgeschlossen sein, bevor die App aufgerufen `Commit()` wird. Von einem Leseaufruf zurückgegebene Zeiger werden ungültig, wenn die App aufruft. `Commit()`

Important

- Das Lesen von einem zwischengespeicherten `Blob-Pointer` nach `a Commit()` wird nicht unterstützt und kann dazu führen, dass die Simulation fehlschlägt.
- Das Schreiben in einen `Blob-Zeiger`, der von einem Leseaufruf zurückgegeben wurde, wird nicht unterstützt und kann dazu führen, dass die Simulation fehlschlägt.

Themen

- [Speichern Sie die Felddaten einer Entität](#)

- [Lädt die Felddaten einer Entität](#)
- [Die Felddaten der entfernten Entitäten werden geladen](#)

Speichern Sie die Felddaten einer Entität

Die folgenden Beispiele zeigen, wie Sie die Felddaten einer Entität speichern (in die State Fabric schreiben) können, deren Eigentümer die App ist. In diesen Beispielen wird die folgende Funktion verwendet:

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

Der `Api::TypeId keyTypeId` Parameter repräsentiert den Datentyp der übergebenen Daten.

Der `Api::TypeId keyTypeId` Parameter sollte das entsprechende Formular `Api::TypeId` erhalten `Api::BuiltinTypeId`. Wenn es keine entsprechende Konvertierung gibt, können Sie verwenden `Api::BuiltinTypeId::Dynamic`.

Verwenden Sie für komplexe Datentypen `Api::BuiltinTypeId::Dynamic`.

Note

Der Wert von `FieldIndex index` muss größer als 0 sein. Der Wert 0 ist für den Indexschlüssel reserviert (siehe `StoreEntityIndexKey()`).

Example Beispiel für die Verwendung primitiver Datentypen

```
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 Beispiel mit einem struct um die Daten zu speichern

```

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));
}

```

Lädt die Felddaten einer Entität

Die folgenden Beispiele zeigen, wie Sie die Felddaten einer Entität laden (aus der State Fabric lesen) können. In diesen Beispielen wird die folgende Funktion verwendet:

```

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

```

Der `Api::TypeId keyTypeId` Parameter sollte das entsprechende Format `Api::TypeId` von `Api::BuiltinTypeId` erhalten. Wenn es keine entsprechende Konvertierung gibt, können Sie `Api::BuiltinTypeId::Dynamic` verwenden.

Note

Der Wert des `FieldIndex` Index muss größer als 0 sein. Der Wert 0 ist für den Indexschlüssel reserviert (siehe `StoreEntityIndexKey()`).

Example Beispiel für die Verwendung primitiver Datentypen

```

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 Beispiel mit einem struct um die Daten zu speichern

```

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);
}

```

Die Felddaten der entfernten Entitäten werden geladen

Sie können keine Entitätsfelddaten für Entitäten laden (aus der State Fabric lesen), die aus dem Besitz- und Abonnementbereich der App entfernt wurden. Das folgende Beispiel führt zu einem Fehler, da es eine Entität als Ergebnis eines aufruft `Api::LoadIndexKey()` `Api::ChangeListAction::Remove`. Das zweite Beispiel zeigt eine korrekte Methode zum Speichern und Laden von Entitätsdaten direkt in der App.

Example Beispiel für falschen Code

```
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 Beispiel für eine korrekte Methode zum Speichern und Laden von Entitätsdaten in der App

```

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;
    }
}
}

/* ... */
}

```

Speichert die Position einer Entität

Sie können die Position einer Entität mithilfe einer Integer-Datenstruktur speichern (in die State Fabric schreiben). In diesen Beispielen wird die folgende Funktion verwendet:

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

Note

Sie `Api::BuiltinTypeId::Vector3F32` müssen angeben `Api::StoreEntityIndexKey()`, wie in den folgenden Beispielen gezeigt.

Example Beispiel für die Verwendung eines Arrays zur Darstellung der Position

```
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 Beispiel mit einem struct um die Position darzustellen

```
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));  
}
```

Lädt die Position einer Entität

Sie können die Position einer Entität mithilfe einer Integer-Datenstruktur laden (aus der State Fabric lesen). In diesen Beispielen wird die folgende Funktion verwendet:

Note

Sie `Api::BuiltinTypeId::Vector3F32` müssen angeben `Api::LoadEntityIndexKey()`, wie in den folgenden Beispielen gezeigt.

Example Beispiel für die Verwendung eines Arrays zur Darstellung der Position

```
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 Beispiel mit einem struct um die Position darzustellen

```

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);
}

```

Ereignisse der Entität

Sie können die folgenden Funktionen im SimSpace Weaver App-SDK verwenden, um alle Eigentums- und Abonnementereignisse abzurufen:

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

Sie können das SimSpace Weaver Demo-Framework verwenden, wenn Sie eine rückrufgesteuerte Verarbeitung von Entitätsereignissen benötigen. Weitere Informationen finden Sie in der folgenden Header-Datei:

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

Sie können auch Ihre eigene Entitäts-Eventverarbeitung erstellen.

Themen

- [Iterieren Sie Ereignisse für eigene Entitäten](#)
- [Iterieren Sie Ereignisse für abonnierte Entitäten](#)
- [Führen Sie Iterationen durch Ereignisse zum Eigentümerwechsel für Entitäten durch](#)

Iterieren Sie Ereignisse für eigene Entitäten

Wird verwendet `OwnershipChanges()`, um eine Liste von Ereignissen für eigene Entitäten (Entitäten im Eigentumsbereich der App) abzurufen. Die Funktion hat die folgende Signatur:

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

Iterieren Sie dann mit einer Schleife durch die Entitäten, wie im folgenden Beispiel gezeigt.

Example Beispiel

```
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;
}
}
```

Ereignistypen

- None— Die Entität befindet sich in dem Bereich und ihre Positions- und Felddaten wurden nicht geändert.
- Remove— Die Entität wurde aus dem Bereich entfernt.
- Add— Die Entität wurde dem Bereich hinzugefügt.
- Update— Die Entität befindet sich in dem Bereich und wurde geändert.
- Reject— Die App konnte die Entität nicht aus dem Bereich entfernen.

Note

Im Falle eines `Reject` Ereignisses versucht die App beim nächsten Häkchen erneut, die Übertragung durchzuführen.

Iterieren Sie Ereignisse für abonnierte Entitäten

Wird verwendet `AllSubscriptionEvents()`, um eine Liste von Ereignissen für abonnierte Entitäten (Entitäten im Abonnementbereich der App) abzurufen. Die Funktion hat die folgende Signatur:

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

Iterieren Sie dann mit einer Schleife durch die Entitäten, wie im folgenden Beispiel gezeigt.

Example Beispiel

```
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;
    }
}
```

Ereignistypen

- None— Die Entität befindet sich in dem Bereich und ihre Positions- und Felddaten wurden nicht geändert.
- Remove— Die Entität wurde aus dem Bereich entfernt.
- Add— Die Entität wurde dem Bereich hinzugefügt.
- Update— Die Entität befindet sich in dem Bereich und wurde geändert.
- Reject— Die App konnte die Entität nicht aus dem Bereich entfernen.

Note

Im Falle eines Reject Ereignisses versucht die App beim nächsten Häkchen erneut, die Übertragung durchzuführen.

Führen Sie Iterationen durch Ereignisse zum Eigentümerwechsel für Entitäten durch

Um Ereignisse zu ermitteln, bei denen eine Entität zwischen einem Eigentümerbereich und einem Abonnementbereich wechselt, vergleichen Sie die Änderungen zwischen den aktuellen und früheren Eigentümerschafts- und Abonnementereignissen der Entität.

Sie können mit diesen Ereignissen umgehen, indem Sie Folgendes lesen:

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

Anschließend können Sie die Änderungen mit zuvor gespeicherten Daten vergleichen.

Das folgende Beispiel zeigt, wie Sie mit Ereignissen umgehen können, bei denen sich die Eigentümerschaft einer Entität ändert. In diesem Beispiel wird davon ausgegangen, dass bei Entitäten, die zwischen abonnierten Entitäten und eigenen Entitäten (in beide Richtungen) wechseln, das `remove/add event occurs first followed by the subscription remove/add` Eigentümerereignis im nächsten Tick stattfindet.

Example Beispiel

```
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 = entityIdByOwnershipAction;

  return Success();
}

```

Result und Fehlerbehandlung

Die `Aws::WeaverRuntime::Result<T>` Klasse verwendet eine `Outcome` Bibliothek eines Drittanbieters. Sie können das folgende Muster verwenden, um die von API-Aufrufen zurückgegebenen Fehler zu überprüfen `Result` und abzufangen.

```

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 Steueranweisungs-Makro

In einer Funktion mit einem Rückgabety `Aws::WeaverRuntime::Result<T>` können Sie das `WEAVERRUNTIME_TRY` Makro anstelle des vorherigen Codemusters verwenden. Das Makro führt die ihm übergebene Funktion aus. Wenn die übergebene Funktion fehlschlägt, veranlasst das Makro, dass die umschließende Funktion einen Fehler zurückgibt. Wenn die übergebene Funktion erfolgreich ist, wird die Ausführung mit der nächsten Zeile fortgesetzt. Das folgende Beispiel zeigt eine Neuschreibung der vorherigen Funktion. `DoBeginUpdate()` Diese Version verwendet das `WEAVERRUNTIME_TRY` Makro anstelle von `if-else` Kontrollstruktur. Beachten Sie, dass der Rückgabety `void` der Funktion ist `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();
}

```

Wenn ein `BeginUpdate()` Fehler auftritt, `DoBeginUpdate()` kehrt das Makro mit einem Fehler vorzeitig zurück. Sie können das `WEAVERRUNTIME_EXPECT_ERROR` Makro verwenden, um das `Aws::WeaverRuntime::ErrorCode` von `abzurufenBeginUpdate()`. Das folgende Beispiel zeigt, wie die `Update()` Funktion bei einem Fehler den Fehlercode aufruft `DoBeginUpdate()` und abrufft.

```

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);
}
}
```

Sie können den Fehlercode von `BeginUpdate()` available in eine Funktion umwandeln, die sie aufruft, `Update()` indem Sie den Rückgabety `Update()` auf `ändernAws::WeaverRuntime::Result<void>`. Sie können diesen Vorgang wiederholen, um den Fehlercode weiter unten in der Aufrufliste zu senden.

Generika und Domaintypen

Das SimSpace Weaver App-SDK stellt die Datentypen mit einfacher Genauigkeit `Api::Vector2F32` und `Api::BoundingBox2F32` die Datentypen mit doppelter `Api::Vector2F64` Genauigkeit und bereit. `Api::BoundingBox2F64` Bei diesen Datentypen handelt es sich um passive Datenstrukturen ohne praktische Methoden. Beachten Sie, dass die API nur `Api::Vector2F32` und verwendet `Api::BoundingBox2F32`. Sie können diese Datentypen verwenden, um Abonnements zu erstellen und zu ändern.

Das SimSpace Weaver Demo-Framework bietet eine Minimalversion von AzCore Mathe-Bibliothek, die `Vector3` und enthält `Aabb`. Weitere Informationen finden Sie in den Header-Dateien unter:

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

Verschiedene App-SDK-Operationen

Themen

- [AllSubscriptionEvents and OwnershipChanges enthalten Ereignisse aus dem letzten Aufruf](#)

- [Löst Lesesperren nach der Verarbeitung SubscriptionChangeList](#)
- [Erstellen Sie eine eigenständige App-Instanz zum Testen](#)

AllSubscriptionEvents and OwnershipChanges enthalten Ereignisse aus dem letzten Aufruf

Die Rückgabewerte von Aufrufen `Api::AllSubscriptionEvents()` und `Api::OwnershipChanges()` enthalten Ereignisse des letzten Aufrufs, nicht des letzten Ticks. Im folgenden Beispiel `secondOwnershipChangeList` sind sie leer, `secondSubscriptionEvents` weil ihre Funktionen unmittelbar nach den ersten Aufrufen aufgerufen werden.

Wenn Sie 10 Ticks warten und dann `Api::AllSubscriptionEvents()` und `aufrufenApi::OwnershipChanges()`, enthalten ihre Ergebnisse sowohl Ereignisse als auch Änderungen der letzten 10 Ticks (nicht des letzten Ticks).

Example Beispiel

```
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

Die Funktion `AllSubscriptionEvents()` ist implementiert, aber die Funktion `SubscriptionEvents()` ist nicht implementiert.

Löst Lesesperren nach der Verarbeitung `SubscriptionChangeList`

Wenn Sie mit einem Update beginnen, gibt es gemeinsame Speichersegmente für die übergebenen Daten in anderen Partitionen für den vorherigen Tick. Diese gemeinsam genutzten Speichersegmente werden möglicherweise von Lesern gesperrt. Eine App kann erst dann vollständig festgeschrieben werden, wenn alle Leser die Sperren aufgehoben haben. Zur Optimierung sollte eine App aufrufen, um die Sperren `Api::ReleaseReadLeases()` nach der Verarbeitung von `Api::SubscriptionChangeList` Elementen aufzuheben. Dadurch werden Konflikte beim Festschreiben reduziert. `Api::Commit()` veröffentlicht die Read-Leases standardmäßig, aber es hat sich bewährt, sie nach der Verarbeitung von Abonnement-Updates manuell freizugeben.

Example Beispiel

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

    /**
     * Done processing Api::SubscriptionChangeList items.
     * Release read locks.
     */

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Erstellen Sie eine eigenständige App-Instanz zum Testen

Sie können `Api::CreateStandaloneApplication()` damit eine eigenständige App erstellen, um die Anwendungslogik zu testen, bevor Sie den Code in einer tatsächlichen Simulation ausführen.

Example Beispiel

```
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 Demo-Framework

Das AWS SimSpace Weaver Demo-Framework (Demo-Framework) ist eine Bibliothek von Hilfsprogrammen, mit denen Sie SimSpace Weaver Apps entwickeln können.

Das Demo-Framework bietet

- Codebeispiele und Programmiermuster, die Sie verwenden und untersuchen können
- Abstraktionen und Hilfsfunktionen, die die Entwicklung einfacher Apps vereinfachen
- Eine einfachere Möglichkeit, experimentelle Funktionen des App-SDK zu testen SimSpace Weaver

Wir haben das SimSpace Weaver App-SDK mit einfachem Zugriff entwickelt, SimSpace Weaver APIs um eine höhere Leistung zu erzielen. Im Gegensatz dazu haben wir das Demo-Framework so konzipiert, dass es Abstraktionen auf höherer Ebene und den Zugriff darauf bietet, was APIs die Bedienung erleichtert SimSpace Weaver . Der Preis für die Benutzerfreundlichkeit ist ein geringeres Leistungsniveau als bei der direkten Verwendung des SimSpace Weaver App-SDK. Simulationen, die eine geringere Leistung tolerieren können (z. B. solche ohne Echtzeit-Leistungsanforderungen), könnten sich für die Verwendung des Demo-Frameworks eignen. Wir empfehlen, die systemeigenen Funktionen des SimSpace Weaver App-SDK für komplexe Anwendungen zu verwenden, da das Demo-Framework kein vollständiges Toolkit ist.

Das Demo-Framework umfasst

- Funktionierende Codebeispiele, die Folgendes unterstützen und demonstrieren:
 - Verwaltung des Anwendungsflusses
 - Verarbeitung von Entitätsereignissen durch Rückrufe
- Eine Reihe von Dienstprogrammibliotheken von Drittanbietern:

- spdlog (eine Logging-Bibliothek)
- Eine Minimalversion von AZCore (eine Mathe-Bibliothek), die nur Folgendes enthält:
 - Vector3
 - Aabb
- cxxopts (eine Parser-Bibliothek für Befehlszeilenoptionen)
- Spezifische Hilfsfunktionen für SimSpace Weaver

Das Demo-Framework besteht aus einer Bibliothek, Quelldateien und CMakeLists. Die Dateien sind im verteilbaren SimSpace Weaver App-SDK-Paket enthalten.

Mit Servicekontingenten arbeiten

In diesem Abschnitt wird beschrieben, wie Sie mit den Servicekontingenten für arbeiten SimSpace Weaver. Kontingente werden auch als Limits bezeichnet. Eine Liste der Dienstkontingente finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#). Die APIs in diesem Abschnitt enthaltenen Daten stammen aus dem App-Set APIs. Apps APIs unterscheiden sich von dem Service APIs. Die Apps APIs sind Teil des SimSpace Weaver App-SDK. Sie finden die Dokumentation für die App APIs im App-SDK-Ordner auf Ihrem lokalen System:

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

Themen

- [Holen Sie sich die Grenzwerte für eine App](#)
- [Ruft die Menge der von einer App verwendeten Ressourcen ab](#)
- [Metriken zurücksetzen](#)
- [Überschreitung eines Limits](#)
- [Der Speicher geht zur Neige](#)
- [Bewährte Methoden](#)

Holen Sie sich die Grenzwerte für eine App

Sie können das RuntimeLimits App-API zur Abfrage der Grenzwerte für eine App.

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

Parameter

Application und App

Ein Verweis auf die App.

LimitType Typ

Eine Aufzählung mit den folgenden Limittypen:

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

Im folgenden Beispiel wird das Limit für die Anzahl der Entitäten abgefragt.

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

Ruft die Menge der von einer App verwendeten Ressourcen ab

Sie können den anrufen RuntimeMetrics App-API, um die Menge der von einer App verwendeten Ressourcen abzurufen:

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

Parameter

Application und App

Ein Verweis auf die App.

Die API gibt einen Verweis auf a zurück struct das enthält die Metriken. Eine Zählermetrik enthält einen laufenden Gesamtwert und nimmt nur zu. Eine Messmetrik enthält einen Wert, der steigen oder sinken kann. Die Anwendungslaufzeit aktualisiert einen Zähler, wenn ein Ereignis den Wert

erhöht. Die Runtime aktualisiert die Messgeräte nur, wenn Sie die API aufrufen. SimSpace Weaver garantiert, dass die Referenz für die gesamte Lebensdauer der App gültig ist. Wiederholte Aufrufe der API ändern die Referenz nicht.

```
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
}
```

Metriken zurücksetzen

Das Tool `ResetRuntimeMetrics` Die App-API setzt die Werte in der zurück `AppRuntimeMetrics` struct.

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

Das folgende Beispiel zeigt, wie Sie aufrufen können `ResetRuntimeMetrics` in deiner App.

```
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));
}
```

Überschreitung eines Limits

Bei einem App-API-Aufruf, der ein Limit überschreitet, wird ein zurückgegeben `ErrorCode::CapacityExceeded`, außer bei Entitätsübertragungen. SimSpace Weaver verarbeitet Entitätsübertragungen asynchron als Teil von Commit und BeginUpdate App-API-Operationen, sodass es keinen bestimmten Vorgang gibt, der einen Fehler zurückgibt, wenn eine Übertragung aufgrund des Limits für die Übertragung von Entitäten fehlschlägt. Um Übertragungsfehler zu erkennen, können Sie die aktuellen Werte von `rejected_incoming_transfer_counter` und vergleichen `rejected_outgoing_transfer_counter` (in `AppRuntimeMetrics` struct) mit ihren vorherigen Werten. Abgelehnte Entitäten werden sich nicht in der Partition befinden, aber die App kann sie trotzdem simulieren.

Der Speicher geht zur Neige

SimSpace Weaver verwendet einen Garbage-Collector-Prozess, um freigegebenen Speicher zu bereinigen und freizugeben. Es ist möglich, Daten schneller zu schreiben, als der Garbage-Collector Speicher freigeben kann. In diesem Fall können Schreibvorgänge das für die App reservierte Speicherlimit überschreiten. SimSpace Weaver gibt einen internen Fehler mit einer Meldung zurück, die `OutOfMemory` (und zusätzliche Details) enthält. Weitere Informationen finden Sie unter [Schreibvorgänge über einen bestimmten Zeitraum verteilen](#).

Bewährte Methoden

Die folgenden bewährten Methoden sind allgemeine Richtlinien für die Gestaltung Ihrer Apps, um zu vermeiden, dass Grenzwerte überschritten werden. Sie gelten möglicherweise nicht für Ihr spezifisches App-Design.

Überwachen Sie häufig und verlangsamen Sie die Geschwindigkeit

Sie sollten Ihre Messwerte häufig überwachen und Abläufe, die kurz vor dem Erreichen eines Grenzwerts stehen, verlangsamen.

Vermeiden Sie es, die Abonnement- und Übertragungslimits zu überschreiten

Wenn möglich, sollten Sie Ihre Simulation so gestalten, dass die Anzahl der Fernabonnements und Objektübertragungen reduziert wird. Sie können Platzierungsgruppen verwenden, um mehrere Partitionen auf demselben Worker zu platzieren und so die Notwendigkeit von Entitätsübertragungen zwischen Workern aus der Ferne zu reduzieren.

Schreibvorgänge über einen bestimmten Zeitraum verteilen

Die Anzahl und Größe der Aktualisierungen in einem Tick können erhebliche Auswirkungen auf den Zeit- und Speicherbedarf haben, der für die Durchführung einer Transaktion erforderlich ist. Große Speicheranforderungen können dazu führen, dass der Laufzeit der Anwendung nicht mehr genügend Arbeitsspeicher zur Verfügung steht. Sie können Schreibvorgänge über einen längeren Zeitraum verteilen, um die durchschnittliche Gesamtgröße der Updates pro Tick zu verringern. Dies kann dazu beitragen, die Leistung zu verbessern und zu verhindern, dass Grenzwerte überschritten werden. Wir empfehlen, dass Sie nicht mehr als durchschnittlich 12 MB pro Tick oder 1,5 KB für jede Entität schreiben.

Simulationen debuggen

Sie können die folgenden Methoden verwenden, um Informationen zu Ihren Simulationen zu erhalten.

Themen

- [Verwenden Sie SimSpace Weaver Local und schauen Sie sich die Konsolenausgabe an](#)
- [Sehen Sie sich Ihre Logs in Amazon CloudWatch Logs an](#)
- [Verwenden Sie Beschreiben API-Aufrufe](#)
- [Einen Client Connect](#)

Verwenden Sie SimSpace Weaver Local und schauen Sie sich die Konsolenausgabe an

Wir empfehlen, dass Sie Ihre Simulationen zuerst lokal entwickeln und sie dann in der ausführen AWS Cloud. Sie können die Konsolenausgabe direkt anzeigen, wenn Sie mit SimSpace Weaver Local. Weitere Informationen finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).

Sehen Sie sich Ihre Logs in Amazon CloudWatch Logs an

Wenn Sie Ihre Simulation in AWS Cloud der Konsole ausführen, wird die Ausgabe Ihrer Apps an Protokollstreams in Amazon CloudWatch Logs gesendet. Ihre Simulation schreibt auch andere Protokolldaten. Sie müssen die Protokollierung in Ihrem Simulationsschema aktivieren, wenn Sie möchten, dass Ihre Simulation Protokolldaten schreibt. Weitere Informationen finden Sie unter [SimSpace Weaver loggt sich in Amazon CloudWatch Logs ein](#).

⚠ Warning

Ihre Simulation kann große Mengen an Protokolldaten erzeugen. Die Protokolldaten können sehr schnell anwachsen. Sie sollten Ihre Protokolle genau beobachten und Ihre Simulationen beenden, wenn Sie sie nicht mehr benötigen. Die Protokollierung kann hohe Kosten verursachen.

Verwenden Sie Beschreiben API-Aufrufe

Sie können den folgenden Service nutzen APIs , um Informationen zu Ihren Simulationen in der zu erhalten AWS Cloud.

- ListSimulations— Holen Sie sich eine Liste all Ihrer Simulationen in der AWS Cloud.

Example Beispiel

```
aws simspaceweaver list-simulations
```

- DescribeSimulation— Details zu einer Simulation abrufen.

Example Beispiel

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

- DescribeApp— Details zu einer App abrufen.

Example Beispiel

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

Weitere Informationen über die finden SimSpace Weaver APIs Sie unter [SimSpace Weaver API-Referenzen](#).

Einen Client Connect

Sie können einen Client mit einer laufenden benutzerdefinierten App oder einer Service-App verbinden, die Sie mit einem `endpoint_config` in Ihrem Simulationsschema definiert haben. Das

SimSpace Weaver App-SDK enthält Beispielclients, mit denen Sie die Beispielanwendung anzeigen können. Sie können sich den Quellcode für diese Beispielclients und die Beispielanwendung ansehen, um zu sehen, wie Sie Ihre eigenen Clients erstellen können. Weitere Informationen zum Erstellen und Ausführen der Beispielclients finden Sie in den Tutorials unter [Erste Schritte mit SimSpace Weaver](#).

Den Quellcode für die Beispielclients finden Sie im folgenden Ordner:

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

Debuggen lokaler Simulationen

Sie können Ihre SimSpace Weaver Local Apps mit Microsoft Visual Studio debuggen. [Weitere Informationen zum Debuggen mit finden Sie Visual Studio unter. Microsoft Visual Studio documentation](#)

So debuggen Sie Ihre lokale Simulation

1. Stellen Sie sicher, dass Sie `schema.yaml` sich in Ihrem Arbeitsverzeichnis befinden.
2. Öffnen Sie in das Kontextmenü für jede App Visual Studio, die Sie debuggen möchten (z. B. `PathfindingSampleLocalSpatial` oder `PathfindingSampleLocalView`), und legen Sie das Arbeitsverzeichnis im Debugging-Bereich fest.
3. Öffnen Sie das Kontextmenü für die App, die Sie debuggen möchten, und wählen Sie Als Startprojekt festlegen aus.
4. Wählen Sie `F5`, ob Sie mit dem Debuggen der App beginnen möchten.

Die Anforderungen für das Debuggen einer Simulation entsprechen den Anforderungen für die normale Ausführung einer Simulation. Sie müssen mit der Anzahl von Geo-Apps beginnen, die im Schema angegeben ist. Wenn Ihr Schema beispielsweise ein 2x2-Raster angibt und Sie eine Geo-App im Debug-Modus starten, wird die Simulation erst ausgeführt, wenn Sie 3 weitere Spatial-Apps starten (im Debug-Modus oder nicht im Debug-Modus).

Um eine benutzerdefinierte App zu debuggen, müssen Sie zuerst Ihre Geo-Apps und dann die benutzerdefinierte App im Debugger starten.

Beachten Sie, dass Ihre Simulation im Lockstep ausgeführt wird. Sobald eine App einen Breakpoint erreicht, werden alle anderen Apps pausiert. Wenn Sie von diesem Breakpoint aus weitermachen, werden die anderen Apps fortgesetzt.

Maßgeschneiderte Behälter

AWS SimSpace Weaver Apps werden in Containern ausgeführt Amazon Linux 2 (AL2) Umgebungen. In der werden Ihre Simulationen in Docker-Containern SimSpace Weaver ausgeführt AWS Cloud, die auf einem `amazonlinux:2` Image basieren, das von Amazon Elastic Container Registry (Amazon ECR) bereitgestellt wird. Sie können ein benutzerdefiniertes Docker-Image erstellen, es in Amazon ECR speichern und dieses Image anstelle des von uns bereitgestellten Standard-Docker-Images für Ihre Simulation verwenden.

Sie können einen benutzerdefinierten Container verwenden, um Ihre Softwareabhängigkeiten zu verwalten und zusätzliche Softwarekomponenten einzubeziehen, die nicht im Standard-Docker-Image enthalten sind. Sie können beispielsweise die öffentlich verfügbaren Softwarebibliotheken, die Ihre App verwendet, zum Container hinzufügen und nur Ihren benutzerdefinierten Code in die Zip-Datei der App einfügen.

Important

Wir unterstützen nur AL2 Docker-Images, die in Amazon ECR-Repositoryys gehostet werden, entweder in der Amazon ECR Public Gallery oder in Ihrer privaten Amazon ECR-Registrierung. Wir unterstützen keine Docker-Images, die außerhalb von Amazon ECR gehostet werden. Weitere Informationen zu Amazon ECR finden Sie in der [Dokumentation zu Amazon Elastic Container Registry](#).

Themen

- [Erstellen Sie einen benutzerdefinierten Container](#)
- [Ändern Sie ein Projekt, um einen benutzerdefinierten Container zu verwenden](#)
- [Häufig gestellte Fragen zu benutzerdefinierten Containern](#)
- [Problembehandlung bei benutzerdefinierten Containern](#)

Erstellen Sie einen benutzerdefinierten Container

Bei diesen Anweisungen wird davon ausgegangen, dass Sie wissen, wie Docker und Amazon Elastic Container Registry (Amazon ECR) verwendet werden. Weitere Informationen zur Amazon ECR finden Sie unter Sicherheit im [Amazon-ECR-Benutzerhandbuch](#).

Voraussetzungen

- Die IAM-Identität (Verwendung oder Rolle), die Sie für diese Aktionen verwenden, verfügt über die richtigen Berechtigungen zur Verwendung von Amazon ECR
- Docker ist auf Ihrem lokalen System installiert

Um einen benutzerdefinierten Container zu erstellen

1. Erstellen Sie Ihr Dockerfile.

A Dockerfile zum Ausführen AWS SimSpace Weaver von Apps beginnt mit Amazon Linux 2 Bild in Amazon ECR.

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

2. Baue deinDockerfile.
3. Laden Sie Ihr Container-Image auf Amazon ECR hoch.
 - [Verwenden Sie die AWS Management Console.](#)
 - [Benutze die AWS Command Line Interface.](#)

Note

Wenn Sie beim Versuch, Ihr Container-Image auf Amazon ECR hochzuladen, eine `AccessDeniedException` Fehlermeldung erhalten, verfügt Ihre IAM-Identität (Benutzer oder Rolle) möglicherweise nicht über die erforderlichen Berechtigungen zur Verwendung von Amazon ECR. Sie können die `AmazonEC2ContainerRegistryPowerUser` AWS verwaltete Richtlinie an Ihre IAM-Identität anhängen und es erneut versuchen. Weitere Informationen zum Anhängen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im AWS Identity and Access Management Benutzerhandbuch.

Ändern Sie ein Projekt, um einen benutzerdefinierten Container zu verwenden

Bei diesen Anweisungen wird davon ausgegangen, dass Sie bereits wissen, wie Sie Ihre App verwenden AWS SimSpace Weaver können, und dass Sie Ihre Workflows für App-Speicher und Entwicklung AWS Cloud effizienter gestalten möchten.

Voraussetzungen

- Sie haben einen benutzerdefinierten Container in Amazon Elastic Container Registry (Amazon ECR). Weitere Informationen zum Erstellen eines benutzerdefinierten Containers finden Sie unter [Erstellen Sie einen benutzerdefinierten Container](#).

So ändern Sie Ihr Projekt so, dass es einen benutzerdefinierten Container verwendet

1. Fügen Sie der Simulations-App-Rolle Ihres Projekts Berechtigungen zur Verwendung von Amazon ECR hinzu.
 - a. Wenn Sie noch keine IAM-Richtlinie mit den folgenden Berechtigungen haben, erstellen Sie die Richtlinie. Wir empfehlen den Namen `simspaceweaver-ecr` der Richtlinie. Weitere Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im AWS Identity and Access Management Benutzerhandbuch.

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

- b. Suchen Sie nach dem Namen der Simulations-App-Rolle Ihres Projekts:

- i. Öffnen Sie die AWS CloudFormation Vorlage in einem Texteditor:

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

- ii. Suchen Sie die roleName Immobilie unterWeaverAppRole. Der Wert ist der Name der Simulations-App-Rolle Ihres Projekts.

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. Ordnen Sie die `simspaceweaver-ecr` Richtlinie der Simulations-App-Rolle des Projekts zu. Weitere Informationen zum Anhängen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im AWS Identity and Access Management Benutzerhandbuch.
- d. Navigieren Sie zu dem folgenden Befehl `sdk-folder` und führen Sie ihn aus, um den SimSpace Weaver Beispielstapel zu aktualisieren:

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

2. Geben Sie Ihre Container-Images im Simulationsschema des Projekts an.

- Sie können die optionale `default_image` Eigenschaft unter `hinzufügensimulation_properties`, um ein benutzerdefiniertes Standard-Container-Image für alle Domänen anzugeben.
- Fügen Sie die `image` Eigenschaft `app_config` für eine Domain hinzu, für die Sie ein benutzerdefiniertes Container-Image verwenden möchten. Geben Sie die Amazon ECR-Repository-URI als Wert an. Sie können für jede Domain ein anderes Bild angeben.

- Wenn es `image` nicht für eine Domain angegeben `default_image` ist, aber angegeben ist, verwenden Apps in dieser Domain das Standardbild.
- Wenn es `image` nicht für eine Domain angegeben und auch `default_image` nicht angegeben ist, werden Apps in dieser Domain in einem SimSpace Weaver Standardcontainer ausgeführt.

Example Schemasnippet, das benutzerdefinierte Container-Einstellungen enthält

```
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. Erstellen Sie Ihr Projekt und laden Sie es wie gewohnt hoch.

Häufig gestellte Fragen zu benutzerdefinierten Containern

F1. Was mache ich, wenn ich den Inhalt meines Containers ändern möchte?

- Für eine laufende Simulation — Sie können den Container nicht für eine laufende Simulation ändern. Sie müssen einen neuen Container erstellen und eine neue Simulation starten, die diesen Container verwendet.
- Für eine neue Simulation — Erstellen Sie einen neuen Container, laden Sie ihn in Amazon Elastic Container Registry (Amazon ECR) hoch und starten Sie eine neue Simulation, die diesen Container verwendet.

F2. Wie kann ich das Container-Image für meine Simulation ändern?

- Für eine laufende Simulation — Sie können den Container für eine laufende Simulation nicht ändern. Sie müssen eine neue Simulation starten, die den neuen Container verwendet.
- Für eine neue Simulation — Geben Sie das neue Container-Image im Simulationsschema Ihres Projekts an. Weitere Informationen finden Sie unter [Ändern Sie ein Projekt, um einen benutzerdefinierten Container zu verwenden](#).

Problembehandlung bei benutzerdefinierten Containern

Themen

- [AccessDeniedException wenn Sie Ihr Bild in Amazon Elastic Container Registry \(Amazon ECR\) hochladen](#)
- [Eine Simulation, die einen benutzerdefinierten Container verwendet, kann nicht gestartet werden](#)

AccessDeniedException wenn Sie Ihr Bild in Amazon Elastic Container Registry (Amazon ECR) hochladen

Wenn Sie beim Versuch, Ihr Container-Image auf Amazon ECR hochzuladen, eine `AccessDeniedException` Fehlermeldung erhalten, verfügt Ihre IAM-Identität (Benutzer oder Rolle) möglicherweise nicht über die erforderlichen Berechtigungen zur Verwendung von Amazon ECR. Sie können die `AmazonEC2ContainerRegistryPowerUser` AWS verwaltete Richtlinie an Ihre IAM-

Identität anhängen und es erneut versuchen. Weitere Informationen zum Anhängen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im AWS Identity and Access Management Benutzerhandbuch.

Eine Simulation, die einen benutzerdefinierten Container verwendet, kann nicht gestartet werden

Tipps zur Problembhebung

- Wenn die Protokollierung für Ihre Simulation aktiviert ist, überprüfen Sie Ihre Fehlerprotokolle.
- Testen Sie Ihre Simulation ohne einen benutzerdefinierten Container.
- Testen Sie Ihre Simulation lokal. Weitere Informationen finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).

Verwenden von Python

Sie können Python für Ihre SimSpace Weaver Apps und Ihren Client verwenden. Das Python Software Development Kit (Python SDK) ist Teil des verteilbaren Standardpakets des SimSpace Weaver App-SDK. Die Entwicklung mit Python funktioniert ähnlich wie die Entwicklung in den anderen unterstützten Sprachen.

Important

SimSpace Weaver unterstützt nur Python-Version 3.9.

Important

SimSpace Weaver Für die Unterstützung von Python ist SimSpace Weaver Version 1.15.0 oder höher erforderlich.

Themen

- [Ein Python-Projekt erstellen](#)
- [Eine Python-Simulation starten](#)
- [Der Python-Beispielclient](#)

- [Häufig gestellte Fragen zur Verwendung von Python](#)
- [Behebung von Problemen im Zusammenhang mit Python](#)

Ein Python-Projekt erstellen

Benutzerdefinierter Python-Container

Um Ihre Python-basierte SimSpace Weaver Simulation in der auszuführen AWS Cloud, können Sie einen benutzerdefinierten Container erstellen, der die erforderlichen Abhängigkeiten enthält. Weitere Informationen finden Sie unter [Maßgeschneiderte Behälter](#).

Ein benutzerdefinierter Python-Container muss Folgendes enthalten:

- gcc
- openssl-devel
- bzip2-Entwicklung
- libffi-entwickeln
- wget
- tar
- gzip
- make
- Python (Version 3.9)

Wenn Sie die PythonBubblesSample Vorlage verwenden, um Ihr Projekt zu erstellen, können Sie das `quick-start.py` Skript (das sich im `tools` Ordner Ihres Projekts befindet) ausführen, um ein Docker-Image mit den erforderlichen Abhängigkeiten zu erstellen. Das Skript lädt das Bild in Amazon Elastic Container Registry (Amazon ECR) hoch.

Das `quick-start.py` Skript verwendet Folgendes: `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
```

Sie können Ihre eigenen Abhängigkeiten hinzufügen zu `Dockerfile`:

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

Die `requirements.txt` Datei enthält eine Liste von Python-Paketen, die für die `PythonBubblesSample` Beispielsimulation benötigt werden:

```
Flask==2.1.1
```

Sie können Ihre eigenen Python-Paketabhängigkeiten hinzufügen zu `requirements.txt`:

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

Die `Dockerfile` und `requirements.txt` befinden sich im `tools` Ordner Ihres Projekts.

Important

Sie müssen technisch gesehen keinen benutzerdefinierten Container für Ihre Python-Simulation verwenden, aber wir empfehlen dringend, einen benutzerdefinierten Container zu verwenden. Der von uns bereitgestellte Standardcontainer für Amazon Linux 2 (AL2) enthält kein Python. Wenn Sie also keinen benutzerdefinierten Container mit Python verwenden, müssen Sie Python und die erforderlichen Abhängigkeiten in jede App-ZIP-Datei aufnehmen, in die Sie hochladen SimSpace Weaver.

Eine Python-Simulation starten

Sie können Ihre Python-basierte Simulation genauso starten wie eine normale SimSpace Weaver Simulation, sowohl in SimSpace Weaver Local und in der SimSpace Weaver . AWS Cloud Weitere Informationen finden Sie in den Tutorials unter [Erste Schritte mit SimSpace Weaver](#).

Das `PythonBubblesSample` beinhaltet einen eigenen Python-Beispielclient. Weitere Informationen finden Sie unter [Der Python-Beispielclient](#).

Der Python-Beispielclient

Wenn Sie die `PythonBubblesSample` Vorlage verwenden, um ein Projekt zu erstellen, enthält Ihr Projekt einen Python-Beispielclient. Sie können den Beispielclient verwenden, um die `PythonBubblesSample` Simulation anzusehen. Sie können den Beispielclient auch als Ausgangspunkt verwenden, um Ihren eigenen Python-Client zu erstellen.

Beim folgenden Verfahren wird davon ausgegangen, dass Sie ein `PythonBubblesSample` Projekt erstellt und dessen Simulation gestartet haben.

Um den Python-Client zu starten

1. Gehen Sie in einem Befehlszeilenfenster zum `PyBubbleClient` Beispielprojektordner.

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

2. Führen Sie den Python-Client aus.

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

Parameter

host

Die IP-Adresse Ihrer Simulation. Für eine Simulation AWS Cloud, die in gestartet wurde, können Sie die IP-Adresse Ihrer Simulation in der [SimSpace Weaver Konsole](#) finden oder das Verfahren im [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#) Schnellstart-Tutorial verwenden. Verwenden Sie `127.0.0.1` für eine lokale Simulation die IP-Adresse.

port

Die Portnummer Ihrer Simulation. Für eine Simulation AWS Cloud, die in gestartet wurde, ist dies die `Actual` Portnummer. Sie finden die Portnummer Ihrer Simulation in der [SimSpace Weaver Konsole](#) oder verwenden Sie das Verfahren im [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#) Schnellstart-Tutorial. Verwenden Sie für eine lokale Simulation `7000` die Portnummer.

simsizes

Die maximale Anzahl von Entitäten, die im Client angezeigt werden sollen.

Häufig gestellte Fragen zur Verwendung von Python

F1. Welche Versionen von Python werden unterstützt?

SimSpace Weaver unterstützt nur Python-Version 3.9.

Behebung von Problemen im Zusammenhang mit Python

Themen

- [Fehler bei der Erstellung eines benutzerdefinierten Containers](#)
- [Ihre Python-Simulation kann nicht gestartet werden](#)
- [Ein Python-Simulations- oder View-Client gibt einen ModuleNotFoundError aus](#)

Fehler bei der Erstellung eines benutzerdefinierten Containers

Wenn Sie `no basic auth credentials` nach der Ausführung eine Fehlermeldung erhalten, `quick-start.py` liegt möglicherweise ein Problem mit Ihren temporären Anmeldeinformationen für Amazon ECR vor. Führen Sie den folgenden Befehl mit Ihrer AWS-Region ID und AWS Kontonummer aus:

```
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

Stellen Sie sicher, dass es sich bei dem von AWS-Region Ihnen angegebenen um denselben handelt, den Sie für Ihre Simulation verwenden. Verwenden Sie eine der AWS-Regionen

SimSpace Weaver Unterstützungen. Weitere Informationen finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#).

Nachdem Sie den `aws ecr` Befehl ausgeführt haben, führen Sie ihn `quick-start.py` erneut aus.

Weitere Ressourcen zur Fehlerbehebung, die Sie überprüfen sollten

- [Problembehandlung bei benutzerdefinierten Containern](#)
- [Amazon ECR-Fehlerbehebung](#) im Amazon ECR-Benutzerhandbuch
- [Einrichtung mit Amazon ECR](#) im Amazon ECR-Benutzerhandbuch

Ihre Python-Simulation kann nicht gestartet werden

Möglicherweise wird im Verwaltungsprotokoll Ihrer Simulation ein `Unable to start app` Fehler angezeigt. Dies kann passieren, wenn die Erstellung Ihres benutzerdefinierten Containers fehlgeschlagen ist. Weitere Informationen finden Sie unter [Fehler bei der Erstellung eines benutzerdefinierten Containers](#). Weitere Informationen über Protokolle finden Sie unter [SimSpace Weaver loggt sich in Amazon CloudWatch Logs ein](#).

Wenn Sie sicher sind, dass an Ihrem Container nichts falsch ist, überprüfen Sie den Python-Quellcode Ihrer App. Sie können Folgendes verwenden ... `SimSpace Weaver Local` um deine App zu testen. Weitere Informationen finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).

Ein Python-Simulations- oder View-Client gibt einen `ModuleNotFound` Fehler aus

Python gibt einen `ModuleNotFound` Fehler aus, wenn es ein erforderliches Python-Paket nicht finden kann.

Wenn sich Ihre Simulation in der befindet AWS Cloud, stellen Sie sicher, dass Ihr benutzerdefinierter Container alle erforderlichen Abhängigkeiten enthält, die in Ihrem `requirements.txt` aufgeführt sind. Denken Sie daran, den `quick-start.py` Vorgang bei der Bearbeitung erneut auszuführen `requirements.txt`.

Wenn Sie die Fehlermeldung für den `PythonBubblesSample` Client erhalten, verwenden Sie, `pip` um das angegebene Paket zu installieren:

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

Support für andere Motoren

Sie können Ihre eigenen benutzerdefinierten verwenden C++ Motor mit SimSpace Weaver. Wir entwickeln derzeit Unterstützung für die folgenden Engines. Für jede dieser Engines gibt es eine separate Dokumentation.

Important

Integrationen mit den hier aufgeführten Engines sind experimentell. Sie sind als Vorschau verfügbar.

Motoren

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

Unity

Sie müssen die haben Unity Die Entwicklungsumgebung ist bereits installiert, bevor Sie SimSpace Weaver Simulationen mit Unity erstellen. Weitere Informationen finden Sie in den einzelnen Anweisungen:

`sdk-folder\Unity-Guide.pdf`

Unreal Engine

Sie müssen eine erstellen Unreal Engine dedizierter Server aus dem Quellcode. Das SimSpaceWeaverAppSdkDistributable beinhaltet eine Version des PathfindingSample für Unreal Engine. Weitere Informationen finden Sie in den einzelnen Anweisungen:

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

Verwendung von lizenzierte Software mit AWS SimSpace Weaver

AWS SimSpace Weaver ermöglicht es Ihnen, Simulationen mit der Simulationsengine und den Inhalten Ihrer Wahl zu erstellen. Im Zusammenhang mit Ihrer Nutzung von sind Sie dafür

verantwortlich SimSpace Weaver, die Lizenzbedingungen aller Software oder Inhalte, die Sie in Ihren Simulationen verwenden, zu beschaffen, zu pflegen und einzuhalten. Stellen Sie sicher, dass Ihr Lizenzvertrag Ihnen die Bereitstellung Ihrer Software und Inhalte in einer virtuellen, gehosteten Umgebung ermöglicht.

Verwaltung Ihrer Ressourcen mit AWS CloudFormation

Sie können es verwenden AWS CloudFormation , um Ihre AWS SimSpace Weaver Ressourcen zu verwalten. AWS CloudFormation ist ein separater AWS Dienst, der Sie dabei unterstützt, Ihre AWS Infrastruktur als Code zu spezifizieren, bereitzustellen und zu verwalten. Mit erstellen AWS CloudFormation Sie eine JSON- oder YAML-Datei, die als [Vorlage](#) bezeichnet wird. Ihre Vorlage spezifiziert die Details Ihrer Infrastruktur. AWS CloudFormation verwendet Ihre Vorlage, um Ihre Infrastruktur als eine einzelne Einheit bereitzustellen, die als [Stack](#) bezeichnet wird. Wenn Sie Ihren Stack löschen, können Sie AWS CloudFormation festlegen, dass alles im Stack gleichzeitig gelöscht wird. Sie können Ihre Vorlage mithilfe von Standardprozessen zur Quellcodeverwaltung verwalten (z. B. indem Sie sie in einem Versionskontrollsystem wie [Git](#) verfolgen). Weitere Informationen zu AWS CloudFormation finden Sie im [AWS CloudFormation Benutzerhandbuch](#).

Ihre Simulationsressource

AWS In ist eine Ressource eine Entität, mit der Sie arbeiten können. Beispiele hierfür sind eine EC2 Amazon-Instance, ein Amazon S3-Bucket oder eine IAM-Rolle. Ihre SimSpace Weaver Simulation ist eine Ressource. In Konfigurationen geben Sie normalerweise eine AWS Ressource im Formular `anAWS::service::resource`. Für SimSpace Weaver geben Sie Ihre Simulationsressource als `anAWS::SimSpaceWeaver::Simulation`. Weitere Informationen zu Ihrer Simulationsressource finden Sie im [SimSpace Weaver](#) Abschnitt im AWS CloudFormation Benutzerhandbuch. AWS CloudFormation

Wie kann ich AWS CloudFormation mit verwenden SimSpace Weaver?

Sie können eine AWS CloudFormation Vorlage erstellen, die die AWS Ressourcen angibt, die Sie bereitstellen möchten. Ihre Vorlage kann eine gesamte Architektur, einen Teil einer Architektur oder eine kleine Lösung angeben. Sie könnten beispielsweise eine Architektur für Ihre SimSpace Weaver Lösung angeben, die Amazon S3 S3-Buckets, IAM-Berechtigungen, eine unterstützende Datenbank in Amazon Relational Database Service oder Amazon DynamoDB und Ihre Ressource umfasst. `Simulation` Anschließend können Sie all diese Ressourcen als Einheit und gleichzeitig bereitstellen. AWS CloudFormation

Example Vorlage, die IAM-Ressourcen erstellt und eine Simulation startet

Die folgende Beispielvorlage erstellt eine IAM-Rolle und SimSpace Weaver Berechtigungen, die für die Ausführung von Aktionen in Ihrem Konto erforderlich sind. Die SimSpace Weaver App-SDK-Skripts erstellen die Rolle und die Berechtigungen in einem bestimmten Bereich, AWS-Region wenn Sie ein Projekt erstellen. Sie können jedoch eine AWS CloudFormation Vorlage verwenden, um die Simulation für eine andere bereitzustellen, AWS-Region ohne die Skripts erneut ausführen zu müssen. Sie können dies beispielsweise tun, um eine Backup-Simulation für Disaster Recovery-Zwecke einzurichten.

In diesem Beispiel lautet der ursprüngliche Simulationsname `MySimulation`. In dem Bereich, in dem der Stack erstellt AWS CloudFormation wird, ist bereits ein Bucket für das Schema vorhanden. AWS-Region Der Bucket enthält eine Version des Schemas, die ordnungsgemäß konfiguriert ist, um die Simulation darin auszuführen AWS-Region. Denken Sie daran, dass das Schema den Speicherort Ihrer App-ZIP-Dateien angibt, bei denen es sich genauso AWS-Region wie bei der Simulation um einen Amazon S3 S3-Bucket handelt. Der ZIP-Bucket und die Dateien der App müssen bereits im AWS-Region Zeitpunkt der Erstellung des AWS CloudFormation Stacks vorhanden sein, da Ihre Simulation sonst nicht gestartet wird. Beachten Sie, dass der Bucket-Name in diesem Beispiel den enthält AWS-Region, aber das bestimmt nicht, wo sich der Bucket tatsächlich befindet. Sie müssen sicherstellen, dass sich der Bucket tatsächlich darin befindet AWS-Region (Sie können die Bucket-Eigenschaften in der Amazon S3-Konsole, mit Amazon S3 APIs oder mit den Amazon S3 S3-Befehlen in der überprüfen AWS CLI).

In diesem Beispiel werden einige integrierte Funktionen und Parameter verwendet AWS CloudFormation , um die Variablenersetzung durchzuführen. Weitere Informationen finden Sie unter Referenz zu [systeminternen Funktionen und Referenz zu Pseudo-Parametern im Benutzerhandbuch](#).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'

```

Verwenden von Schnappschüssen mit AWS CloudFormation

Ein [Snapshot](#) ist eine Sicherungskopie einer Simulation. Im folgenden Beispiel wird eine neue Simulation anhand eines Snapshots statt anhand eines Schemas gestartet. Der Snapshot in diesem Beispiel wurde anhand einer SimSpace Weaver App-SDK-Projektsimulation erstellt. AWS CloudFormation erstellt die neue Simulationsressource und initialisiert sie mit Daten aus dem

Snapshot. Die neue Simulation kann sich von der `MaximumDuration` ursprünglichen Simulation unterscheiden.

Wir empfehlen Ihnen, eine Kopie der App-Rolle Ihrer ursprünglichen Simulation zu erstellen und zu verwenden. Die App-Rolle der ursprünglichen Simulation könnte gelöscht werden, wenn Sie den AWS CloudFormation Stack dieser Simulation löschen.

```
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"
```

Snapshots

Sie können jederzeit einen Snapshot erstellen, um Ihre Simulationsentitätsdaten zu sichern. SimSpace Weaver erstellt eine `.zip`-Datei in einem Amazon S3 S3-Bucket. Sie können mit dem Snapshot eine neue Simulation erstellen. SimSpace Weaver initialisiert die State Fabric Ihrer neuen Simulation mit den im Snapshot gespeicherten Entitätsdaten, startet die Spatial- und Service-Apps, die bei der Erstellung des Snapshots ausgeführt wurden, und setzt die Uhr auf das entsprechende Tick. SimSpace Weaver ruft die Konfiguration Ihrer Simulation aus dem Snapshot statt aus einer Schemadatei ab. Ihre ZIP-Dateien der App müssen sich in Amazon S3 am selben Speicherort befinden wie in der ursprünglichen Simulation. Sie müssen alle benutzerdefinierten Apps separat starten.

Themen

- [Anwendungsfälle für Schnappschüsse](#)
- [Verwenden Sie die SimSpace Weaver Konsole, um mit Snapshots zu arbeiten](#)
- [Verwenden Sie den AWS CLI , um mit Schnappschüssen zu arbeiten](#)

- [Verwenden von Schnappschüssen mit AWS CloudFormation](#)
- [Häufig gestellte Fragen zu Schnappschüssen](#)

Anwendungsfälle für Schnappschüsse

Kehren Sie zu einem früheren Status zurück und erkunden Sie Verzweigungsszenarien

Sie können einen Snapshot Ihrer Simulation erstellen, um sie in einem bestimmten Status zu speichern. Anschließend können Sie anhand dieses Snapshots mehrere neue Simulationen erstellen und verschiedene Szenarien untersuchen, die von diesem Status ausgehen könnten.

Bewährte Methoden für Notfallwiederherstellung und Sicherheit

Wir empfehlen, dass Sie Ihre Simulation regelmäßig sichern, insbesondere bei Simulationen, die länger als eine Stunde dauern oder mehrere Workers verwenden. Backups können Ihnen bei der Wiederherstellung nach Katastrophen und Sicherheitsvorfällen helfen. Schnappschüsse bieten Ihnen die Möglichkeit, Ihre Simulation zu sichern. Für Snapshots müssen sich Ihre ZIP-Dateien Ihrer App am selben Speicherort in Amazon S3 befinden wie zuvor. Wenn Sie in der Lage sein müssen, Ihre ZIP-Dateien Ihrer App an einen anderen Speicherort zu verschieben, müssen Sie eine benutzerdefinierte Backup-Lösung verwenden.

Weitere Informationen zu anderen bewährten Methoden finden Sie unter [Bewährte Methoden bei der Arbeit mit SimSpace Weaver](#) und [Bewährte Sicherheitsmethoden für SimSpace Weaver](#).

Verlängern Sie die Dauer Ihrer Simulation

Ihre Simulationsressource ist die Darstellung Ihrer Simulation in SimSpace Weaver. Alle Simulationsressourcen haben eine `MaximumDuration` Einstellung. Eine Simulationsressource stoppt automatisch, wenn sie ihren Wert erreicht `MaximumDuration`. Der Höchstwert von `MaximumDuration` ist 14D (14 Tage).

Wenn Ihre Simulation länger als die `MaximumDuration` zugehörige Simulationsressource bestehen soll, können Sie einen Snapshot erstellen, bevor die Simulationsressource ihren `MaximumDuration` Wert erreicht. Sie können mit Ihrem Snapshot eine neue Simulation starten (eine neue Simulationsressource erstellen). SimSpace Weaver initialisiert Ihre Entitätsdaten aus dem Snapshot, startet dieselben Spatial- und Service-Apps, die zuvor ausgeführt wurden, und stellt die Uhr wieder her. Sie können Ihre benutzerdefinierten Apps starten und jede zusätzliche

benutzerdefinierte Initialisierung durchführen. Sie können für die `MaximumDuration` neue Simulationsressource einen anderen Wert festlegen, wenn Sie sie starten.

Verwenden Sie die SimSpace Weaver Konsole, um mit Snapshots zu arbeiten

Sie können die SimSpace Weaver Konsole verwenden, um einen Snapshot Ihrer Simulation zu erstellen.

Themen

- [Verwenden Sie die Konsole, um einen Snapshot zu erstellen](#)
- [Verwenden Sie die Konsole, um eine Simulation von einem Snapshot aus zu starten](#)

Verwenden Sie die Konsole, um einen Snapshot zu erstellen

So erstellen Sie einen Snapshot

1. Melden Sie sich bei der an AWS Management Console und stellen Sie eine Verbindung zur [SimSpace Weaver Konsole](#) her.
2. Wählen Sie im Navigationsbereich die Option Simulationen aus.
3. Wählen Sie das Optionsfeld neben dem Namen der Simulation aus. Der Status Ihrer Simulation muss „Gestartet“ lauten.
4. Wählen Sie oben auf der Seite die Option Snapshot erstellen aus.
5. Geben Sie unter Snapshot-Einstellungen für Snapshot-Ziel die Amazon S3 S3-URI eines Buckets oder Buckets und des Ordners ein, in dem Sie Ihren Snapshot erstellen SimSpace Weaver möchten. Sie können „S3 durchsuchen“ wählen, wenn Sie lieber Ihre verfügbaren Buckets durchsuchen und einen Standort auswählen möchten.

Important

Der Amazon S3 S3-Bucket muss sich in derselben Konfiguration AWS-Region befinden wie die Simulation.

 Note

SimSpace Weaver erstellt einen snapshot Ordner innerhalb des ausgewählten Snapshot-Ziels. SimSpace Weaver erstellt die Snapshot-.zip-Datei in diesem snapshot Ordner.

6. Wählen Sie Snapshot erstellen aus.

Verwenden Sie die Konsole, um eine Simulation von einem Snapshot aus zu starten

Um eine Simulation von einem Snapshot aus zu starten, muss Ihre Snapshot-.zip-Datei in einem Amazon S3 S3-Bucket vorhanden sein, auf den Ihre Simulation zugreifen kann. Ihre Simulation verwendet Berechtigungen, die in der App-Rolle definiert sind, die Sie beim Start der Simulation auswählen. Alle ZIP-Dateien der App aus der Originalsimulation müssen sich an denselben Speicherorten befinden wie bei der Erstellung des Snapshots.

Um eine Simulation von einem Snapshot aus zu starten

1. Melden Sie sich bei der an AWS Management Console und stellen Sie eine Verbindung zur [SimSpace Weaver Konsole](#) her.
2. Wählen Sie im Navigationsbereich die Option Simulationen aus.
3. Wählen Sie oben auf der Seite die Option Simulation starten aus.
4. Geben Sie unter Simulationseinstellungen einen Namen und optional eine Beschreibung für Ihre Simulation ein. Ihr Simulationsname muss in Ihrem Fall eindeutig sein AWS-Konto.
5. Wählen Sie unter Simulationsstartmethode die Option Einen Snapshot in Amazon S3 verwenden aus.
6. Geben Sie für Amazon S3 S3-URI für Snapshot die Amazon S3 S3-URI Ihrer Snapshot-Datei ein, oder wählen Sie Browse S3, um die Datei zu durchsuchen und auszuwählen.

 Important

Der Amazon S3 S3-Bucket muss sich in derselben Konfiguration AWS-Region befinden wie die Simulation.

7. Wählen Sie für die IAM-Rolle die App-Rolle aus, die Ihre Simulation verwenden soll.

8. Geben Sie für Maximale Dauer die maximale Dauer ein, für die Ihre Simulationsressource ausgeführt werden soll. Die maximale Wert ist 14D. Weitere Informationen zur maximalen Dauer finden Sie unter [_](#).
9. Wählen Sie unter Tags — optional die Option Neues Tag hinzufügen aus, wenn Sie ein Tag hinzufügen möchten.
10. Wählen Sie Simulation starten aus.

Verwenden Sie den AWS CLI , um mit Schnappschüssen zu arbeiten

Sie können das verwenden AWS CLI , um das SimSpace Weaver APIs von einer Befehlszeile aus aufzurufen. Sie müssen das ordnungsgemäß AWS CLI installiert und konfiguriert haben. Weitere Informationen finden Sie unter [Installation oder Aktualisierung der neuesten Version der AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch für Version 2.

Themen

- [Verwenden Sie den AWS CLI , um einen Snapshot zu erstellen](#)
- [Verwenden Sie den AWS CLI , um eine Simulation von einem Snapshot aus zu starten](#)

Verwenden Sie den AWS CLI , um einen Snapshot zu erstellen

So erstellen Sie einen Snapshot

- Rufen Sie in einer Befehlszeile die CreateSnapshot API auf.

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

Parameter

Simulation

Der Name einer gestarteten Simulation. Sie können ihn verwenden `aws simspaceweaver list-simulations`, um die Namen und den Status Ihrer Simulationen zu sehen.

Ziel

Eine Zeichenfolge, die den Amazon S3-Ziel-Bucket und das optionale Objektschlüsselpräfix für Ihre Snapshot-Datei angibt. Ihr Objektschlüsselpräfix ist normalerweise ein Ordner in

Ihrem Bucket. SimSpace Weaver erstellt Ihren Snapshot in einem snapshot Ordner an diesem Ziel.

⚠ Important

Der Amazon S3 S3-Bucket muss sich in derselben Konfiguration AWS-Region befinden wie die Simulation.

Beispiel

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

Weitere Informationen zur CreateSnapshot API finden Sie [CreateSnapshot](#) in der AWS SimSpace Weaver API-Referenz.

Verwenden Sie den AWS CLI , um eine Simulation von einem Snapshot aus zu starten

Um eine Simulation von einem Snapshot aus zu starten

- Rufen Sie in einer Befehlszeile die StartSimulation API auf.

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

Parameter

Name

Der Name der neuen Simulation. Der Simulationsname muss in Ihrem Fall eindeutig sein AWS-Konto. Sie können verwenden `aws simspaceweaver list-simulations`, um die Namen Ihrer vorhandenen Simulationen zu sehen.

role-arn

Der Amazon-Ressourcenname (ARN) der App-Rolle, die Ihre Simulation verwenden wird.

snapshot-s3-Standort

Eine Zeichenfolge, die den Amazon S3 S3-Bucket und den Objektschlüssel Ihrer Snapshot-Datei angibt.

Important

Der Amazon S3 S3-Bucket muss sich in derselben Konfiguration AWS-Region befinden wie die Simulation.

Beispiel

```
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
```

Weitere Informationen zur `StartSimulation` API finden Sie [StartSimulation](#) in der AWS SimSpace Weaver API-Referenz.

Häufig gestellte Fragen zu Schnappschüssen

Läuft meine Simulation während eines Snapshots weiter?

Ihre Simulationsressourcen werden während eines Snapshots weiter ausgeführt, und Sie erhalten für diesen Zeitraum weiterhin Abrechnungsgebühren. Die Zeit wird auf die maximale Dauer Ihrer Simulation angerechnet. Während der Snapshot läuft, erhalten Ihre Apps keine Häkchen. Wenn Ihr Uhrstatus zu dem `STARTED` Zeitpunkt war, an dem die Snapshot-Erstellung gestartet wurde, zeigt Ihre Uhr trotzdem `STARTED` den Status an. Ihre Apps erhalten nach Abschluss des Snapshots wieder Häkchen. Wenn Ihr Uhrstatus `warSTOPPED`, bleibt Ihr Uhrstatus `bestehenSTOPPED`. Beachten Sie, dass eine Simulation mit einem `STARTED` Status auch dann läuft, wenn ihr Uhrstatus lautet `STOPPED`.

Was passiert, wenn gerade ein Snapshot läuft und meine Simulation ihre maximale Dauer erreicht?

Ihre Simulation beendet den Snapshot und stoppt dann, sobald der Snapshot-Vorgang beendet ist (entweder erfolgreich oder erfolglos). Wir empfehlen Ihnen, den Snapshot-Prozess vorher zu testen, um herauszufinden, wie lange er dauert, mit welcher Größe die Snapshot-Datei zu rechnen ist und ob er erfolgreich abgeschlossen werden sollte.

Was passiert, wenn ich eine Simulation beende, bei der gerade ein Snapshot läuft?

Ein laufender Snapshot wird sofort beendet, wenn Sie die Simulation beenden. Es wird keine Snapshot-Datei erstellt.

Wie kann ich einen laufenden Snapshot stoppen?

Die einzige Möglichkeit, einen laufenden Snapshot zu beenden, besteht darin, die Simulation zu beenden. Sie können eine Simulation nicht erneut starten, nachdem Sie sie beendet haben.

Wie lange dauert es, bis mein Snapshot fertig ist?

Die Zeit, die für die Erstellung eines Snapshots benötigt wird, hängt von Ihrer Simulation ab. Wir empfehlen Ihnen, den Snapshot-Prozess vorher zu testen, um herauszufinden, wie lange Ihre Simulation dauern wird.

Wie groß wird meine Snapshot-Datei sein?

Die Größe einer Snapshot-Datei hängt von Ihrer Simulation ab. Wir empfehlen Ihnen, den Snapshot-Prozess vorher zu testen, um herauszufinden, wie groß die Datei für Ihre Simulation sein könnte.

Messaging

Die Messaging-API vereinfacht die Kommunikation von Anwendung zu Anwendung innerhalb der Simulation. APIs Das Senden und Empfangen von Nachrichten ist Teil des SimSpace Weaver App-SDK. Messaging verwendet derzeit einen Best-Effort-Ansatz zum Senden und Empfangen von Nachrichten. SimSpace Weaver versucht, beim nächsten Simulations-Tick Nachrichten zu senden/zu empfangen, aber es gibt keine garantierten Liefer-, Bestell- oder Ankunftszeiten.

Themen

- [Anwendungsfälle für Messaging](#)
- [Das Messaging verwenden APIs](#)
- [Wann sollte Messaging verwendet werden](#)
- [Tipps für die Arbeit mit Nachrichten](#)
- [Fehler bei der Nachrichtenübermittlung und Problembehandlung](#)

Anwendungsfälle für Messaging

Kommunizieren Sie zwischen Simulationsanwendungen

Verwenden Sie die Messaging-API, um zwischen den Anwendungen in Ihrer Simulation zu kommunizieren. Verwenden Sie sie, um den Status von Objekten in der Ferne zu ändern, das Verhalten von Objekten zu ändern oder Informationen an die gesamte Simulation zu übertragen.

Bestätigen Sie den Empfang einer Nachricht

Gesendete Nachrichten enthalten Informationen über den Absender im Nachrichtenkopf. Verwenden Sie diese Informationen, um nach Erhalt einer Nachricht eine Bestätigungsantwort zurückzusenden.

Leiten Sie die von einer benutzerdefinierten App empfangenen Daten an andere Apps innerhalb der Simulation weiter

Messaging ist kein Ersatz für die Art und Weise, wie Clients eine Verbindung zu benutzerdefinierten Apps herstellen, die in ausgeführt SimSpace Weaver werden. Mit Messaging können Benutzer jedoch Daten von benutzerdefinierten Apps, die Clientdaten empfangen, an andere Apps weiterleiten, die keine externe Verbindung haben. Der Nachrichtenfluss kann auch umgekehrt funktionieren, sodass Apps ohne externe Verbindung Daten an eine benutzerdefinierte App und dann an einen Client weiterleiten können.

Das Messaging verwenden APIs

Die Nachrichten APIs sind im SimSpace Weaver App-SDK (Mindestversion 1.16.0) enthalten. Messaging wird in C++, Python und unseren Integrationen mit Unreal Engine 5 und Unity unterstützt.

Es gibt zwei Funktionen, die Nachrichtentransaktionen verarbeiten: `SendMessage` und `ReceiveMessages`. Alle gesendeten Nachrichten enthalten ein Ziel und eine Nutzlast. Die `ReceiveMessages` API gibt eine Liste der Nachrichten zurück, die sich derzeit in der Warteschlange für eingehende Nachrichten einer App befinden.

C++

Nachricht senden

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

Empfangen Sie Nachrichten

```
AWS_WEAVERRUNTIME_API Result<MessageList> ReceiveMessages(  
    Transaction& txn) noexcept;
```

Python

Nachricht senden

```
api.send_message(  
    txn, # Transaction  
    payload, # api.MessagePayload  
    destination, # api.MessageDestination  
    api.MessageDeliveryType.BestEffort # api.MessageDeliveryType  
)
```

Empfangen Sie Nachrichten

```
api.receive_messages(  
    txn, # Transaction  
) -> api.MessageList
```

Themen

- [Senden von Nachrichten](#)
- [Empfangen von Nachrichten](#)
- [Dem Absender antworten](#)

Senden von Nachrichten

Nachrichten bestehen aus einer Transaktion (ähnlich wie bei anderen Weaver-API-Aufrufen), einer Nutzlast und einem Ziel.

Nachrichten-Payload

Die Nachrichtennutzlast ist eine flexible Datenstruktur mit bis zu 256 Byte. Wir empfehlen Folgendes als bewährte Methode für die Erstellung Ihrer Nachrichtennutzlasten.

Um die Nachrichten-Payload zu erstellen

1. Erstellen Sie eine Datenstruktur (z. B. eine `struct` in C++), die den Inhalt der Nachricht definiert.
2. Erstellen Sie die Nachrichtennutzlast, die die Werte enthält, die in Ihrer Nachricht gesendet werden sollen.
3. Erstellen Sie das `MessagePayload` Objekt.

Ziel der Nachricht

Das Ziel einer Nachricht wird durch das `MessageEndpoint` Objekt definiert. Dies beinhaltet sowohl einen Endpunkttyp als auch eine Endpunkt-ID. Derzeit wird nur der Endpunkttyp `Partition`, der es Ihnen ermöglicht, Nachrichten an andere Partitionen in der Simulation zu adressieren. Die Endpunkt-ID ist die `Partitions-ID` Ihres Zielziels.

Sie können in einer Nachricht nur eine Zieladresse angeben. Erstellen und senden Sie mehrere Nachrichten, wenn Sie Nachrichten an mehr als eine Partition gleichzeitig senden möchten.

Hinweise zum Auflösen eines Nachrichtenendpunkts von einer Position aus finden Sie unter [Tipps für die Arbeit mit Nachrichten](#).

Senden Sie die Nachricht

Sie können die `SendMessage` API verwenden, nachdem Sie die Ziel- und Payload-Objekte erstellt haben.

C++

```
Api::SendMessage(transaction, payload, destination,  
MessageDeliveryType::BestEffort);
```

Python

```
api.send_message(txn, payload, destination, api.MessageDeliveryType.BestEffort)
```

Vollständiges Beispiel für das Senden von Nachrichten

Das folgende Beispiel zeigt, wie Sie eine generische Nachricht erstellen und senden können. In diesem Beispiel werden 16 einzelne Nachrichten gesendet. Jede Nachricht enthält eine Nutzlast mit einem Wert zwischen 0 und 15 und den aktuellen Simulationsticker.

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

```

Empfangen von Nachrichten

SimSpace Weaver übermittelt Nachrichten in die Warteschlange für eingehende Nachrichten einer Partition. Verwenden Sie die `ReceiveMessages` API, um ein `MessageList` Objekt abzurufen, das die Nachrichten aus der Warteschlange enthält. Verarbeiten Sie jede Nachricht mit der `ExtractMessage` API, um die Nachrichtendaten abzurufen.

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

```

Dem Absender antworten

Jede empfangene Nachricht enthält einen Nachrichtenkopf mit Informationen über den ursprünglichen Absender der Nachricht. Sie können den `message.header.source_endpoint` verwenden, um eine Antwort zu senden.

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
```

Wann sollte Messaging verwendet werden

Das Einsenden von Nachrichten SimSpace Weaver bietet ein weiteres Muster für den Informationsaustausch zwischen Simulationsanwendungen. Abonnements bieten einen Pull-Mechanismus zum Lesen von Daten aus bestimmten Anwendungen oder Bereichen der Simulation; Nachrichten bieten einen Push-Mechanismus, um Daten an bestimmte Anwendungen oder Bereiche der Simulation zu senden.

Im Folgenden finden Sie zwei Anwendungsfälle, in denen es hilfreicher ist, Daten mithilfe von Nachrichten weiterzuleiten, als Daten über ein Abonnement abzurufen oder zu lesen.

Example 1: Senden eines Befehls an eine andere App, um die Position einer Entität zu ändern

```
// 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);

```

Auf der Empfangsseite aktualisiert die App die Position der Entität und schreibt sie in die 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: Senden einer Nachricht zum Erstellen einer Entität an eine räumliche App

```

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);

```

Auf der Empfangsseite erstellt die App eine neue Entität in der State Fabric und aktualisiert deren Position.

```

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)));
}
}

```

Tipps für die Arbeit mit Nachrichten

Löst einen Endpunkt anhand einer Position oder eines Anwendungsnamens auf

Sie können die `AllPartitions` Funktion verwenden, um die räumlichen Grenzen und eine Domänen-ID abzurufen, die Sie zur Bestimmung der Nachrichtenpartition IDs und der Nachrichtenziele benötigen. Wenn Sie jedoch die Position kennen, an der Sie eine Nachricht senden möchten, aber nicht die Partitions-ID, können Sie die `MessageEndpointResolver` Funktion verwenden.

```

/**
 * 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);
};

```

Serialisierung und Deserialisierung der Nachrichtennutzdaten

Sie können die folgenden Funktionen verwenden, um Nachrichtennutzlasten zu erstellen und zu lesen. Weitere Informationen finden Sie unter `MessagingUtils.h` in der App-SDK-Bibliothek auf Ihrem lokalen System.

```

/**
 * 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());
}
```

Fehler bei der Nachrichtenübermittlung und Problembehandlung

Bei der Verwendung von Messaging können die folgenden Fehler auftreten APIs.

Fehler bei der Endpunktauflösung

Diese Fehler können auftreten, bevor eine App eine Nachricht sendet.

Überprüfung des Domainnamens

Das Senden einer Nachricht an einen ungültigen Endpunkt führt zu dem folgenden Fehler:

```
ManifoldError::InvalidArgument {"No DomainId found for the given domain name" }
```

Dies kann passieren, wenn Sie versuchen, eine Nachricht an eine benutzerdefinierte App zu senden und diese benutzerdefinierte App der Simulation noch nicht beigetreten ist. Stellen Sie mithilfe der `DescribeSimulation` API sicher, dass Ihre benutzerdefinierte App gestartet wurde, bevor Sie eine Nachricht an sie senden. Dieses Verhalten ist dasselbe in SimSpace Weaver Local und der AWS Cloud.

Stellungsprüfung

Der Versuch, einen Endpunkt mit einem gültigen Domainnamen, aber einer ungültigen Position aufzulösen, führt zu dem folgenden Fehler.

```
ManifoldError::InvalidArgument {"Could not resolve endpoint from domain : DomainId
{ value: domain-id } and position: Vector2F32 { x: x-position, y: y-position}" }
```

Wir empfehlen, das `MessageEndpointResolver` in der im SimSpace Weaver App-SDK enthaltenen `MessageUtils` Bibliothek zu verwenden.

Fehler beim Senden von Nachrichten

Die folgenden Fehler können auftreten, wenn eine App eine Nachricht sendet.

Das Limit für das Senden von Nachrichten pro App und Tick wurde überschritten

Das aktuelle Limit für die Anzahl der Nachrichten, die pro App pro Simulationstreck gesendet werden können, liegt bei 128. Nachfolgende Aufrufe mit demselben Tick schlagen mit dem folgenden Fehler fehl:

```
ManifoldError::CapacityExceeded {"At Max Outgoing Message capacity: {}", 128}
```

SimSpace Weaver versucht, beim nächsten Tick noch nicht gesendete Nachrichten zu senden. Senken Sie die Sendefrequenz, um dieses Problem zu beheben. Kombinieren Sie Nachrichtennutzlasten, die unter dem 256-Byte-Limit liegen, um die Anzahl der ausgehenden Nachrichten zu verringern.

Dieses Verhalten ist dasselbe in SimSpace Weaver Local und in der AWS Cloud.

Die Größenbeschränkung für die Nachrichtennutzlast wurde überschritten

Das aktuelle Limit für die Größe der Nachrichtennutzdaten liegt in beiden Fällen bei 256 Byte SimSpace Weaver Local und in der AWS Cloud. Das Senden einer Nachricht mit einer Nutzlast von mehr als 256 Byte führt zu dem folgenden Fehler:

```
ManifoldError::CapacityExceeded {"Message data too large! Max size: {}", 256}
```

SimSpace Weaver überprüft jede Nachricht und lehnt nur diejenigen ab, die das Limit überschreiten. Wenn Ihre App beispielsweise versucht, 10 Nachrichten zu senden, und eine Nachricht die Prüfung nicht besteht, wird nur diese eine Nachricht zurückgewiesen. SimSpace Weaver sendet die anderen 9 Nachrichten.

Dieses Verhalten ist dasselbe in SimSpace Weaver Local und der AWS Cloud.

Ziel ist dasselbe wie Quelle

Apps können keine Nachrichten an Partitionen senden, deren Eigentümer sie sind. Sie erhalten die folgende Fehlermeldung, wenn eine App eine Nachricht an eine Partition sendet, deren Eigentümer sie ist.

```
ManifoldError::InvalidArgument { "Destination is the same as source" }
```

Dieses Verhalten ist dasselbe in SimSpace Weaver Local und der AWS Cloud.

Nachrichtenübermittlung nach besten Kräften

SimSpace Weaver garantiert keine Nachrichtenzustellung. Der Dienst versucht, die Zustellung der Nachrichten beim nächsten Simulations-Tick abzuschließen, aber Nachrichten können verloren gehen oder sich verzögern.

Bewährte Methoden bei der Arbeit mit SimSpace Weaver

Wir empfehlen die folgenden bewährten Methoden für die Arbeit mit SimSpace Weaver.

Themen

- [Richten Sie Abrechnungsalarme ein](#)
- [Verwenden von SimSpace Weaver Local](#)
- [Beenden Sie Simulationen, die Sie nicht benötigen](#)
- [Löschen Sie Ressourcen, die Sie nicht benötigen](#)
- [Haben Sie Backups](#)

Richten Sie Abrechnungsalarme ein

Es ist einfach, Ressourcen bereitzustellen AWS und sie ständig laufen zu lassen, auch wenn sie nicht mehr benötigt werden. Dies kann zu unvorhersehbaren Kosten führen, die Sie überraschen können, wenn Sie Ihre Rechnung erhalten. Sie können in Amazon einen Alarm konfigurieren CloudWatch , der ausgelöst wird und Sie benachrichtigt, wenn Ihre Kosten einen von Ihnen festgelegten Schwellenwert überschreiten. Sie können Ihre Kosten mithilfe von Kostenmanagement-Tools überprüfen. Weitere Informationen finden Sie unter:

- [Richten Sie einen Abrechnungsalarm ein, um Ihre geschätzten AWS Gebühren im Auge zu behalten](#)
- [Was ist AWS Cost Management](#)

Verwenden von SimSpace Weaver Local

Wir empfehlen Ihnen, Ihre Simulationen SimSpace Weaver Local zu entwickeln und zu testen, bevor Sie sie in den SimSpace Weaver Dienst hochladen. AWS Cloud Zu den Vorteilen der Entwicklung mit SimSpace Weaver Local gehören:

- Sie müssen nicht auf große Uploads warten
- Die Anzahl der lokalen Simulationen, die Sie erstellen können, ist unbegrenzt
- Die Rechenzeit auf Ihrem lokalen Computer wird Ihnen nicht in Rechnung gestellt
- Direkter Zugriff auf die Konsolenausgabe aus Ihren Apps

- Ändern Sie Ihre lokale Simulation, erstellen Sie sie neu und starten Sie sie neu, ohne sie im AWS Cloud

Beenden Sie Simulationen, die Sie nicht benötigen

Sie erhalten Abrechnungsgebühren für eine Simulation, während sie läuft. Sie müssen eine Simulation beenden, um keine Gebühren mehr dafür zu zahlen. Das Ausführen von Simulationen wird auch auf Ihre Quote für die maximale Anzahl von Simulationen angerechnet. Bei einer laufenden Simulation, für die die Protokollierung konfiguriert ist, können auch große Mengen an Protokollen generiert werden, für die Ihnen auch Abrechnungsgebühren berechnet werden. Sie sollten jede Simulation beenden, die Sie nicht benötigen, damit keine zusätzlichen Gebühren mehr anfallen.

Important

Wenn Sie die Simulationsuhr anhalten, wird die Simulation nicht gestoppt, die Uhr veröffentlicht lediglich keine Ticks mehr in Ihren Apps. Sie können eine Simulation nicht erneut starten, nachdem Sie sie beendet haben.

Löschen Sie Ressourcen, die Sie nicht benötigen

Mit jeder Simulation, die Sie erstellen, werden SimSpace Weaver auch Ressourcen in anderen AWS Diensten erstellt. Sie können Abrechnungsgebühren für Ressourcen und Daten in diesen anderen Diensten abrufen. Laufende und fehlgeschlagene Simulationen werden auf Ihr Kontingent für die maximale Anzahl von Simulationen angerechnet. Sie sollten nicht benötigte fehlgeschlagene Simulationen löschen, damit Sie neue Simulationen starten können. Wenn Sie eine Simulation löschen, werden Ressourcen für Ihre Simulation, die in anderen AWS Diensten vorhanden sind, möglicherweise nicht gelöscht. Beispielsweise bleiben alle Simulationsprotokolldaten in Amazon CloudWatch Logs dort, bis Sie sie löschen. Für diese Protokolldaten werden Ihnen Abrechnungsgebühren berechnet. Sie sollten alle zugehörigen Ressourcen für Ihre Simulationen bereinigen, falls Sie sie nicht mehr benötigen.

Haben Sie Backups

Es ist eine gute Idee, Backups und Backup-Pläne für alles zu haben. Sie sollten nicht davon ausgehen, AWS dass Sie Ihre Daten nicht sichern müssen, nur weil sie gespeichert sind. Sie müssen

Ihr eigenes System erstellen, wenn Sie Ihren Simulationsstatus sichern müssen. Erwägen Sie, mehrere zu verwenden AWS-Regionen und einen Plan zu haben, um Ihre Produktionslast bei Bedarf schnell auf ein anderes AWS-Region umstellen zu können. Weitere Informationen zu AWS-Regionen dieser Unterstützung SimSpace Weaver finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#).

Sicherheit in AWS SimSpace Weaver

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS SimSpace Weaver, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können SimSpace Weaver. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen SimSpace Weaver , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer SimSpace Weaver Ressourcen unterstützen.

Themen

- [Datenschutz in AWS SimSpace Weaver](#)
- [Identity and Access Management für AWS SimSpace Weaver](#)
- [Protokollierung und Überwachung von Sicherheitsereignissen in AWS SimSpace Weaver](#)
- [Überprüfung der Einhaltung der Vorschriften für AWS SimSpace Weaver](#)
- [Resilienz in AWS SimSpace Weaver](#)
- [Sicherheit der Infrastruktur in AWS SimSpace Weaver](#)
- [Konfiguration und Schwachstellenanalyse in AWS SimSpace Weaver](#)
- [Bewährte Sicherheitsmethoden für SimSpace Weaver](#)

Datenschutz in AWS SimSpace Weaver

Das [Modell der AWS gemeinsamen Verantwortung](#) und geteilter Verantwortung gilt für den Datenschutz in AWS SimSpace Weaver. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der SimSpace Weaver API oder auf andere AWS-Services Weise arbeiten oder diese verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet

werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

Daten gelten als ruhend, wenn sie sich in einem nichtflüchtigen (persistenten) Datenspeicher, wie z. B. einer Festplatte, befinden. Daten, die sich in flüchtigen Datenspeichern wie Speicher und Registern befinden, gelten nicht als ruhend.

Wenn Sie verwenden SimSpace Weaver, sind die einzigen Daten im Ruhezustand:

- Apps und Schemas, die Sie auf Amazon Simple Storage Service (Amazon S3) hochladen
- In Amazon gespeicherte Simulationsprotokolldaten CloudWatch

Andere Daten, die intern SimSpace Weaver verwendet werden, bleiben nicht erhalten, nachdem Sie Ihre Simulation beendet haben.

Informationen zum Verschlüsseln Ihrer Daten im Ruhezustand finden Sie unter:

- [Verschlüsseln Sie Ihre Daten in Amazon S3](#)
- [Verschlüsseln Sie Ihre Protokolldaten](#)

Verschlüsselung während der Übertragung

Ihre Verbindungen zur SimSpace Weaver API über das AWS Command Line Interface (AWS CLI), das AWS SDK und das SimSpace Weaver App-SDK verwenden TLS-Verschlüsselung mit dem [Signaturprozess von Signature Version 4](#). AWS verwaltet die Authentifizierung mithilfe der von IAM definierten Zugriffsrichtlinien für die Sicherheitsanmeldedaten, die Sie für die Verbindung verwenden.

SimSpace Weaver verwendet intern TLS, um eine Verbindung zu anderen AWS Diensten herzustellen, die es verwendet.

Important

Die Kommunikation zwischen Ihren Apps und ihren Kunden beinhaltet nichts SimSpace Weaver. Es liegt in Ihrer Verantwortung, die Kommunikation mit Simulationsclients bei Bedarf

zu verschlüsseln. Wir empfehlen Ihnen, eine Lösung zur Verschlüsselung aller Daten zu entwickeln, die über Client-Verbindungen übertragen werden.

Weitere Informationen zu AWS Diensten, die Ihre Verschlüsselungslösungen unterstützen können, finden Sie [im AWS Sicherheitsblog](#).

Datenschutz für den Datenverkehr zwischen Netzwerken

SimSpace Weaver Rechenressourcen befinden sich in einer Amazon VPC, die von allen SimSpace Weaver Kunden gemeinsam genutzt wird. Der gesamte interne SimSpace Weaver Servicedatenverkehr verbleibt im AWS Netzwerk und wird nicht über das Internet übertragen. Die Kommunikation zwischen Simulationsclients und Ihren Apps erfolgt über das Internet.

Identity and Access Management für AWS SimSpace Weaver

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. SimSpace Weaver IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie AWS SimSpace Weaver funktioniert mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)
- [Berechtigungen, die für Sie SimSpace Weaver erstellt werden](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Fehlerbehebung bei AWS SimSpace Weaver Identität und Zugriff](#)

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. SimSpace Weaver

Dienstbenutzer — Wenn Sie den SimSpace Weaver Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr SimSpace Weaver Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter [Fehlerbehebung bei AWS SimSpace Weaver Identität und Zugriff](#) finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in SimSpace Weaver haben.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die SimSpace Weaver Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf SimSpace Weaver. Es ist Ihre Aufgabe, zu bestimmen, auf welche SimSpace Weaver Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann SimSpace Weaver, finden Sie unter [Wie AWS SimSpace Weaver funktioniert mit IAM](#).

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf SimSpace Weaver verfassen können. Beispiele für SimSpace Weaver identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen

Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter [AWS Signature Version 4 für API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [AWS Multi-Faktor-Authentifizierung \(MFA\) in IAM](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem

beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb von Ihnen AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management

Console, können Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Methoden für die Übernahme einer Rolle](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.
- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen

mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer Service-Verknüpfung verbunden ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon ausgeführte Anwendungen EC2** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen. Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden](#).

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter [Auswählen zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein

bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten, die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen

zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.

- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter [Resource Control Policies \(RCPs\)](#) im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

Wie AWS SimSpace Weaver funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren SimSpace Weaver, mit welchen IAM-Funktionen Sie arbeiten können. SimSpace Weaver

IAM-Funktionen, die Sie mit verwenden können AWS SimSpace Weaver

IAM-Feature	SimSpace Weaver Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein

IAM-Feature	SimSpace Weaver Unterstützung
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie SimSpace Weaver und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für SimSpace Weaver

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet

ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für SimSpace Weaver

Beispiele für SimSpace Weaver identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)

Ressourcenbasierte Richtlinien finden Sie in SimSpace Weaver

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für SimSpace Weaver

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der SimSpace Weaver Aktionen finden Sie unter [Aktionen definiert von AWS SimSpace Weaver](#) in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix SimSpace Weaver verwendet:

```
simspaceweaver
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "simspaceweaver:action1",  
  "simspaceweaver:action2"  
]
```

Beispiele für SimSpace Weaver identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)

Politische Ressourcen für SimSpace Weaver

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen](#)

[\(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*" 
```

Eine Liste der SimSpace Weaver Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Resources defined by AWS SimSpace Weaver](#) in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von AWS SimSpace Weaver definierte Aktionen](#).

Beispiele für SimSpace Weaver identitätsbasierte Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)

Bedingungsschlüssel für Richtlinien für SimSpace Weaver

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich oder kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der SimSpace Weaver Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für AWS SimSpace Weaver](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen definiert von AWS SimSpace Weaver](#).

Beispiele für SimSpace Weaver identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)

Zugriffskontrolllisten (ACLs) in SimSpace Weaver

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit SimSpace Weaver

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit SimSpace Weaver

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.](#)

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln von einer Benutzerrolle zu einer IAM-Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM.](#)

Serviceübergreifende Prinzipalberechtigungen für SimSpace Weaver

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten.](#)

Servicerollen für SimSpace Weaver

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die SimSpace Weaver Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, SimSpace Weaver wenn Sie dazu eine Anleitung erhalten.

Die SimSpace Weaver App-SDK-Skripts verwenden eine AWS CloudFormation Vorlage, um Ressourcen in anderen AWS Diensten zur Unterstützung Ihrer Simulation zu erstellen. Eine dieser Ressourcen ist die App-Rolle für Ihre Simulation. SimSpace Weaver übernimmt die App-Rolle, um in Ihrem Namen Aktionen in AWS-Konto Ihrem Namen auszuführen, z. B. Protokolldaten in CloudWatch Logs zu schreiben. Weitere Informationen zur App-Rolle finden Sie unter [Berechtigungen, die für Sie SimSpace Weaver erstellt werden](#).

Mit Diensten verknüpfte Rollen für SimSpace Weaver

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, SimSpace Weaver -Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden SimSpace Weaver, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für AWS SimSpace Weaver](#) in der Service Authorization Reference.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der SimSpace Weaver -Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Erlauben Sie Benutzern, Simulationen zu erstellen und auszuführen](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand SimSpace Weaver Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden

Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der SimSpace Weaver -Konsole

Um auf die AWS SimSpace Weaver Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den SimSpace Weaver Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die SimSpace Weaver Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die SimSpace Weaver *ConsoleAccess* oder die *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-API.

```
{
  "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": "*"
    }
]
}

```

Erlauben Sie Benutzern, Simulationen zu erstellen und auszuführen

Diese Beispiel-IAM-Richtlinie bietet die grundlegenden Berechtigungen, die zum Erstellen und Ausführen von Simulationen in SimSpace Weaver erforderlich sind.

```

{
  "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"
        }
    }
}
]
}

```

Berechtigungen, die für Sie SimSpace Weaver erstellt werden

Wenn Sie ein SimSpace Weaver Projekt erstellen, erstellt der Dienst eine AWS Identity and Access Management (IAM-) Rolle mit dem Namen `weaver-project-name-app-role` und einer IAM-Vertrauensrichtlinie. Die Vertrauensrichtlinie ermöglicht es SimSpace Weaver, die Rolle zu übernehmen, sodass sie Operationen für Sie ausführen kann.

Richtlinie für Berechtigungen für App-Rollen

Die Simulations-App-Rolle hat die folgende Berechtigungsrichtlinie.

```

{
    "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": "*"
}
]
}

```

Vertrauensrichtlinie für App-Rollen

SimSpace Weaver fügt der Rolle der Simulations-App eine Vertrauensbeziehung als [Vertrauensrichtlinie](#) hinzu. SimSpace Weaver erstellt eine Vertrauensrichtlinie für jede Simulation, ähnlich dem folgenden Beispiel.

```

{
  "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

In diesem Beispiel lautet die Kontonummer 111122223333 und der Simulationsname MySimName. Diese Werte unterscheiden sich in Ihren Vertrauensrichtlinien.

Serviceübergreifende Confused-Deputy-Prävention

Das [Problem des verwirrten Stellvertreters](#) ist ein Sicherheitsproblem, bei dem eine Entität, die nicht berechtigt ist, eine Aktion auszuführen, eine Entität mit mehr Rechten dazu verleiten kann, die Aktion auszuführen. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel [aws:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die der AWS SimSpace Weaver Ressource einen anderen Dienst gewähren. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den Amazon-Ressourcennamen (ARN) eines Amazon-S3-Buckets, müssen Sie beide globale Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der Wert von `aws:SourceArn` muss den ARN der Erweiterung verwenden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Erweiterung nicht kennen oder wenn Sie mehrere Erweiterungen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannt Teile des ARN. Beispiel, `arn:aws:simspaceweaver*:111122223333:*`.

Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, SimSpace Weaver um das Problem des verwirrten Stellvertreters zu vermeiden. Diese Richtlinie erlaubt es nur, die Rolle SimSpace Weaver zu übernehmen, wenn die Anfrage vom angegebenen Quellkonto stammt und mit dem angegebenen ARN versehen ist. In diesem Fall SimSpace Weaver kann die Rolle nur für Anfragen aus Simulationen im eigenen Konto des Anforderers (`111122223333`) und nur in der angegebenen Region (`us-west-2`) übernommen werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/*"
        }
      }
    }
  ]
}
```

Eine sicherere Methode, diese Richtlinie zu schreiben, besteht darin, den Simulationsnamen in die aufzunehmen `aws:SourceArn`, wie im folgenden Beispiel gezeigt, wodurch die Richtlinie auf eine Simulation mit dem Namen beschränkt wird: `MyProjectSimulation_22-10-04_22_10_15`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15"
        }
      }
    }
  ]
}
```

Wenn Sie `aws:SourceArn` ausdrücklich eine Kontonummer angeben, können Sie den `Condition` Elementtest für die weglassen `aws:SourceAccount` (weitere Informationen finden Sie im [IAM-Benutzerhandbuch](#)), z. B. in der folgenden vereinfachten Richtlinie:

```
{
  "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/
MyProjectSimulation_22-10-04_22_10_15"
      }
    }
  }
]
```

Fehlerbehebung bei AWS SimSpace Weaver Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit SimSpace Weaver und IAM auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in SimSpace Weaver](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte meine Zugriffsschlüssel anzeigen](#)
- [Ich bin Administrator und möchte anderen den Zugriff ermöglichen SimSpace Weaver](#)
- [Ich möchte Personen außerhalb von mir den Zugriff auf meine Ressourcen ermöglichen AWS-Konto SimSpace Weaver](#)

Ich bin nicht berechtigt, eine Aktion durchzuführen in SimSpace Weaver

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der mateojackson IAM-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven *my-example-widget*-Ressource zu verwenden, jedoch nicht über `simspaceweaver:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
simspaceweaver:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *my-example-widget* auf die Ressource `simspaceweaver:GetWidget` zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an SimSpace Weaver übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in SimSpace Weaver auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen odzur Verfügung gestellt.

Ich möchte meine Zugriffsschlüssel anzeigen

Nachdem Sie Ihre IAM-Benutzerzugriffsschlüssel erstellt haben, können Sie Ihre Zugriffsschlüssel-ID jederzeit anzeigen. Sie können Ihren geheimen Zugriffsschlüssel jedoch nicht erneut anzeigen. Wenn Sie den geheimen Zugriffsschlüssel verlieren, müssen Sie ein neues Zugriffsschlüsselpaar erstellen.

Zugriffsschlüssel bestehen aus zwei Teilen: einer Zugriffsschlüssel-ID (z. B. `AKIAIOSFODNN7EXAMPLE`) und einem geheimen Zugriffsschlüssel (z. B. `wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Ähnlich wie bei Benutzernamen und Passwörtern müssen Sie die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel zusammen verwenden, um Ihre Anforderungen zu authentifizieren. Verwalten Sie Ihre Zugriffsschlüssel so sicher wie Ihren Benutzernamen und Ihr Passwort.

⚠ Important

Geben Sie Ihre Zugriffsschlüssel nicht an Dritte weiter, auch nicht für die [Suche nach Ihrer kanonischen Benutzer-ID](#). Auf diese Weise können Sie jemandem dauerhaften Zugriff auf Ihre gewähren AWS-Konto.

Während der Erstellung eines Zugriffsschlüsselpaars werden Sie aufgefordert, die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel an einem sicheren Speicherort zu speichern. Der geheime Zugriffsschlüssel ist nur zu dem Zeitpunkt verfügbar, an dem Sie ihn erstellen. Wenn Sie Ihren geheimen Zugriffsschlüssel verlieren, müssen Sie Ihrem IAM-Benutzer neue Zugriffsschlüssel hinzufügen. Sie können maximal zwei Zugriffsschlüssel besitzen. Wenn Sie bereits zwei Zugriffsschlüssel besitzen, müssen Sie ein Schlüsselpaar löschen, bevor Sie ein neues erstellen. Anweisungen hierfür finden Sie unter [Verwalten von Zugriffsschlüsseln](#) im IAM-Benutzerhandbuch.

Ich bin Administrator und möchte anderen den Zugriff ermöglichen SimSpace Weaver

Um anderen den Zugriff zu ermöglichen SimSpace Weaver, müssen Sie den Personen oder Anwendungen, die Zugriff benötigen, die entsprechenden Berechtigungen erteilen. Wenn Sie Personen und Anwendungen verwalten, weisen Sie Benutzern oder Gruppen Berechtigungssätze zu, um deren Zugriffsebene zu definieren. AWS IAM Identity Center Mit Berechtigungssätzen werden automatisch IAM-Richtlinien erstellt und den IAM-Rollen zugewiesen, die der Person oder Anwendung zugeordnet sind. Weitere Informationen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter [Berechtigungssätze](#).

Wenn Sie IAM Identity Center nicht verwenden, müssen Sie IAM-Entitäten (Benutzer oder Rollen) für die Personen oder Anwendungen erstellen, die Zugriff benötigen. Anschließend müssen Sie der Entität eine Richtlinie anfügen, die dieser die korrekten Berechtigungen in SimSpace Weaver gewährt. Nachdem die Berechtigungen erteilt wurden, stellen Sie dem Benutzer oder Anwendungsentwickler die Anmeldeinformationen zur Verfügung. Sie werden diese Anmeldeinformationen für den Zugriff verwenden AWS. Weitere Informationen zum Erstellen von IAM-Benutzern, -Gruppen, -Richtlinien und -Berechtigungen finden Sie im [IAM-Benutzerhandbuch unter IAM-Identitäten sowie Richtlinien und Berechtigungen in IAM](#).

Ich möchte Personen außerhalb von mir den Zugriff auf meine Ressourcen ermöglichen AWS-KontoSimSpace Weaver

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen SimSpace Weaver unterstützt werden, finden Sie unter [Wie AWS SimSpace Weaver funktioniert mit IAM](#)
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Protokollierung und Überwachung von Sicherheitsereignissen in AWS SimSpace Weaver

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit SimSpace Weaver und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können.

AWS und SimSpace Weaver stellen Sie verschiedene Tools bereit, mit denen Sie Ihre Simulationsressourcen überwachen und auf mögliche Vorfälle reagieren können.

Loggt sich bei Amazon ein CloudWatch

SimSpace Weaver speichert seine Logs in CloudWatch. Sie können diese Protokolle verwenden, um Ereignisse in Ihrer Simulation zu überwachen (z. B. das Starten und Stoppen von Apps) sowie zum Debuggen. Weitere Informationen finden Sie unter [SimSpace Weaver loggt sich in Amazon CloudWatch Logs ein](#).

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Scaling-Richtlinie gesendet. CloudWatch Alarme werden ausgelöst, wenn sich ihr Status ändert, und sie werden für eine bestimmte Anzahl von Zeiträumen aufrechterhalten, nicht wenn sich ein bestimmter Zustand befindet. Weitere Informationen finden Sie unter [Überwachung SimSpace Weaver mit Amazon CloudWatch](#).

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden SimSpace Weaver. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, an die die Anfrage gestellt wurde SimSpace Weaver, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen ermitteln. Weitere Informationen finden Sie unter [Protokollieren von AWS SimSpace Weaver API-Aufrufen mit AWS CloudTrail](#).

Überprüfung der Einhaltung der Vorschriften für AWS SimSpace Weaver

SimSpace Weaver fällt nicht in den Geltungsbereich AWS irgendwelcher Compliance-Programme.

Externe Prüfer bewerten die Sicherheit und Konformität anderer AWS Dienste im Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Um zu erfahren, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Compliance und Governance im Bereich Sicherheit](#) – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- [Referenz für berechnete HIPAA-Services](#) – Listet berechnete HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerelementreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS SimSpace Weaver

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale](#) Infrastruktur.

Sicherheit der Infrastruktur in AWS SimSpace Weaver

Als verwalteter Dienst AWS SimSpace Weaver ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff SimSpace Weaver über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sicherheitsmodell für Netzwerkkonnektivität

Ihre Simulationen werden auf Compute-Instances innerhalb einer Amazon VPC ausgeführt, die sich in einer von Ihnen ausgewählten AWS Region befindet. Eine Amazon VPC ist ein virtuelles Netzwerk

in der AWS Cloud, das die Infrastruktur nach Arbeitslast oder organisatorischer Einheit isoliert. Die Kommunikation zwischen Recheninstanzen innerhalb der Amazon VPC bleibt im AWS Netzwerk und wird nicht über das Internet übertragen. Ein Teil der internen Dienstkommunikation erfolgt über das Internet und ist verschlüsselt. Simulationen für alle Kunden, die in derselben AWS Region laufen, verwenden dieselbe Amazon VPC. Simulationen für verschiedene Kunden verwenden separate Recheninstanzen innerhalb derselben Amazon VPC.

Kommunikation zwischen Ihren Simulations-Clients und Ihren laufenden Simulationen SimSpace Weaver erfolgt über das Internet. SimSpace Weaver verarbeitet diese Verbindungen nicht. Es liegt in Ihrer Verantwortung, Ihre Client-Verbindungen zu sichern.

Ihre Verbindungen zum SimSpace Weaver Dienst erfolgen über das Internet und sind verschlüsselt. Dazu gehören Verbindungen AWS Management Console, die das, AWS Command Line Interface (AWS CLI), das AWS Software Development Kits (SDK) und das SimSpace Weaver App-SDK verwenden.

Konfiguration und Schwachstellenanalyse in AWS SimSpace Weaver

Konfiguration und IT-Kontrollen liegen in der gemeinsamen Verantwortung von Ihnen AWS und Ihnen. Weitere Informationen finden Sie unter Modell der AWS [gemeinsamen Verantwortung, Modell](#) der der . AWS erledigt grundlegende Sicherheitsaufgaben für die zugrunde liegende Infrastruktur, wie z. B. das Patchen des Betriebssystems auf Recheninstanzen, die Firewallkonfiguration und die Notfallwiederherstellung der AWS Infrastruktur. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie unter [Bewährte Methoden für Sicherheit, Identität und Compliance](#).

Sie sind für die Sicherheit Ihrer Simulationssoftware verantwortlich:

- Pflegen Sie Ihren App-Code, einschließlich Updates und Sicherheitspatches.
- Authentifizieren und verschlüsseln Sie die Kommunikation zwischen Ihren Simulationsclients und den Apps, mit denen sie eine Verbindung herstellen.
- Aktualisieren Sie Ihre Simulationen, um die neuesten SDK-Versionen zu verwenden, einschließlich des AWS SDK und des SimSpace Weaver App-SDK.

Note

SimSpace Weaver unterstützt keine Updates für Apps in einer laufenden Simulation. Wenn Sie Ihre Apps aktualisieren müssen, müssen Sie die Simulation beenden und löschen und dann eine neue Simulation mit dem aktualisierten App-Code erstellen. Wir empfehlen, den Simulationsstatus in einem externen Datenspeicher zu speichern, damit Sie ihn wiederherstellen können, falls Sie die Simulation erneut erstellen müssen.

Bewährte Sicherheitsmethoden für SimSpace Weaver

In diesem Abschnitt werden bewährte Sicherheitsmethoden beschrieben, die spezifisch für sind SimSpace Weaver. Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie unter [Bewährte Methoden für Sicherheit, Identität und Compliance](#). AWS

Themen

- [Verschlüsseln Sie die Kommunikation zwischen Ihren Apps und ihren Clients](#)
- [Erstellen Sie regelmäßig eine Sicherungskopie Ihres Simulationsstatus](#)
- [Pflegen Sie Ihre Apps und SDKs](#)

Verschlüsseln Sie die Kommunikation zwischen Ihren Apps und ihren Clients

SimSpace Weaver verwaltet nicht die Kommunikation zwischen Ihren Apps und ihren Kunden. Sie sollten irgendeine Form der Authentifizierung und Verschlüsselung für Clientsitzungen implementieren.

Erstellen Sie regelmäßig eine Sicherungskopie Ihres Simulationsstatus

SimSpace Weaver speichert Ihren Simulationsstatus nicht. Simulationen, die gestoppt wurden (aufgrund eines API-Aufrufs, einer Konsolenoption oder eines Systemabsturzes), speichern ihren Status nicht und es gibt keine Möglichkeit, sie wiederherzustellen. Gestoppte Simulationen können nicht neu gestartet werden. Die einzige Möglichkeit, das Äquivalent eines Neustarts durchzuführen, besteht darin, Ihre Simulation mit derselben Konfiguration und denselben Daten neu zu erstellen. Sie können Backups Ihres Simulationsstatus verwenden, um die neue Simulation zu initialisieren. AWS

bietet äußerst zuverlässige und verfügbare [Cloud-Speicher](#) - und [Datenbankdienste](#), mit denen Sie Ihren Simulationsstatus speichern können.

Pflegen Sie Ihre Apps und SDKs

Pflegen Sie Ihre Apps, lokalen Installationen der AWS Software Development Kits (SDKs) und das SimSpace Weaver App-SDK. Sie können neue Versionen von herunterladen und installieren AWS SDKs. Testen Sie neue Versionen des SimSpace Weaver App-SDK mit App-Builds, die nicht für die Produktion bestimmt sind, um sicherzustellen, dass Ihre Apps weiterhin wie erwartet ausgeführt werden. Sie können Ihre Apps in einer laufenden Simulation nicht aktualisieren. So aktualisieren Sie Ihre Apps:

1. Aktualisieren und testen Sie den App-Code lokal (oder in einer Testumgebung).
2. Beenden Sie die Änderung Ihres Simulationsstatus und speichern Sie ihn (falls erforderlich).
3. Stoppen Sie Ihre Simulation (wenn sie einmal gestoppt wurde, kann sie nicht neu gestartet werden).
4. Löschen Sie Ihre Simulation (gestoppte Simulationen, die nicht gelöscht wurden, werden auf Ihre Dienstlimits angerechnet).
5. Erstellen Sie Ihre Simulation mit derselben Konfiguration und dem aktualisierten App-Code neu.
6. Initialisieren Sie Ihre Simulation mit gespeicherten Zustandsdaten (falls verfügbar).
7. Starten Sie Ihre neue Simulation.

Note

Eine neue Simulation, die mit derselben Konfiguration erstellt wurde, ist von der alten Simulation getrennt. Es wird eine neue Simulations-ID haben und Protokolle an einen neuen Protokollstream in Amazon senden CloudWatch.

Einloggen und Überwachen SimSpace Weaver

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer SimSpace Weaver anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, mit denen Sie beobachten SimSpace Weaver, melden können, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarmer festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldaten Ihrer SimSpace Weaver Mitarbeiter und anderer Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann Informationen in den Protokolldaten überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Konto -Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im [AWS CloudTrail - Benutzerhandbuch](#).

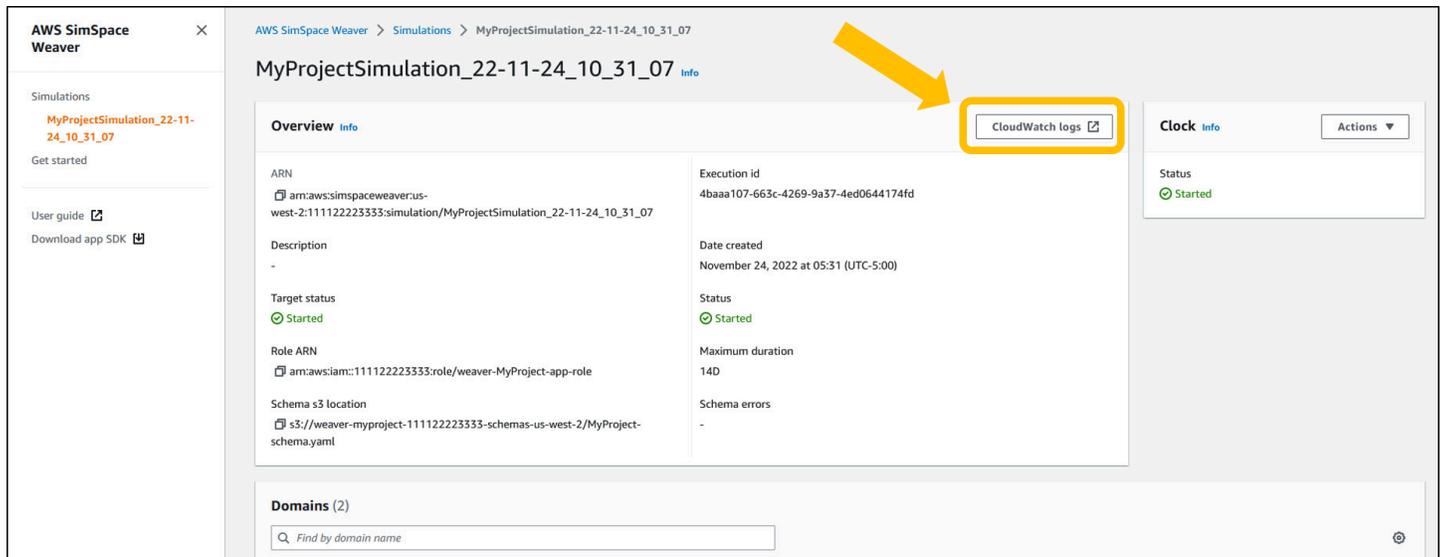
Themen

- [SimSpace Weaver loggt sich in Amazon CloudWatch Logs ein](#)
- [Überwachung SimSpace Weaver mit Amazon CloudWatch](#)
- [Protokollieren von AWS SimSpace Weaver API-Aufrufen mit AWS CloudTrail](#)

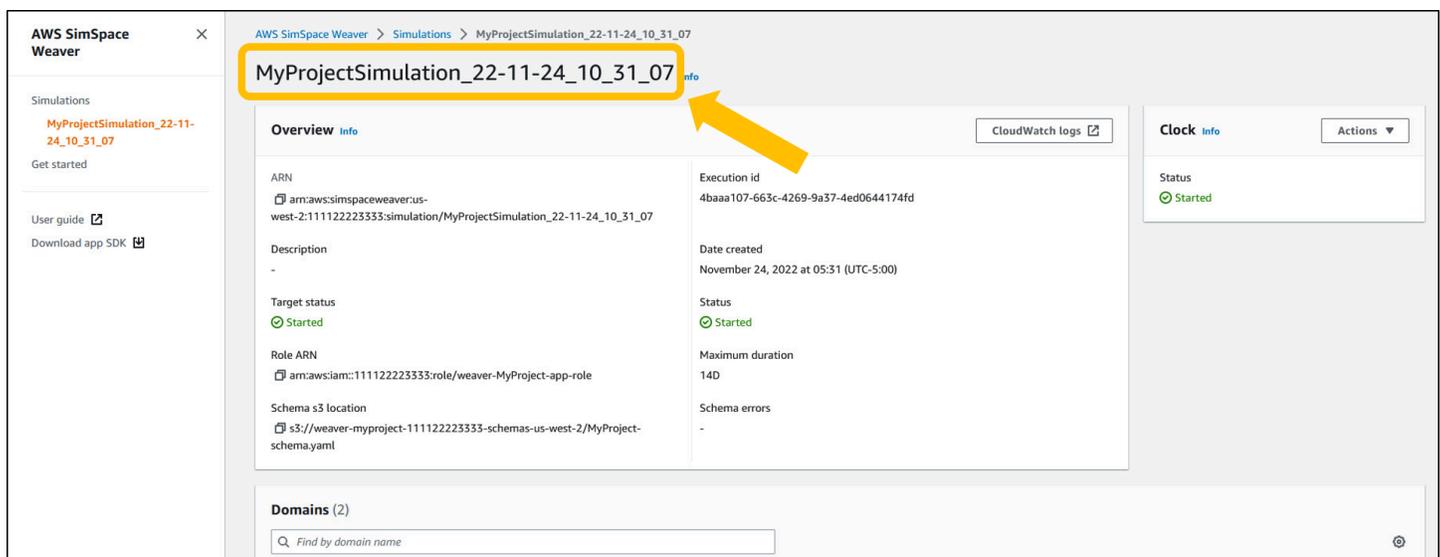
SimSpace Weaver loggt sich in Amazon CloudWatch Logs ein

Greife auf deine SimSpace Weaver Logs zu

Alle aus Ihren SimSpace Weaver Simulationen generierten Protokolle werden in Amazon CloudWatch Logs gespeichert. Um auf Ihre Protokolle zuzugreifen, können Sie die Schaltfläche „CloudWatch Protokolle“ im Übersichtsbereich Ihrer Simulation in der SimSpace Weaver Konsole verwenden, wodurch Sie direkt zu den Protokollen für diese spezifische Simulation gelangen.



Sie können auch über die CloudWatch Konsole auf die Protokolle zugreifen. Sie benötigen den Namen Ihrer Simulation, um nach ihren Protokollen suchen zu können.



SimSpace Weaver Logs

SimSpace Weaver schreibt Simulationsverwaltungsnachrichten und die Konsolenausgabe Ihrer Apps in Amazon CloudWatch Logs. Weitere Informationen zur Arbeit mit Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollströmen](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Jede Simulation, die Sie erstellen, hat ihre eigene Protokollgruppe in CloudWatch Logs. Der Name der Protokollgruppe ist im Simulationsschema angegeben. Im folgenden Schemaausschnitt ist der Wert von `log_destination_service.logs`. Das bedeutet, dass der Wert von der Name einer Protokollgruppe `log_destination_resource_name` ist. In diesem Fall ist die Protokollgruppe `MySimulationLogs`.

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

Sie können die `DescribeSimulation` API auch verwenden, um den Namen der Protokollgruppe für die Simulation zu ermitteln, nachdem Sie sie gestartet haben.

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

Das folgende Beispiel zeigt den Teil der Ausgabe von `DescribeSimulation`, der die Protokollkonfiguration beschreibt. Der Name der Protokollgruppe wird am Ende von `LogGroupArn` angezeigt.

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

Jede Simulationsprotokollgruppe enthält mehrere Protokollströme:

- Management-Log-Stream — vom SimSpace Weaver Dienst erzeugte Simulationsverwaltungsnachrichten.

```
/sim/management
```

- Fehlerprotokollstream — vom SimSpace Weaver Dienst erzeugte Fehlermeldungen. Dieser Protokollstream ist nur vorhanden, wenn Fehler vorliegen. SimSpace Weaver speichert Fehler, die von Ihren Apps geschrieben wurden, in ihren eigenen App-Protokollstreams (siehe die folgenden Protokollstreams).

```
/sim/errors
```

- Spatial App Log Streams (1 für jede Spatial App auf jedem Worker) — Konsolenausgabe, die von Spatial Apps erzeugt wird. Jede Spatial-App schreibt in ihren eigenen Log-Stream. *spatial-app-id* Das sind alle Zeichen nach dem abschließenden Schrägstrich am Ende von. *worker-id*

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

- Benutzerdefinierte App-Protokollstreams (1 für jede benutzerdefinierte App-Instanz) — Konsolenausgabe, die von benutzerdefinierten Apps erzeugt wird. Jede benutzerdefinierte Anwendungsinstanz schreibt in ihren eigenen Protokollstream.

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

- Protokollstreams von Service-Apps (1 für jede Service-App-Instanz) — Konsolenausgabe, die von Service-Apps erzeugt wird. Jede Service-App schreibt in ihren eigenen Protokollstream. *service-app-id* Das sind alle Zeichen nach dem abschließenden Schrägstrich am Ende von. *service-app-name*

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

Überwachung SimSpace Weaver mit Amazon CloudWatch

Sie können die Überwachung SimSpace Weaver mithilfe von Amazon durchführen CloudWatch. Amazon sammelt Rohdaten und verarbeitet sie zu lesbaren Metriken, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsdaten zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Der SimSpace Weaver Service meldet die folgenden Metriken im `AWS/simspaceweaver` Namespace.

SimSpace Weaver Metriken auf Kontoebene

Der SimSpace Weaver Namespace umfasst die folgenden Metriken, die sich auf Aktivitäten auf AWS Kontoebene beziehen.

Metrik	Beschreibung
<code>SimulationCount</code>	Die Anzahl der Simulationen für das Girokonto. Einheiten: Anzahl Abmessungen: keine Statistiken: Durchschnitt, Minimum, Maximum

Protokollieren von AWS SimSpace Weaver API-Aufrufen mit AWS CloudTrail

AWS SimSpace Weaver ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden SimSpace Weaver. CloudTrail erfasst alle API-Aufrufe SimSpace Weaver als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der SimSpace Weaver Konsole und Codeaufrufen für die SimSpace Weaver API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich

Ereignissen für SimSpace Weaver. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole unter anzeigen Event history. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde SimSpace Weaver, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

SimSpace Weaver Informationen in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet SimSpace Weaver, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen in einem Ereignis aufgezeichnet Event history. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter [Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für SimSpace Weaver, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Alle SimSpace Weaver Aktionen werden von der [AWS SimSpace Weaver API-Referenz](#) protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe von DescribeSimulation und DeleteSimulation Aktionen Einträge in den CloudTrail Protokolldateien. ListSimulations

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

SimSpace Weaver Logdateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, wie Datum und Uhrzeit der Aktion, Anforderungsparameter und andere Details. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `ListSimulations` Aktion demonstriert.

```
{
  "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 Weaver-Endpunkte und Kontingente

In den folgenden Tabellen werden die Dienstendpunkte und Dienstkontingente für Weaver beschrieben. SimSpace Service-Kontingente, auch als Limits bezeichnet, sind die maximale Anzahl von Serviceressourcen oder -vorgängen für Ihr AWS-Konto. Weitere Informationen finden Sie unter [AWS Service Quotas](#) in der Allgemeinen AWS -Referenz.

Service-Endpunkte

Name der Region	Region	Endpunkt	Protokoll
USA Ost (Nord-Virginia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
USA Ost (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europa (Stockholm)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Name der Region	Region	Endpoint	Protokoll
Europa (Frankfurt)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europa (Irland)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (USA-Ost)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-West)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Servicekontingente

Name	Standard	Anpas	Beschreibung
Rechenressourceneinheiten für jede Anwendung	Jede unterstützte Region: 4	Nein	Die maximale Anzahl der Rechenressourceneinheiten, die Sie jeder Anwendung zuweisen können.
Rechenressourceneinheiten für jeden Worker	Jede unterstützte Region: 17	Nein	Die Anzahl der Rechenressourceneinheiten, die für jeden Worker verfügbar sind.
Datenfelder für jede Entität	Jede unterstützte Region: 7	Nein	Die maximale Anzahl von Datenfeldern (ohne Index), über die eine Entität verfügen kann.

Name	Standard	Anpas	Beschreibung
Entitäten in einer Partition	Jede unterstützte Region: 8 192	Nein	Die maximale Anzahl von Entitäten in einer Partition .
Größe des Entitätsdatenfeldes	Jede unterstützte Region: 1 024 Byte	Nein	Die maximale Größe eines Datenfeldes (kein Index) einer Entität.
Entitätsübertragungen zwischen Workern	Jede unterstützte Region: 25	Nein	Die maximale Anzahl von Entitätsübertragungen zwischen Workern für jede Partition und jedes Tick.
Entitätsübertragungen an denselben Worker	Jede unterstützte Region: 500	Nein	Die maximale Anzahl von Entitätsübertragungen auf demselben Worker für jede Partition und jedes Tick.
Indexfelder für jede Entität	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von Indexfeldern, über die eine Entität verfügen kann.

Name	Standard	Anpas	Beschreibung
Größte maximale Dauer (in Tagen) für eine Simulation	Jede unterstützte Region: 14	Nein	Die größte Anzahl von Tagen, die Sie als maximale Dauer für eine Simulation angeben können. Alle Simulationen haben eine maximale Dauer, auch wenn Sie den Wert nicht angeben. Eine Simulation wird automatisch beendet, wenn sie ihre maximale Dauer erreicht hat.
Speicher für jede Rechenressourceneinheit	Jede unterstützte Region: 1 Gigabyte	Nein	Die Menge an Arbeitsspeicher (RAM) mit wahlfreiem Zugriff, die eine App für jede Rechenressourceneinheit erhält.
Remote-Abonnements für jeden Worker	Jede unterstützte Region: 24	Nein	Die maximale Anzahl der Remote-Abonnements für jeden Worker.
Anzahl der Simulationen	Jede unterstützte Region: 2	Ja	Die maximale Anzahl von Simulationen mit dem Zielstatus GESTARTET in Ihrem Konto. Sie können eine Erhöhung des Kontingents auf bis zu 10 beantragen.

Name	Standard	Anpassbar	Beschreibung
Worker für eine Simulation	Jede unterstützte Region: 2	Ja	Die maximale Anzahl von Mitarbeitern, die Sie einer Simulation zuweisen können. Sie können eine Erhöhung des Kontingents auf bis zu 10 beantragen.
v CPUs für jede Rechenressourceneinheit	Jede unterstützte Region: 2	Nein	Die Anzahl der virtuellen Zentraleinheiten (vCPUs), die eine App für jede Rechenressourceneinheit erhält.

Kontingente für Nachrichten

Die folgenden Kontingente gelten für App-zu-App-Messaging für SimSpace Weaver Local und in der AWS Cloud.

Name	Standard	Anpassbar	Beschreibung
Maximale Nachrichtengröße	Jede unterstützte Region: 256 Byte	Nein	Die maximale Größe einer Nachrichtennutzlast.
Maximale Senderate für Nachrichten	Jede unterstützte Region: 128	Nein	Die maximale Anzahl von Nachrichten, die jede App pro Tick senden kann.

Taktraten

Das Simulationsschema gibt die Taktrate (auch Tickrate genannt) für eine Simulation an. Die folgende Tabelle gibt die gültigen Taktraten an, die Sie verwenden können.

Name	Zulässige Werte	Beschreibung
Taktrate	Jede unterstützte Region: „10“, „15“, „30“, „unbegrenzt“	Die gültigen Taktraten für eine Simulation.
Taktrate (Versionen 1.13 und 1.12)	Jede unterstützte Region: 10, 15, 30	Die gültigen Taktraten für eine Simulation.

Servicekontingenten für SimSpace Weaver Local

Die folgenden Servicekontingenten gelten für SimSpace Weaver Local nur. Alle anderen Kontingente gelten auch für SimSpace Weaver Local.

Name	Standard	Anpassbar	Beschreibung
Maximale Anzahl an Partitionen	SimSpace Weaver Local: 24	Nein	Die maximale Anzahl von Partitionen für eine Simulation.
Maximale Anzahl an Apps	SimSpace Weaver Local: 24	Nein	Die maximale Gesamtzahl von Apps (beliebigen Typs) für eine Simulation.
Maximale Anzahl an Domänen	SimSpace Weaver Local: 24	Nein	Die maximale Gesamtzahl von Domänen (beliebigen Typs) für eine Simulation.
Entitäten pro Partition	SimSpace Weaver Local: 4.096	Nein	Die maximale Anzahl der Entitäten in jeder Partition.
Felder pro Entität	SimSpace Weaver Local: 8	Nein	Die maximale Anzahl von Feldern für jede Entität.

Name	Standard	Anpassbar	Beschreibung
Größe des Feldes	SimSpace Weaver Local: 1024 Byte	Nein	Die maximale Größe eines Entitätsfeldes.

Problembehandlung in SimSpace Weaver

Themen

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [Zugriff während des Simulationsstarts verweigert](#)
- [Probleme im Zusammenhang mit der Zeit bei der Verwendung Docker](#)
- [PathfindingSample Der Konsolenclient kann keine Verbindung herstellen](#)
- [Das AWS CLI erkennt nicht simspaceweaver](#)

AssumeRoleAccessDenied

Möglicherweise wird die folgende Fehlermeldung angezeigt, wenn Ihre Simulation nicht gestartet werden kann:

```
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
```

Sie können diesen Fehler erhalten, wenn einer der folgenden Punkte auf die AWS Identity and Access Management (IAM-) Rolle für Ihre Simulation zutrifft:

- Der Name der Amazon-Ressource (ARN) bezieht sich auf eine IAM-Rolle, die nicht existiert.
- Die Vertrauensrichtlinie für die IAM-Rolle, die nicht zulässt, dass der Name der neuen Simulation die Rolle übernimmt.

Vergewissern Sie sich, dass die Rolle vorhanden ist. Wenn die Rolle existiert, überprüfen Sie Ihre Vertrauensrichtlinie für die Rolle. Mit `aws:SourceArn` der folgenden Beispiel-Vertrauensrichtlinie kann nur eine Simulation (im Konto 111122223333), deren Name mit `beginnt`, `MySimulation` die Rolle übernehmen.

```
{
```

```

"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*"
      }
    }
  }
]
}

```

Damit eine andere Simulation, deren Name mit `MyOtherSimulation` beginnt, die Rolle übernehmen kann, muss die Vertrauensrichtlinie wie im folgenden bearbeiteten Beispiel geändert werden:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "simspaceweaver.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*",
            "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MyOtherSimulation*"
          ]
        }
      }
    }
  ]
}

```

```
}
```

InvalidBucketName

Möglicherweise wird beim Erstellen eines Projekts die folgende Fehlermeldung angezeigt:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

Sie haben diesen Fehler erhalten, weil der Name, der an Amazon Simple Storage Service (Amazon S3) SimSpace Weaver übergeben wurde, gegen die Regeln zur Benennung von Buckets verstieß (weitere Informationen finden Sie unter [Regeln zur Benennung von Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch).

Das `create-project` Skript im SimSpace Weaver App-SDK erstellt Bucket-Namen anhand des Projektnamens, den Sie dem Skript zur Verfügung stellen. Die Bucket-Namen verwenden die folgenden Formate:

- Version 1.13.x oder höher
 - `weaver-lowercase-project-name-account-number-region`
- Version 1.12.x
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

Zum Beispiel angesichts der folgenden Projekteigenschaften:

- Name des Projekts: `MyProject`
- AWS-Konto Nummer: `111122223333`
- AWS-Region: `us-west-2`

Das Projekt hätte die folgenden Bereiche:

- Version 1.13.x oder höher
 - `weaver-myproject-111122223333-us-west-2`
- Version 1.12.x

- `weaver-myproject-111122223333-app-zips-us-west-2`
- `weaver-myproject-111122223333-schemas-us-west-2`

Ihr Projektname darf nicht gegen die Amazon S3 S3-Benennungsregeln verstoßen. Sie müssen auch einen Projektnamen verwenden, der kurz genug ist, damit die vom `create-project` Skript erstellten Bucket-Namen die Längenbeschränkung für Amazon S3 S3-Buckets nicht überschreiten.

ServiceQuotaExceededException

Möglicherweise erhalten Sie die folgende Fehlermeldung, wenn Sie eine Simulation starten:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

Sie erhalten diesen Fehler, wenn Sie versuchen, eine neue Simulation zu starten, Ihr Konto jedoch derzeit die maximale Anzahl von Simulationen mit dem Zielstatus GESTARTET hat. Dazu gehören laufende Simulationen, fehlgeschlagene Simulationen und Simulationen, die beendet wurden, weil sie ihre maximale Dauer erreicht haben. Sie können eine gestoppte oder fehlgeschlagene Simulation löschen, damit Sie eine neue Simulation starten können. Wenn alle Ihre Simulationen laufen, können Sie eine laufende Simulation beenden und löschen. Sie können auch eine Erhöhung Ihrer Servicekontingenten beantragen, wenn Sie das Anforderungslimit noch nicht erreicht haben.

TooManyBuckets

Möglicherweise wird beim Erstellen eines Projekts die folgende Fehlermeldung angezeigt:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) hat ein Limit für die Anzahl der Buckets, die Sie in Ihrem AWS Konto haben können (weitere Informationen finden Sie unter [Bucket-Einschränkungen und Einschränkungen](#) im Amazon Simple Storage Service-Benutzerhandbuch).

Sie müssen einen der folgenden Schritte ausführen, bevor Sie fortfahren können:

- Löschen Sie 2 oder mehr bestehende Amazon S3 S3-Buckets, die Sie nicht benötigen.

- Beantragen Sie eine Erhöhung des Amazon S3 S3-Limits (weitere Informationen finden Sie unter [Bucket-Einschränkungen und](#) -Beschränkungen im Amazon Simple Storage Service-Benutzerhandbuch).
- Verwenden Sie ein anderes AWS Konto.

Note

Die `DeleteSimulation` API-Eingabe löscht SimSpace Weaver keine Amazon S3 S3-Ressourcen, die mit Ihrer Simulation verknüpft sind. Wir empfehlen Ihnen, alle mit Ihren Simulationen verknüpften Ressourcen zu entfernen, wenn Sie sie nicht mehr benötigen.

Zugriff während des Simulationsstarts verweigert

Wenn Sie eine Simulation starten, erhalten Sie möglicherweise eine Fehlermeldung, die darauf hinweist, dass die Berechtigung verweigert wurde oder dass beim Zugriff auf Ihre App-Artefakte ein Fehler aufgetreten ist. Dieses Problem kann auftreten, wenn Sie Amazon S3 S3-Buckets für Ihre Simulation angeben, die SimSpace Weaver nicht für Sie erstellt wurden (entweder über die Konsole oder die SimSpace Weaver App-SDK-Skripts).

Die folgenden Situationen sind die wahrscheinlichsten Hauptursachen:

- Der Service ist nicht berechtigt, auf einen oder mehrere der Amazon S3 S3-Buckets zuzugreifen, die Sie in Ihrem Simulationsschema angegeben haben. Überprüfen Sie Ihre App-Rollenberechtigungsrichtlinien, Amazon S3 S3-Bucket-Richtlinien und Amazon S3 S3-Bucket-Berechtigungen, um sicherzustellen, dass er `simspaceweaver.amazonaws.com` über die richtigen Berechtigungen für den Zugriff auf Ihre Buckets verfügt. Weitere Informationen zur Richtlinie für App-Rollenberechtigungen finden Sie unter [Berechtigungen, die für Sie SimSpace Weaver erstellt werden](#)
- Ihr Amazon S3 S3-Bucket könnte sich in einer anderen AWS-Region als Ihrer Simulation befinden — Ihre Amazon S3 S3-Buckets für Ihre Simulationsartefakte müssen sich in derselben AWS-Region wie Ihre Simulation befinden. Sehen Sie in Ihrer Amazon S3 S3-Konsole nach, worin sich AWS-Region Ihr Bucket befindet. Wenn sich Ihr Amazon S3 S3-Bucket in einem anderen befindet AWS-Region, wählen Sie einen Bucket aus, der sich in derselben AWS-Region Simulation befindet.

Probleme im Zusammenhang mit der Zeit bei der Verwendung Docker

Wenn du verwendest Docker und Sie erhalten zeitbedingte Fehler beim Ausführen von Skripten aus dem SimSpace Weaver App-SDK. Die Ursache könnte darin liegen, dass Docker Die Uhr der virtuellen Maschine ist falsch. Dies kann passieren, wenn Ihr Computer lief Docker und wird dann aus dem Standbymodus oder Ruhezustand wieder aufgenommen.

Lösungen zum Ausprobieren

- Neustart Docker.
- Deaktivieren Sie die Zeitsynchronisierung und aktivieren Sie sie dann erneut in Windows PowerShell:

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample Der Konsolencient kann keine Verbindung herstellen

Möglicherweise wird der folgende Fehler vom Konsolencient angezeigt, wenn Sie eine Verbindung zu der PathfindingSample Simulation herstellen, die in den Tutorials unter beschrieben ist [Erste Schritte mit SimSpace Weaver](#). Dieser Fehler tritt auf, weil der Client ViewApp unter der von Ihnen angegebenen kombinierten IP-Adresse und Portnummer keine Netzwerkverbindung herstellen kann.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Für eine Simulation in AWS Cloud

- Funktioniert Ihre Netzwerkverbindung korrekt? Stellen Sie sicher, dass Sie eine Verbindung zu anderen IP-Adressen oder Websites herstellen können, die funktionieren sollten. Stellen Sie sicher, dass Ihr Webbrowser keine Website aus dem Cache lädt.
- Läuft Ihre Simulation? Sie können die ListSimulationsAPI verwenden, um den Status Ihrer Simulation abzurufen. Weitere Informationen finden Sie unter [Rufen Sie die IP-Adresse und die](#)

[Portnummer einer benutzerdefinierten App ab](#). Sie können auch die [SimSpace Weaver Konsole](#) verwenden, um den Status Ihrer Simulationen zu überprüfen.

- Laufen Ihre Apps? Sie können die DescribeAppAPI verwenden, um den Status Ihrer Apps abzurufen. Weitere Informationen finden Sie unter [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#). Sie können die [SimSpace Weaver Konsole](#) auch verwenden, um den Status Ihrer Simulationen zu überprüfen.
- Laufen Ihre Apps? Sie können die DescribeAppAPI verwenden, um den Status Ihrer Apps abzurufen. Weitere Informationen finden Sie unter [Rufen Sie die IP-Adresse und die Portnummer einer benutzerdefinierten App ab](#). Sie können die [SimSpace Weaver Konsole](#) auch verwenden, um den Status Ihrer Simulationen zu überprüfen.
- Haben Sie die richtige IP-Adresse und Portnummer verwendet? Wenn Sie eine Verbindung über das Internet herstellen, müssen Sie die IP-Adresse und die Actual Portnummer von verwendenViewApp. Sie finden die IP Address und die Actual Portnummer im EndpointInfo Block der DescribeAppAPI-Ausgabe. Sie können die [SimSpace Weaver Konsole](#) auch verwenden, um die IP-Adresse (URI) und die Portnummer (Eingangsport) für Sie ViewApp auf der MyViewDomain Detailseite zu finden.
- Durchläuft Ihre Netzwerkverbindung eine Firewall? Ihre Firewall blockiert möglicherweise Ihre Verbindung zur IP-Adresse oder Portnummer (oder zu beiden). Überprüfen Sie Ihre Firewall-Einstellungen oder wenden Sie sich an Ihren Firewall-Administrator.

Für eine lokale Simulation

- Können Sie eine Verbindung zu Ihrer Loopback-Adresse (127.0.0.1) herstellen? Wenn Sie das ping Befehlszeilentool in Windows haben, können Sie ein Befehlszeilenfenster öffnen und versuchen, 127.0.0.1 zu ping. Drücken Sie Ctrl -, C um den Ping zu beenden.

```
ping 127.0.0.1
```

Example Ping-Ausgabe

```
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:\>
```

Wenn der Ping-Befehl besagt, dass Pakete verloren gegangen sind, blockiert möglicherweise eine andere Software (z. B. eine lokale Firewall, Sicherheitseinstellungen oder Anti-Malware-Programme) Ihre Verbindung.

- Laufen Ihre Apps? Ihre lokale Simulation wird für jede App in separaten Fenstern ausgeführt. Stellen Sie sicher, dass die Fenster für Ihre Geo-Apps geöffnet ViewApp sind. Weitere Informationen finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).
- Haben Sie die richtige IP-Adresse und Portnummer verwendet? Sie müssen dies verwendet `cp://127.0.0.1:7000`, wenn Sie eine Verbindung zu einer lokalen Simulation herstellen. Weitere Informationen finden Sie unter [Lokale Entwicklung in SimSpace Weaver](#).
- Haben Sie lokale Sicherheitssoftware, die Ihre Verbindung blockieren könnte? Überprüfen Sie Ihre Sicherheitseinstellungen, Ihre lokale Firewall oder Anti-Malware-Programme, um festzustellen, ob sie Ihre Verbindung zu `127.0.0.1` einem TCP-Port `7000` blockieren.

Das AWS CLI erkennt nicht **simspaceweaver**

Wenn das AWS CLI Ihnen Fehler anzeigt, die darauf hindeuten, dass es nichts davon weiß SimSpace Weaver, führen Sie den folgenden Befehl aus.

```
aws simspaceweaver help
```

Wenn Sie eine Fehlermeldung erhalten, die mit den folgenden Zeilen beginnt und alle verfügbaren Optionen auflistet, handelt AWS CLI es sich möglicherweise um eine ältere Version.

```
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:
```

Führen Sie den folgenden Befehl aus, um die Version Ihres zu überprüfen AWS CLI.

```
aws --version
```

Wenn die Versionsnummer älter als 2.9.19 ist, müssen Sie Ihre aktualisieren. AWS CLI Beachten Sie, dass die aktuelle Version von neuer als AWS CLI 2.9.19 ist.

Informationen zum Aktualisieren Ihrer AWS CLI finden [Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch für Version 2.

SimSpace Weaver Referenz zum Simulationsschema

SimSpace Weaver verwendet eine YAML-Datei, um die Eigenschaften einer Simulation zu konfigurieren. Diese Datei wird Simulationsschema (oder einfach Schema) genannt. Die im SimSpace Weaver App-SDK enthaltene Beispielsimulation enthält ein Schema, das Sie für Ihre eigene Simulation kopieren und bearbeiten können.

Themen

- [Beispiel für ein vollständiges Schema](#)
- [Schemaformat](#)

Beispiel für ein vollständiges Schema

Das folgende Beispiel zeigt die YAML-formatierte Textdatei, die eine SimSpace Weaver Simulation beschreibt. Dieses Beispiel enthält Dummy-Werte für die Eigenschaften. Das Format der Datei variiert je nach dem in der Datei `sdk_version` angegebenen Wert von. Eine [Schemaformat](#) vollständige Beschreibung der Eigenschaften und ihrer gültigen Werte finden Sie unter.

```
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"]
```

Schemaformat

Das folgende Beispiel zeigt die Gesamtstruktur eines Schemas. Die Reihenfolge der Eigenschaften auf jeder Ebene des Schemas spielt keine Rolle, solange die Beziehungen zwischen Eltern und Kindern identisch sind. Die Reihenfolge ist wichtig für Elemente in einem 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
```

Sections

- [SDK-Version](#)
- [Eigenschaften der Simulation](#)
- [Worker](#)
- [Uhr](#)
- [Strategien zur Partitionierung](#)
- [Domains](#)
- [Einschränkungen bei der Platzierung](#)

SDK-Version

Der `sdk_version` Abschnitt (erforderlich) identifiziert die Version des SimSpace Weaver App-SDK, die dieses Schema unterstützt. Zulässige Werte: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

Der Wert von `beinhaltet` `sdk_version` nur die Hauptversionsnummer und die erste Nebenversionsnummer. Der Wert 1.12 gibt beispielsweise alle Versionen an 1.12.x, wie 1.12.01.12.1, und 1.12.2.

```
sdk_version: "1.17"
```

Eigenschaften der Simulation

Der `simulation_properties` Abschnitt (erforderlich) spezifiziert verschiedene Eigenschaften Ihrer Simulation. Verwenden Sie diesen Abschnitt, um die Protokollierung zu konfigurieren und ein Standard-Container-Image anzugeben. Dieser Abschnitt ist auch dann erforderlich, wenn Sie die Protokollierung nicht konfigurieren oder ein Standard-Container-Image angeben.

```
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"
```

Eigenschaften

`log_destination_resource_name`

Gibt die Ressource an, in die Protokolle geschrieben SimSpace Weaver werden.

Erforderlich: Nein. Wenn diese Eigenschaft nicht enthalten ist, SimSpace Weaver werden keine Protokolle für die Simulation geschrieben.

Typ: Zeichenfolge

Zulässige Werte:

- Der Name einer CloudWatch Logs-Protokollgruppe (zum Beispiel `MySimulationLogs`)
- Der Amazon-Ressourcenname (ARN) einer CloudWatch Logs-Protokollgruppe (z. B. `arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`)

 Note

SimSpace Weaver unterstützt nur ein Protokollziel im selben Konto und AWS-Region wie in der Simulation.

`log_destination_service`

Gibt den Typ der Protokollierungszielressource an, wenn Sie eine angeben `logging_destination_resource_name`, die kein ARN ist.

Erforderlich: Sie müssen diese Eigenschaft angeben, wenn die angegeben `log_destination_resource_name` ist und es sich nicht um einen ARN handelt. Sie können diese Eigenschaft nicht angeben, wenn sie `log_destination_resource_name` nicht angegeben ist oder es sich um einen ARN handelt.

Typ: Zeichenfolge

Zulässige Werte:

- `logs`: Die Protokollzielressource ist eine Protokollgruppe.

`default_entity_index_key_type`

Gibt den Datentyp für das Indexschlüselfeld von Simulationsentitäten an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: `Vector<f32>`

`default_image`

Gibt das Standard-Container-Image für Ihre Simulation an (wird für Version 1.13 und nicht unterstützt 1.12). Wenn diese Eigenschaft angegeben ist, `image` verwenden Domänen, die keinen angeben, den `default_image`.

Required: No

Typ: Zeichenfolge

Gültige Werte:

- Die URI eines Repositorys in Amazon Elastic Container Registry (Amazon ECR) (z. B. `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Worker

Der `workers` Abschnitt (erforderlich) spezifiziert Konfigurationen für Arbeitergruppen (Gruppen von Arbeitern). SimSpace Weaver verwendet diese Informationen zusammen mit `placement_constraints`, um die Ihrer Simulation zugrunde liegende Infrastruktur zu konfigurieren. Derzeit wird nur eine Worker-Gruppe unterstützt.

Um die Eigenschaften für eine Workergruppe anzugeben, *worker-group-name* ersetzen Sie sie durch einen Namen Ihrer Wahl. Der Name muss 3-64 Zeichen lang sein und kann die Zeichen A-Z, a-z, 0-9 und `_` (Bindestrich) enthalten. Geben Sie nach dem Namen die Eigenschaften der Worker-Gruppe an.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Eigenschaften

type

Gibt den Arbeitstyp an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: `sim.c5.24xlarge`

desired

Gibt die gewünschte Anzahl von Arbeitskräften für diese Arbeitergruppe an.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1 -3. Ihr Servicekontingent (Limit) für die Anzahl der Mitarbeiter für Ihre Simulationen bestimmt den Höchstwert dieser Eigenschaft. Wenn Ihre Servicequote beispielsweise ist, 2 dann ist der Höchstwert für diese Eigenschaft2. Sie können eine Erhöhung Ihres Servicekontingents beantragen. Weitere Informationen finden Sie unter [SimSpace Weaver-Endpunkte und Kontingente](#).

Uhr

Der `clock` Abschnitt (erforderlich) spezifiziert die Eigenschaften der Simulationsuhr.

```
clock:  
  tick_rate: tick-rate
```

Eigenschaften

`tick_rate`

Gibt die Anzahl der Ticks pro Sekunde an, die die Uhr in Apps veröffentlicht.

Erforderlich: Ja

Type:

- Version 1.14 und 1.15: Zeichenfolge
- Version 1.13 und 1.12: Integer

Zulässige Werte:

- Version 1.14 und 1.15: "10" | "15" | "30" | "unlimited"
 - "unlimited": Die Uhr sendet den nächsten Tick, sobald alle Apps ihre Commit-Operationen für den aktuellen Tick abgeschlossen haben.
- Version 1.13 und 1.12: 10 | 15 | 30

Strategien zur Partitionierung

Der `partitioning_strategies` Abschnitt (erforderlich) spezifiziert die Organisation der Partitionen für eine räumliche Domäne.

Note

SimSpace Weaver unterstützt nur eine Partitionierungsstrategie.

Um die Eigenschaften für eine Partitionierungsstrategie anzugeben, *partitioning-strategy-name* ersetzen Sie sie durch einen Namen Ihrer Wahl. Der Name muss 3-64 Zeichen lang sein und kann die Zeichen A-Z, a-z, 0-9 und _ - (Bindestrich) enthalten. Geben Sie nach dem Namen die Eigenschaften der Partitionierungsstrategie an.

```
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
```

Eigenschaften

topology

Gibt die Topologie (Partitionsanordnungsschema) für diese Partitionierungsstrategie an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: "Grid"

aabb_bounds

Gibt die Grenzen des an der Hauptachse ausgerichteten Begrenzungsrahmens an (AABB) für Ihre Simulation. Sie geben die Grenzen als aus zwei Elementen geordnete Arrays an, die die Minimal- und Maximalwerte (in dieser Reihenfolge) für jede Achse beschreiben (x and y).

Erforderlich: Bedingt. Diese Eigenschaft ist erforderlich (und kann nur angegeben werden), wenn die Topologie auf eingestellt ist. "Grid"

Typ: Float Array (für jede Achse)

Gültige Werte: $-3.4028235e38 - 3.4028235e38$

grid_placement_groups

Gibt die Anzahl der Platzierungsgruppen entlang jeder Achse (X und Y) in einer Rastertopologie an. Eine Platzierungsgruppe ist eine Sammlung von Partitionen (in derselben Domäne), die räumlich aneinander grenzen.

Erforderlich: Bedingt. Diese Eigenschaft ist erforderlich (und kann nur angegeben werden), wenn die Topologie auf eingestellt ist. "Grid" Wenn Sie keine Platzierungsgruppenkonfiguration angeben, SimSpace Weaver wird eine für Sie berechnet. Jede Domain, die eine Partitionierungsstrategie ohne Platzierungsgruppenkonfiguration verwendet, muss eine angeben `grid_partition` (siehe [Strategie zur Partitionierung räumlicher Domänen](#)).

Typ: Integer (für jede Achse)

Gültige Werte: 1 -20. Wir empfehlen, dass $x * y$ der gewünschten Anzahl von Arbeitern entspricht. Andernfalls SimSpace Weaver wird versucht, Ihre Platzierungsgruppen auf die verfügbaren Arbeitskräfte aufzuteilen.

Domains

Der `domains` Abschnitt (erforderlich) spezifiziert die Eigenschaften für jede Ihrer Domains. Alle Simulationen müssen mindestens einen Abschnitt für eine räumliche Domäne enthalten. Sie können mehrere Abschnitte für zusätzliche Domänen erstellen. Jeder Domaintyp hat sein eigenes Konfigurationsformat.

Important

Versionen 1.13 und unterstützen 1.12 nicht mehrere räumliche Domänen.

Important

SimSpace Weaver unterstützt bis zu 5 Domänen für jede Simulation. Dies umfasst alle räumlichen, benutzerdefinierten und Dienstdomänen.

```
domains:
```

```

domain-name:
  domain-configuration
domain-name:
  domain-configuration
...

```

Konfiguration der Domäne

- [Konfiguration der räumlichen Domäne](#)
- [Benutzerdefinierte Domain-Konfiguration](#)
- [Konfiguration der Service-Domain](#)

Konfiguration der räumlichen Domäne

Um die Eigenschaften für eine räumliche Domäne anzugeben, *spatial-domain-name* ersetzen Sie sie durch einen Namen Ihrer Wahl. Der Name muss 3-64 Zeichen lang sein und kann die Zeichen A-Z, a-z, 0-9 und _ - (Bindestrich) enthalten. Geben Sie nach dem Namen die Eigenschaften der räumlichen Domäne an.

```

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"

```

Strategie zur Partitionierung räumlicher Domänen

Der `launch_apps_by_partitioning_strategy` Abschnitt (erforderlich) spezifiziert die Partitionierungsstrategie und die Dimensionen (in Anzahl der Partitionen) des Simulationsraums.

```

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*

Eigenschaften

partitioning_strategy

Gibt die Partitionierungsstrategie für diese räumliche Domäne an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Der Wert dieser Eigenschaft muss mit dem Namen einer im Abschnitt definierten Partitionierungsstrategie übereinstimmen. `partitioning_strategies` Weitere Informationen finden Sie unter [Strategien zur Partitionierung](#).

grid_partition

Gibt die Anzahl der Partitionen entlang jeder Achse (X und Y) in einer Rastertopologie an. Diese Dimensionen beschreiben den gesamten Simulationsraum für diese Domäne.

Erforderlich: Bedingt. Diese Eigenschaft kann nur angegeben werden, wenn die Topologie auf "Grid" eingestellt ist. Diese Eigenschaft hängt von der `grid_placement_groups` Eigenschaft der angegebenen Partitionierungsstrategie für diese Domäne ab:

- Diese Eigenschaft ist erforderlich, wenn die Partitionierungsstrategie dieser Domain keine Konfiguration spezifiziert. `grid_placement_groups`
- Wenn es eine `grid_placement_groups` Konfiguration gibt, die Sie aber nicht angeben `grid_partition`, SimSpace Weaver werden dieselben Dimensionen wie in der `grid_placement_groups` Konfiguration verwendet.
- Wenn Sie `grid_placement_groups` sowohl als auch angeben `grid_partition`, `grid_partition` müssen die Abmessungen von ein Vielfaches der Abmessungen von sein `grid_placement_groups` (wenn Ihre `grid_placement_groups` Abmessungen beispielsweise 2x2 sind, dann sind einige gültige Abmessungen für `grid_partition` 2x2, 4x4, 6x6, 8x8, 10x10).

Typ: Integer (für jede Achse)

Gültige Werte: 1 - 20

Räumliche App-Konfiguration

Der `app_config` Abschnitt (erforderlich) spezifiziert das Paket, die Startkonfiguration und die Ressourcenanforderungen für Apps in dieser Domäne.

```
app_config:  
  package: "app-package-s3-uri"  
  launch_command: ["app-launch-command", "parameter1", ...]  
  required_resource_units:  
    compute: app-resource-units
```

Eigenschaften

package

Gibt das Paket (Zip-Datei) an, das die ausführbare/binäre Datei der App enthält. Das Paket muss in einem Amazon S3 S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Die Amazon S3 S3-URI des Pakets in einem Amazon S3 S3-Bucket. Beispiel, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

launch_command

Gibt den Namen der ausführbaren/binären Datei und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolge-Token ist ein Element im Array.

Erforderlich: Ja

Typ: String-Array

required_resource_units

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instanz dieser App zugewiesen werden SimSpace Weaver sollen. Eine Ressourceneinheit ist eine feste Anzahl virtueller Zentraleinheiten (vCPUs) und Direktzugriffsspeicher (RAM) bei einem Arbeiter. Weitere Hinweise zu Ressourceneinheiten finden Sie unter [Endpunkte und Servicekontingenten](#). Die `compute` Eigenschaft gibt eine Zuordnung von Ressourceneinheiten für die `compute` Familie der Arbeitnehmer an und ist derzeit die einzig gültige Zuweisungsart.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1 - 4

Benutzerdefiniertes Container-Image

Die `image` Eigenschaft (optional) gibt den Speicherort eines Container-Images an, das zum Ausführen von Apps in dieser Domain SimSpace Weaver verwendet wird (wird in den Versionen 1.13 und nicht unterstützt 1.12). Geben Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) an, das das Bild enthält. Wenn diese Eigenschaft nicht angegeben ist, aber die im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain `diedefault_image`. Weitere Informationen finden Sie unter [Maßgeschneiderte Behälter](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

`image`

Gibt den Speicherort eines Container-Images an, um Apps in dieser Domain auszuführen.

Required: No

Typ: Zeichenfolge

Gültige Werte:

- Die URI eines Repositorys in Amazon Elastic Container Registry (Amazon ECR) (z. B. `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Benutzerdefinierte Domain-Konfiguration

Um die Eigenschaften für eine benutzerdefinierte Domain anzugeben, *custom-domain-name* ersetzen Sie sie durch einen Namen Ihrer Wahl. Der Name muss 3-64 Zeichen lang sein und kann die Zeichen A-Z, a-z, 0-9 und `_` - (Bindestrich) enthalten. Geben Sie nach dem Namen die

Eigenschaften der benutzerdefinierten Domäne an. Wiederholen Sie diesen Vorgang für jede benutzerdefinierte Domain.

```
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"
```

Eigenschaften

launch_apps_via_start_app_call

Diese Eigenschaft ist erforderlich, um Ihre benutzerdefinierten Apps mit dem zu starten StartApp API.

Erforderlich: Ja

Typ: N/A

Gültige Werte: {}

Benutzerdefinierte App-Konfiguration

Die `app_config` section (erforderlich) spezifiziert das Paket, die Startkonfiguration, die Ressourcenanforderungen und die Netzwerkports für Apps in dieser benutzerdefinierten Domäne.

```
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, ...]
```

Eigenschaften

package

Gibt das Paket (Zip-Datei) an, das die ausführbare/binäre Datei der App enthält. Das Paket muss in einem Amazon S3 S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Die Amazon S3 S3-URI des Pakets in einem Amazon S3 S3-Bucket. Beispiel, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

launch_command

Gibt den Namen der ausführbaren/binären Datei und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolge-Token ist ein Element im Array.

Erforderlich: Ja

Typ: String-Array

required_resource_units

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instanz dieser App zugewiesen werden SimSpace Weaver sollen. Eine Ressourceneinheit ist eine feste Anzahl virtueller Zentraleinheiten (vCPUs) und Direktzugriffsspeicher (RAM) bei einem Arbeiter. Weitere Hinweise zu Ressourceneinheiten finden Sie unter [Endpunkte und Servicekontingenten](#). Die `compute` Eigenschaft gibt eine Zuordnung von Ressourceneinheiten für die `compute` Familie der Arbeitnehmer an und ist derzeit die einzig gültige Zuweisungsart.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1 - 4

endpoint_config

Gibt die Netzwerkendpunkte für Apps in dieser Domäne an. Der Wert von `ingress_ports` gibt die Ports an, an die Ihre benutzerdefinierten Apps für eingehende Clientverbindungen gebunden sind. SimSpace Weaver ordnet dynamisch zugewiesene Ports Ihren angegebenen Eingangsports zu. Eingangsports sind sowohl TCP als auch UDP. Verwenden Sie die `DescribeApp` API, um die tatsächliche Portnummer für die Verbindung Ihrer Clients zu ermitteln.

Erforderlich: Nein. Wenn Sie keine Endpunktconfiguration angeben, haben Ihre benutzerdefinierten Apps in dieser Domain keine Netzwerkendpunkte.

Typ: Integer-Array

Gültige Werte: 1024 -49152. Werte müssen eindeutig sein.

Benutzerdefiniertes Container-Image

Die `image` Eigenschaft (optional) gibt den Speicherort eines Container-Images an, das zum Ausführen von Apps in dieser Domain SimSpace Weaver verwendet wird (wird in den Versionen 1.13 und nicht unterstützt 1.12). Geben Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) an, das das Bild enthält. Wenn diese Eigenschaft nicht angegeben ist, aber die im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain die `default_image`. Weitere Informationen finden Sie unter [Maßgeschneiderte Behälter](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

`image`

Gibt den Speicherort eines Container-Images an, um Apps in dieser Domain auszuführen.

Required: No

Typ: Zeichenfolge

Gültige Werte:

- Die URI eines Repositories in Amazon Elastic Container Registry (Amazon ECR) (z. B. `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Konfiguration der Service-Domain

Um die Eigenschaften für eine Dienstdomäne anzugeben, `service-domain-name` ersetzen Sie sie durch einen Namen Ihrer Wahl. Der Name muss 3-64 Zeichen lang sein und kann die Zeichen

A-Z, a-z, 0-9 und `_` - (Bindestrich) enthalten. Geben Sie nach dem Namen die Eigenschaften der Dienstdomäne an. Wiederholen Sie diesen Vorgang für jede Dienstdomäne.

```
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"
```

Starten Sie Apps pro Mitarbeiter

Der `launch_apps_per_worker` Abschnitt (erforderlich) gibt an, dass es sich um eine Dienstdomänenkonfiguration handelt, und gibt die Anzahl der Service-Apps an, die pro Worker gestartet werden sollen.

```
launch_apps_per_worker:
  count: number-of-apps-to-launch
```

Eigenschaften

count

Diese Eigenschaft gibt die Anzahl der Service-Apps an, die pro Worker gestartet werden sollen.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: `{}` | 1 | 2. Der Wert von `{}` gibt den Standardwert von an1.

Konfiguration der Service-App

Die `app_config` section (erforderlich) spezifiziert das Paket, die Startkonfiguration, die Ressourcenanforderungen und die Netzwerkports für Apps in dieser Dienstdomäne.

```
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, ...]
```

Eigenschaften

package

Gibt das Paket (Zip-Datei) an, das die ausführbare/binäre Datei der App enthält. Das Paket muss in einem Amazon S3 S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Die Amazon S3 S3-URI des Pakets in einem Amazon S3 S3-Bucket. Beispiel, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

launch_command

Gibt den Namen der ausführbaren/binären Datei und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolge-Token ist ein Element im Array.

Erforderlich: Ja

Typ: String-Array

required_resource_units

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instanz dieser App zugewiesen werden SimSpace Weaver sollen. Eine Ressourceneinheit ist eine feste Anzahl virtueller Zentraleinheiten (vCPUs) und Direktzugriffsspeicher (RAM) bei einem Arbeiter. Weitere Hinweise zu Ressourceneinheiten finden Sie unter [Endpunkte und Servicekontingenten](#). Die `compute` Eigenschaft gibt eine Zuordnung von Ressourceneinheiten für die `compute` Familie der Arbeitnehmer an und ist derzeit die einzig gültige Zuweisungsart.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1 - 4

endpoint_config

Gibt die Netzwerkendpunkte für Apps in dieser Domäne an. Der Wert von `ingress_ports` gibt die Ports an, an die Ihre Service-Apps für eingehende Clientverbindungen gebunden sind. SimSpace Weaver ordnet dynamisch zugewiesene Ports Ihren angegebenen Eingangsports zu. Eingangsports sind sowohl TCP als auch UDP. Verwenden Sie die `DescribeApp` API, um die tatsächliche Portnummer für die Verbindung Ihrer Clients zu ermitteln.

Erforderlich: Nein. Wenn Sie keine Endpunktkonfiguration angeben, haben Ihre Service-Apps in dieser Domain keine Netzwerkendpunkte.

Typ: Integer-Array

Gültige Werte: 1024 -49152. Werte müssen eindeutig sein.

Benutzerdefiniertes Container-Image

Die `image` Eigenschaft (optional) gibt den Speicherort eines Container-Images an, das zum Ausführen von Apps in dieser Domain SimSpace Weaver verwendet wird (wird in den Versionen 1.13 und nicht unterstützt 1.12). Geben Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) an, das das Bild enthält. Wenn diese Eigenschaft nicht angegeben ist, aber die im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain `default_image`. Weitere Informationen finden Sie unter [Maßgeschneiderte Behälter](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

image

Gibt den Speicherort eines Container-Images an, um Apps in dieser Domain auszuführen.

Required: No

Typ: Zeichenfolge

Gültige Werte:

- Die URI eines Repositorys in Amazon Elastic Container Registry (Amazon ECR) (z. B. `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Einschränkungen bei der Platzierung

`placement_constraints` In diesem Abschnitt (optional) wird angegeben, welche räumlichen Domänen zusammen auf denselben Workern platziert werden SimSpace Weaver sollen. Weitere Informationen finden Sie unter [Konfiguration von räumlichen Domänen](#).

Important

Versionen 1.13 und werden 1.12 nicht unterstützt `placement_constraints`.

```
placement_constraints:  
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]  
    on_workers: ["worker-group-name"]
```

Eigenschaften

`placed_together`

Gibt die räumlichen Domänen an, die zusammen platziert SimSpace Weaver werden sollen.

Erforderlich: Ja

Typ: Zeichenketten-Array

Gültige Werte: Namen der im Schema angegebenen räumlichen Domänen

`on_workers`

Gibt die Arbeitsgruppe an, der die Domänen zugeordnet werden SimSpace Weaver sollen.

Erforderlich: Ja

Typ: Zeichenkettenarray mit einem Element

Gültige Werte: Name einer im Schema angegebenen Worker-Gruppe

SimSpace Weaver API-Referenzen

SimSpace Weaver hat 2 verschiedene Gruppen von Anwendungsprogrammierschnittstellen (APIs):

- Service APIs — Diese APIs steuern den Service und die Serviceresourcen wie Ihre Simulationen, Uhren und Apps. Sie sind Teil des wichtigsten AWS Software Development Kits (SDK) und können über die AWS Befehlszeilenschnittstelle (CLI) aufgerufen werden. Weitere Informationen zum Dienst APIs finden Sie in der [SimSpace Weaver API-Referenz](#).
- App SDK APIs — Diese APIs steuern die Daten in Ihrer Simulation. Sie verwenden sie in Ihrem App-Code, um beispielsweise Entitätsfelddaten zu lesen und zu schreiben, mit Abonnements zu arbeiten und Ereignisse in der Simulation zu überwachen. Weitere Informationen finden Sie in der SimSpace Weaver App-SDK-Dokumentation in Ihrem App-SDK-Ordner: *sdk-folder*\SimSpaceWeaverAppSdk\documentation

Note

sdk-folder ist der Ordner, in den Sie das SimSpaceWeaverAppSdkDistributable Paket entpackt haben.

AWS SimSpace Weaver Versionen

Wir verbessern uns kontinuierlich AWS SimSpace Weaver. Sie müssen das neueste SimSpace Weaver App-SDK herunterladen, wenn wir eine neue Version veröffentlichen, wenn Sie die Vorteile neuer Funktionen und Funktionsupdates nutzen möchten. Um eine bestehende Simulation mit einer neueren Version auszuführen, müssen Sie möglicherweise ihr Schema und ihren Code aktualisieren und dann eine neue Instanz der Simulation starten. Sie müssen kein Upgrade durchführen und können bestehende Simulationen weiterhin mit früheren Versionen ausführen. Auf dieser Seite können Sie nachsehen, was sich zwischen den Versionen unterscheidet. Alle Versionen werden derzeit unterstützt.

Important

Die neueste Version des [AWS SimSpace Weaver Benutzerhandbuchs](#) deckt nur die neueste Version des Dienstes ab. Die Dokumentation für frühere Versionen finden Sie im [AWS SimSpace Weaver Guide-Katalog](#), der auf der [Landingpage der Hauptdokumentation](#) verfügbar ist.

Aktuelle Version

Die neueste Version ist: 1.17.0

So finden Sie Ihre aktuelle Version

Wenn Sie eine Simulation mit dem SimSpace Weaver App-SDK erstellt haben, hat das `create-project` Skript eine Version der SDK-Bibliotheken in ein Unterverzeichnis in Ihrem heruntergeladenen *sdk-folder*. Das Unterverzeichnis, das die SDK-Bibliotheken enthält, hat einen Namen, der die SDK-Versionsnummer enthält: `SimSpaceWeaverAppSdk-sdk-version`. Die Bibliotheken für Version 1.16.0 befinden sich beispielsweise in `SimSpaceWeaverAppSdk-1.16.0`.

Sie finden die Version des verteilbaren SimSpace Weaver App-SDK-Pakets auch in der Textdatei `app_sdk_distributable_version.txt` in Ihrem *sdk-folder*.

Laden Sie die neueste Version herunter

Verwenden Sie einen der folgenden Links, um die neueste Version herunterzuladen.

- [Vollständiges, verteilbares App-SDK-Paket](#)
- [Nur die App-SDK-Bibliotheken](#)

Sie können auch das komplette verteilbare SimSpace Weaver App-SDK-Paket von der [SimSpace Weaver Konsole](#) im AWS Management Console herunterladen. Wählen Sie im Navigationsbereich die Option App-SDK herunterladen aus.

Warning

Verwenden Sie das nicht AWS CLI , um etwas herunterzuladen, das so aussieht, als ob es sich um das verteilbare SimSpace Weaver App-SDK-Paket handelt. Verwenden Sie nur die Download-Links auf dieser Seite oder den Download-Link in der Konsole. Andere Download-Methoden oder -Verzeichnisse werden nicht unterstützt und enthalten möglicherweise veralteten, falschen oder bösartigen Code.

Problembehandlung bei App-SDK-Downloads

Wir verwenden Amazon CloudFront (CloudFront), um die App-SDK-ZIP-Dateien zu verteilen. Möglicherweise treten einige der folgenden Situationen auf.

- Das heruntergeladene Paket ist nicht die neueste Version
 - Wenn die heruntergeladene ZIP-Datei nicht die neueste Version enthält, ist es möglich, dass der Cache an Ihrem CloudFront Edge-Standort noch nicht aktualisiert wurde. Laden Sie es nach 24 Stunden erneut herunter.
- Sie erhalten einen HTTP 4xx- oder 5xx-Fehler, wenn Sie einen Download-Link verwenden
 - Versuchen Sie es nach 24 Stunden erneut. Wenn Sie denselben Fehler erhalten, verwenden Sie den Feedback-Link unten in der [SimSpace Weaver Konsole](#), um das Problem zu melden. Wählen Sie als Art des Feedbacks ein Problem melden aus.
- Ihr Browser meldet, dass er die Seite nicht laden kann
 - Möglicherweise liegt ein Problem mit der lokalen Netzwerk- oder Browserkonfiguration vor. Stellen Sie sicher, dass Sie andere Seiten laden können. Leeren Sie Ihren Browser-Cache und versuchen Sie es erneut. Stellen Sie sicher, dass Sie keine Firewallregeln haben, die die Download-URL blockieren könnten.
- Beim Versuch, die Datei zu speichern, wird eine Fehlermeldung angezeigt

- Überprüfen Sie Ihre lokalen Dateisystemberechtigungen, um sicherzustellen, dass Sie über die richtigen Berechtigungen zum Speichern der Datei verfügen.
- Ihr Browser zeigt AccessDenied
 - Wenn Sie die URL manuell in Ihren Browser eingegeben haben, überprüfen Sie, ob sie korrekt ist. Wenn Sie einen Download-Link verwendet haben, stellen Sie sicher, dass die URL in Ihrem Browser nicht beeinträchtigt wurde. Verwenden Sie den Link erneut.

Installieren Sie die neueste Version

Installieren der neuesten Version

1. [Laden Sie die neueste Version](#) herunter.
2. Entpacken Sie die SimSpaceWeaverAppSdkDistributable ZIP-Datei in einen Ordner.
3. Führen Sie es `python setup.py` aus dem entpackten App-SDK-Ordner der neuesten Version SimSpace Weaver aus.
4. Verwenden Sie den entpackten SimSpace Weaver App-SDK-Ordner der neuesten Version anstelle der vorherigen Version.

Versionen der Dienste

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.17.0	<p>Wesentliche Änderungen am verteilbaren SimSpace Weaver App-SDK-Paket</p> <ul style="list-style-type: none"> • Wir haben die Windows-Batch- und Linux-Bash-Skripte durch Python-basierte Skripte 	17. April 2024	Dieser Leitfaden	<ul style="list-style-type: none"> • Komplettes Paket • Nur Bibliotheken <p>Siehe Problemlösung.</p>

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<p>ersetzt. Python 3.9 ist daher jetzt erforderlich, um die Skripte und Beispiele zu verwenden, auch wenn Sie das Python-SDK nicht verwenden (oder verwenden möchten).</p> <ul style="list-style-type: none">• Diese Version erweitert die Unterstützung für Amazon Linux 2.• Wir haben mehrere Fehler behoben in SimSpace Weaver Local. <p>Weitere Informationen finden Sie in den Versionshinweisen.</p> <p>Fehlerbehebung</p> <ul style="list-style-type: none">• Wir haben einen Fehler behoben, durch den Entitäten ihr Eigentum verloren,			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	wenn sie ihre Übertragung zwischen Mitarbeitern an entfernten Standorten nicht abgeschlossen haben.			
1.16.0	<p>Neue Funktion:</p> <ul style="list-style-type: none"> Sie können jetzt Nachrichten APIs im SimSpace Weaver App-SDK verwenden, um Nachrichten zwischen Ihren Apps zu senden und zu empfangen. Diese Funktion ist für C++, Python und die Integrationen Unity und Unreal Engine 5 verfügbar. 	12. Februar 2024	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	<ul style="list-style-type: none"> Komplettes Paket Nur Bibliotheken <p>Siehe Problembehandlung.</p>

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.15.3	<p>SimSpace Weaver Local aktualisieren:</p> <ul style="list-style-type: none">Wir haben uns verändert SimSpace Weaver Local um es enger an der Entwicklung für das auszurichten AWS Cloud. Diese Änderungen wirken sich auf C++-, Python-, Unity- und Unreal Engine-Projekte und Workflows für aus SimSpace Weaver Local.	4. Dezember 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.15.2	<p>Update für das verteilbare App SDK-Paket:</p> <ul style="list-style-type: none">Wir haben das aktualisierte <code>Dockerfile</code>, um die spezifische erforderliche Version von <code>cmake</code> zu verwenden. Docker-Container-Builds könnten ohne diese Änderung fehlschlagen.	2. November 2022	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar
1.15.1	<p>Funktionsupdate:</p> <ul style="list-style-type: none">Python-SDK: Diese Version behebt ein Problem, das dazu führte, dass Python-basierte Simulationen in der AWS Cloud fehlschlagen.	22. September 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.15.0	<p>Neue Funktion:</p> <ul style="list-style-type: none">• Python SDK: Sie können Ihre Simulationen jetzt in Python entwickeln. Das verteilbare SimSpace Weaver App-SDK-Paket enthält eine Vorlage für ein Python-Beispielprojekt und den zugehörigen Python-View-Client.	31. August 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.14.0	<p>Neue Funktionen:</p> <ul style="list-style-type: none">• Benutzerdefinierte Container: Erstellen Sie Ihr eigenes auf Amazon Linux 2 (AL2) basierendes Container-Image, speichern Sie es in Amazon Elastic Container Registry (Amazon ECR) und verwenden Sie es, um Ihre SimSpace Weaver Apps in der AWS Cloud auszuführen.• Mehrere räumliche Domänen: Erstellen Sie in einer Simulation mehr als eine räumliche Domäne. Separate Simulationslogik, anstatt alles in einer einzigen	26. Juli 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<p>räumlichen App zu kombinieren. Ordnen Sie räumlichen Domänen je nach Bedarf unterschiedliche Ressourcen zu.</p> <ul style="list-style-type: none">• Unbegrenzte Tick-Rate : Sorgen Sie dafür, dass Ihre Simulationen so schnell ausgeführt werden kann, wie Ihr Code ausgeführt werden kann. Stellen Sie die Uhr Ihrer Simulation so ein, dass sie den nächsten Tick sendet, sobald alle Apps ihre Commit-Operationen für den aktuellen Tick abgeschlossen haben.			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<p>SimSpace Weaver App-SDK:</p> <ul style="list-style-type: none">• Der Wert von <code>tick_rate</code> ist jetzt eine Zeichenfolge. Der Wert muss doppelte Anführungszeichen (") enthalten. Die Tickrate für frühere Versionen ist immer noch eine Ganzzahl.			
1.13.1	<p>SimSpace Weaver App-SDK:</p> <ul style="list-style-type: none">• Funktionsupdate: Die Projekterstellung funktioniert jetzt korrekt mit der <code>PathfindingSampleUnreal</code> Vorlage.	7. Juni 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.13.0	<p>SimSpace Weaver Dienst APIs:</p> <ul style="list-style-type: none"> • Neue CreateSnapshot Aktion • Änderungen an der StartSimulation Aktion: <ul style="list-style-type: none"> • Es wurde ein <code>SnapshotS3Location</code> Parameter hinzugefügt, um mit einem Snapshot zu beginnen. • Der <code>SchemaS3Location</code> Parameter ist jetzt optional. • Änderungen an der DescribeSimulation Ausgabe: 	29. April 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<ul style="list-style-type: none"> • SchemaError ist veraltet. • Ein StartError Feld wurde hinzugefügt. • Ein SnapshotS3Location Feld wurde hinzugefügt. • Ein SNAPSHOT_IN_PROGRESS Simulationstatus wurde hinzugefügt. • Neuer S3Destination Datentyp <p>SimSpace Weaver Konsole:</p> <ul style="list-style-type: none"> • Neue Funktionen zum Erstellen von Schnappschüssen. 			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<ul style="list-style-type: none"> • Neue Funktionalität zum Starten einer Simulation aus einem Snapshot. <p>SimSpace Weaver App-SDK:</p> <ul style="list-style-type: none"> • Neue Skripte zur Unterstützung von Snapshots <ul style="list-style-type: none"> • create-snapshot- <i>project-name</i> .bat • start-from-snapshot- <i>project-name</i> .bat • quick-start-from-snapshot- <i>project-name</i> -cli.bat • list-snapshots- <i>project-name</i> .bat • Projekte verwenden jetzt einen einzigen Amazon S3 S3-Bucket pro Projekt: weaver- 			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
	<pre>lowercase -project- name -- account-n umber region</pre>			
1.12.3	<p>SimSpace Weaver Anwendungs-SDK:</p> <ul style="list-style-type: none"> Die folgenden Skripte unterstützen jetzt den <code>--maximum</code> <code>-duration</code> Parameter: <ul style="list-style-type: none"> <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> 	27. März 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.12.2	<p>SimSpace Weaver Anwendungs-SDK:</p> <ul style="list-style-type: none">• Bugfix: läuft <code>docker-create-image.bat</code> jetzt korrekt.	1. März 2023	Sehen Sie sich den AWS SimSpace Weaver Guide-Katalog an.	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.12.1	<p>SimSpace Weaver App-SDK:</p> <ul style="list-style-type: none"> • Die Skripts akzeptieren jetzt ein AWS CLI Profil, das für die AWS Authentifizierung verwendet werden kann. • Die Skripts unterstützen jetzt AWS IAM Identity Center die AWS Authentifizierung. <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none"> • Bugfix: Gibt <code>Api::BeginUpdateWillBlock</code> jetzt korrekt zurück, <code>true</code> wenn nicht alle Geo-Apps an der Simulation teilgenommen haben. 	28. Februar 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App SDK herunterladen
1.12.0	Veröffentlichung zur allgemeinen Verfügbarkeit (GA)	29. November 2022	Weitere Informationen finden Sie im AWS SimSpace Weaver Guide-Katalog .	Nicht zum Herunterladen verfügbar

AWS SimSpace Weaver Ausführung 1.17.0

Diese Version ist eine Überarbeitung des verteilbaren SimSpace Weaver App-SDK-Pakets. Wir haben veraltete Windows-Batch- und Linux-Bash-Skripte durch Python-basierte Skripte ersetzt.

Important

Python 3.9 ist jetzt eine Voraussetzung für die Verwendung der Skripte und Beispiele, nicht nur für das Python-SDK.

Inhalt

- [Wesentliche Änderungen für 1.17.0](#)
- [Ein Projekt auf Version 1.17.0 aktualisieren](#)
- [Häufig gestellte Fragen zu Version 1.17.0](#)

Wesentliche Änderungen für 1.17.0

- Vereinfachte Projekterstellung
 - Nach der Ausführung können Sie Ihr eigenes Projekt `erstellensetup.py`, indem Sie einfach ein Beispiel kopieren und einfügen.
- Beispiele mit einem Klick
 - Die ZIP-Datei für die Distribution enthält jetzt ready-to-use Beispiele, die auch nach dem Einrichten der Distribution funktionieren.
- Jedes SDK ist jetzt in einem eigenen Verzeichnis vorhanden: `cpppython`, `unreal`, und `unity`. Je nachdem, welches SDK Sie verwenden, müssen Sie möglicherweise Ihre Pfade aktualisieren.

- Verbesserungen der Hilfsskripte.
 - Skripte enthalten jetzt mehrere AWS CLI Optionen, um ihre Flexibilität zu maximieren.
 - Integrierter Konsolen-Client-Start und -Verbindung als Teil des Schnellstarts.
 - Verbesserte Konsolenausgabe.
 - Das Erstellen von Unreal- und Unity-Beispielen funktioniert jetzt mit `quick-start`, ohne dass weitere manuelle Schritte erforderlich sind.
 - SimSpace Weaver Local funktioniert jetzt durch einfaches Aufrufen `quick-start`, kein manuelles Erstellen und Starten mehr.
 - SimSpace Weaver Local `quick-start` hat integrierte Unterstützung für die Protokollierung der App-Ausgabe.
 - SimSpace Weaver Local kann jetzt in einer Umgebung ohne GUI gestartet werden, z. B. in einer SSH-Sitzung.
 - Die Funktion „Benutzerdefinierter Container“ ist jetzt in das `quick-start` Skript integriert.
 - Verbesserte Unterstützung für Amazon Linux 2 (AL2): Skript-Workflows für Windows und AL2 sind jetzt vergleichbar. Bisher erforderten AL2 Projekte mehr manuelle Schritte und SimSpace Weaver Local wurde nicht unterstützt für AL2.
- Die Unreal Engine- und Unity-Plugins sind jetzt als Teil des verteilbaren SimSpace Weaver App-SDK-Pakets enthalten.
- Fehlerkorrekturen für SimSpace Weaver Local
 - Es wurde ein Fehler behoben, bei dem Entitäten dieselbe Entitäts-ID zugewiesen werden konnte.
 - Es wurde ein Fehler behoben, bei dem zwei Partitionen dieselbe Partitions-ID zugewiesen werden konnte.
 - Ein Fehler im Zusammenhang mit dem Versuch von Apps, in Entitäten zu schreiben, die ihnen nicht gehörten, wurde behoben.
 - Ein Problem mit einem Speicherleck wurde behoben.

Ein Projekt auf Version 1.17.0 aktualisieren

1. Richten Sie die 1.17.0-Distribution ein: Gehen Sie die Einrichtungsprozeduren erneut durch, da wir sie für 1.17.0 geändert haben. Weitere Informationen finden Sie unter [Einrichtung für SimSpace Weaver](#).
2. Jedes Weaver App SDK ist jetzt in einem eigenen Verzeichnis vorhanden. Aktualisieren Sie Ihre `Build-Pfade`, um dies widerzuspiegeln.

- a. C++-Verzeichnis: `SimSpaceWeaverAppSdk/cpp`
 - Das SimSpace Weaver C++ App SDK verwendet jetzt eine `FindSimSpaceWeaverAppSdk.cmake` Datei. Diese Datei richtet ein `weaver` Ziel ein, mit dem verknüpft wird, und enthält wichtige Bugfixes beim Erstellen für Weaver in der AWS Cloud. Sie sollten dies verwenden, anstatt direkt auf die Binärdateien zu verlinken.
 - b. Python-Verzeichnis: `SimSpaceWeaverAppSdk/python`
 - c. Unity-Plugin: `SimSpaceWeaverAppSdk/unity`
 - d. Unreal Engine-Plugin: `SimSpaceWeaverAppSdk/unreal`
3. Die vorherigen `tools` Skripte funktionieren mit der neuen SimSpace Weaver Distribution nicht. Um die neuen `tools` Skripte mit Ihrem Projekt zu verwenden:
- a. Löschen Sie Ihre alten `tools/local` Verzeichnisse `tools/windowstools/linux`, und.
 - b. Kopieren Sie das `tools` Verzeichnis eines Beispielprojekts, das dasselbe SimSpace Weaver App-SDK wie Ihr Projekt verwendet. Stellen Sie sicher, dass Sie ausgeführt haben, `setup.py` bevor Sie dieses Verzeichnis kopieren.

Important

Es wird nur garantiert, dass die Tools-Skripte mit den Beispielprojekten funktionieren. Möglicherweise müssen Sie diese Skripte, insbesondere das `build.py` Skript, bearbeiten, um mit Ihrem Projekt arbeiten zu können. Alle Änderungen betreffen nur Ihr Projekt, daher können wir Ihnen keine Hinweise geben.

Häufig gestellte Fragen zu Version 1.17.0

Muss ich auf Version 1.17.0 aktualisieren?

Dies ist kein erforderliches Update, da es keine Änderungen an der SimSpace Weaver API oder dem SimSpace Weaver App-SDK gibt. Sie müssen auf 1.17.0 aktualisieren, wenn Sie 1.17.0 verwenden möchten SimSpace Weaver Local, das mehrere Bugfixes enthält.

Was ist die erforderliche Python-Mindestversion?

Python 3.9 ist die Mindestversion.

Welche CMake Mindestversion ist erforderlich?

CMake Version 3.13 ist die Mindestversion.

Welche Mindestversion von Unreal Engine ist erforderlich?

Unreal Engine 5.0 ist das Minimum.

Welche Mindestversion von Unity ist erforderlich?

Unity-Version 2022.3.19.F1 ist die Mindestversion.

AWS SimSpace Weaver Ausführung 1.15.1

Diese Version ist ein erforderliches Update für das Python-SDK, das ursprünglich in SimSpace Weaver Version 1.15.0 veröffentlicht wurde. Es behebt ein Versionskonflikt, das dazu führte, dass Python-basierte Simulationen in der fehlschlugen. AWS Cloud Verwenden Sie diese Version anstelle von 1.15.0.

Aktualisieren Sie ein vorhandenes Python-Projekt auf 1.15.1

Wenn Sie über ein vorhandenes Python-Projekt verfügen, das Sie mit dem Python-SDK der Version 1.15.0 erstellt haben, müssen Sie die folgenden Schritte ausführen, um es auf 1.15.1 zu aktualisieren, damit es in der ausgeführt werden kann. AWS Cloud

Anstatt dieses Verfahren zu befolgen, können Sie auch ein neues Python-Projekt mit dem Python-SDK 1.15.1 erstellen und Ihren benutzerdefinierten Code in das neue Projekt verschieben.

Um ein Python-Projekt 1.15.0 auf 1.15.1 zu aktualisieren

1. Gehe zum Ordner deines Python-Projekts.
2. `src/PythonBubblesSample/bin/run-python`Ändern Sie die folgende Zeile:

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

In Folgendes:

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. `CMakeLists.txt`Löschen Sie die folgenden Zeilen:

- ```
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)
```

Fehlerbehebung für Version 1.15.1

Nach dem Aktualisieren einer 1.15.0 Python-Simulation kann sie nicht gestartet werden in AWS Cloud

Symptome: Etwa 5-10 Minuten nach dem Start der Simulation meldet das Simulationsverwaltungsprotokoll einen `internal error` und der Simulationsstatus lautet. FAILED

Dies kann passieren, wenn eine Bibliotheksdatei aus dem Python-SDK 1.15.0 in einer App-ZIP-Datei enthalten ist. Stellen Sie sicher, dass Sie die Schritte zur Aktualisierung Ihres Projekts abgeschlossen haben und dass es `libweaver_app_sdk_python_v1.so` nicht in Ihren ZIP-Dateien enthalten ist oder in irgendeiner Weise referenziert wird.

Häufig gestellte Fragen zu Version 1.15.1

Wirkt sich diese Version auf etwas anderes als das Python-SDK aus?

Nein.

Muss ich auf Version 1.15.1 aktualisieren?

Sie müssen nicht auf 1.15.1 aktualisieren, wenn Sie nicht beabsichtigen, Python für Ihre Geo-Apps zu verwenden. Wenn Sie auf 1.15.0 aktualisiert haben, laufen Ihre Python-basierten Simulationen nicht in der. AWS Cloud Wir empfehlen Ihnen, auf 1.15.1 zu aktualisieren, wenn Sie 1.15.0 verwenden.

Was ist `$LD_LIBRARY_PATH`?

Es ist der Speicherort des Python-SDK, wenn Ihre Simulation im ausgeführt wird AWS Cloud. Das ist neu für 1.15.1. Wir haben diese Änderung vorgenommen, um future Probleme mit der Python-Version zu vermeiden. Das Verknüpfen mit diesem Verzeichnis entspricht funktionell dem Verknüpfen mit `libweaver_app_sdk_python_v1.so` 1.15.0.

Dokumentenverlauf für AWS SimSpace Weaver

In der folgenden Tabelle werden wichtige Änderungen an der SimSpace Weaver Dokumentation beschrieben.

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
20. Mai 2025	Ende der Support-Mitteilung	Hinweis zum Ende des Supports: Am 20. Mai 2026 AWS endet der Support für AWS SimSpace Weaver. Nach dem 20. Mai 2026 können Sie nicht mehr auf die SimSpace Weaver Konsole oder SimSpace Weaver die Ressourcen zugreifen. Weitere Informationen finden Sie unter AWS SimSpace Weaver Ende des Supports .	N/A
17. April 2024	Aktualisierter Inhalt	Das gesamte Benutzerhandbuch für die Version 1.17.0 wurde aktualisiert. Wesentliche Änderungen am Einrichtung Kapitel und Erste Schritte an den Tutorials. Weitere Informationen finden Sie in den Versionshinweisen .	N/A
12. Februar 2024	Aktualisierter Inhalt	Das AWS SimSpace Weaver Versionen Kapitel	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
		für die Version 1.16.0 wurde aktualisiert.	
12. Februar 2	Neuer Inhalt	Der Messaging Abschnitt wurde als Teil der Version 1.16.0 hinzugefügt. In diesem Abschnitt werden die Nachrichten beschrieben, die dem SimSpace Weaver App-SDK APIs hinzugefügt wurden. Sie können diese verwenden APIs , um Nachrichten zwischen Ihren Apps zu senden und zu empfangen.	N/A
12. Februar 2	Aktualisierter Inhalt	Das SimSpace Weaver Referenz zum Simulationschema Kapitel für Version 1.16.0 wurde aktualisiert.	N/A
12. Februar 2	Aktualisierter Inhalt	Dem Kapitel wurden Dienstkontingente für Nachrichten hinzugefügt. SimSpace Weaver-Endpunkte und Kontingente	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
12. Februar 2023	Neue Anleitungen	Teilen Sie den Inhalt für Versionen vor 1.16.0 in ein separates Handbuch auf. Der AWS SimSpace Weaver Guide-Katalog (auf der Landingpage der Hauptdokumentation verfügbar) wurde hinzugefügt, um auf Anleitungen für frühere Versionen zuzugreifen.	N/A
4. Dezember 2023	Aktualisierter Inhalt	Das AWS SimSpace Weaver Versionen Kapitel für die Version 1.15.3 wurde aktualisiert.	N/A
4. Dezember 2023	Aktualisierter Inhalt	Das AWS SimSpace Weaver Versionen Kapitel wurde aktualisiert und enthält nun Installationsanweisungen für die neueste Version.	N/A
4. Dezember 2023	Aktualisierter Inhalt	Das Servicekontingente für SimSpace Weaver Local wurde aktualisiert.	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
4. Dezember 2023	Neuer und aktualisierter Inhalt	Der Lokale Entwicklung in SimSpace Weaver Abschnitt wurde neu strukturiert und eine neue Seite hinzugefügt, auf der die in Version 1.15.3 SimSpace Weaver Local eingeführten Unterschiede beschrieben werden.	N/A
7. November	Aktualisierter Inhalt	Die Anweisungen zur Einrichtung von Docker und WSL zur Verwendung des direkten Download-Links/der URL für das App-SDK wurden aktualisiert. Weitere Informationen finden Sie unter Richten Sie Ihre lokale Umgebung ein für SimSpace Weaver .	N/A
2. November	Aktualisierter Inhalt	Die Seite mit den Dienstversionen für die Version 1.15.2 wurde aktualisiert. Weitere Informationen finden Sie unter Versionen der Dienste .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
23. Oktober 2	Aktualisierter Inhalt	Die Seite mit den Serviceversionen wurde mit neuen Anweisungen zum Herunterladen des verteilbaren App-SDK-Pakets aktualisiert. Kunden sollten jetzt nur noch einen unserer genehmigten direkten Download-Links verwenden und den nicht AWS CLI zum Herunterladen des verteilbaren App-SDK-Pakets verwenden. Weitere Informationen finden Sie unter Laden Sie die neueste Version herunter .	N/A
22. September	Neuer Inhalt	Es wurde eine Seite mit Versionshinweisen mit Aktualisierungsanweisungen für die Version 1.15.1 hinzugefügt. Weitere Informationen finden Sie unter AWS SimSpace Weaver Ausführung 1.15.1 .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
10. September 2023	Neuer Inhalt	Es wurde ein Abschnitt zur Fehlerbehebung für Situationen hinzugefügt, in denen der AWS CLI nicht erkannt wird SimSpace Weaver. Weitere Informationen finden Sie unter Das AWS CLI erkennt nicht simspaceweaver .	N/A
10. September 2023	Aktualisierter Inhalt	Aktualisierte Installationsanweisungen für die AWS CLI in WSL. Weitere Informationen finden Sie unter Richten Sie das SimSpace Weaver Distributionspaket ein für Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL) .	N/A
07. September	API-Aktualisierung	BucketName und ObjectKey sind jetzt für den S3Location-Datentyp erforderlich. BucketName ist jetzt für den S3Destination-Datentyp erforderlich.	AWS SDK: 2023-09-07
31. August 2023	Neuer Inhalt	Ein neuer Abschnitt für Version 1.15.0 wurde hinzugefügt: Verwenden von Python	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
15. August 2023	Aktualisierter Inhalt	Die Download-Anweisungen wurden aktualisiert AWS SimSpace Weaver Versionen , sodass nur offizielle SimSpace Weaver Amazon S3 S3-Buckets aufgeführt sind. Andere Download-Speicherorte werden nicht von böartigem Code kontrolliert AWS und können böartigen Code enthalten.	N/A
26. Juli 2023	Aktualisierter Inhalt	Aktualisiert Uhr .	N/A
26. Juli 2023	Aktualisierter Inhalt	Aktualisiert Konfiguration von räumlichen Domänen .	N/A
26. Juli 2023	Neuer Inhalt	Neuer Abschnitt: Maßgeschneiderte Behälter .	N/A
26. Juli 2023	Aktualisierter Inhalt	AWS SimSpace Weaver Versionen Für Version 1.14.0 aktualisiert.	N/A
6. Juli 2023	Neuer Inhalt	Neuer Abschnitt: PathfindingSample Der Konsolenclient kann keine Verbindung herstellen .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
7. Juni 2023	Aktualisierter Inhalt	AWS SimSpace Weaver Versionen Für Version 1.13.1 aktualisiert.	N/A
15. Mai 2023	Neuer Inhalt	Neuer Abschnitt : Verwenden von Schnappschüssen mit AWS CloudFormation.	N/A
29. April 2023	Neuer Inhalt	Inhalt für Version 1.13.0 hinzugefügt. Weitere Informationen finden Sie unter AWS SimSpace Weaver Versionen.	AWS SDK: 2023-04-28
27. März 2023	Neuer Inhalt	Es wurde ein Abschnitt hinzugefügt, der die maximale Dauer von Simulationen beschreibt. In den Tutorials für Version 1.12.3 wurden Hinweise hinzugefügt, mit denen SimSpace Weaver App-SDK-Skripts um Unterstützung für den <code>--maximum-duration</code> Parameter erweitert wurden.	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
9. März 2023	Geänderte Inhalte	Es wurde klargestellt, dass wir nur Anweisungen für Docker on Windows und for Windows Subsystem for Linux (WSL) bereitstellen und dass WSL (und jede andere Linux-Umgebung) nicht unterstützt wird.	N/A
28. Februar 2023	Neuer Inhalt	Es wurde ein Kapitel hinzugefügt, das Versionen beschreibt SimSpace Weaver .	N/A
28. Februar 2023	Geänderte Inhalte	Der Inhalt zur Authentifizierung wurde geändert und umfasst nun auch die Verwendung von AWS IAM Identity Center und benannten Profilen für die AWS Command Line Interface (AWS CLI).	N/A
17. Februar 2023	Neuer Inhalt	Es wurde ein Abschnitt über die Verwaltung Ihrer Ressourcen mit hinzugefügt AWS CloudFormation.	N/A
23. Januar 2023	Neuer Inhalt	Es wurden Anweisungen zum Debuggen lokaler Simulationen hinzugefügt.	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen wurden aktualisiert
29. November	Servicestart	Das Benutzerhandbuch und die API-Referenz für SimSpace Weaver wurden veröffentlicht.	AWS SDK: 29.11.2022

Glossar

Dieses Glossar definiert Begriffe, die spezifisch für sind. AWS SimSpace Weaver

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS Allgemeinen Referenz.

A

App	Ausführbarer Code (auch Binärdateien genannt), den Sie erstellen. Der Begriff App kann sich auf den Code oder eine laufende Instanz dieses Codes beziehen. Eine App kapselt das Simulationsverhalten ein. Apps erstellen, löschen, lesen und aktualisieren Entitäten .
App-SDK	Ein Software Development Kit (SDK), mit dem Sie eine App integrieren SimSpace Weaver. Das SDK ermöglicht APIs das Lesen und Schreiben von Entitätsdaten und die Nachverfolgung der Simulationszeit. Weitere Informationen finden Sie unter SimSpace Weaver App-SDK .

C

Kunde	Prozesse (oder deren Definitionen), die außerhalb der Simulation existieren SimSpace Weaver und über eine benutzerdefinierte App oder Service-App mit der Simulation interagieren. Sie können einen Client verwenden, um den Simulationsstatus anzuzeigen oder zu ändern.
Uhr	Eine Abstraktion der SimSpace Weaver internen Planungsprozesse. Die Uhr veröffentlicht Ticks in Apps , um die Zeitsynchronisierung aufrechtzuerhalten. Jede Simulation hat ihre eigene Uhr.
Taktrate	Die Anzahl der Ticks pro Sekunde, die die Uhr in Apps veröffentlicht. Weitere Informationen zu den unterstützten Taktraten finden Sie unter SimSpace Weaver-Endpunkte und Kontingente .
Takttickrate	Siehe Taktrate .
Ressourceneinheit berechnen	Eine Einheit von Rechenressourcen (Prozessor und Speicher) auf einem Worker . Einer einzelnen Instanz einer App wird normalerweise 1

Rechenressourceneinheit zugewiesen. Sie können jeder App mehr als eine Rechenressourceneinheit zuweisen.

benutzerdefinierte App Eine Art von [App](#), die Sie verwenden, um den Status der Simulation zu lesen und mit ihm zu interagieren. Benutzerdefinierte Apps können Entitäten in der Simulation erstellen, besitzen diese aber nicht. Wenn eine benutzerdefinierte App eine Entität erstellt, muss sie die Entität in die [räumliche Domäne](#) übertragen. Sie steuern den Lebenszyklus einer benutzerdefinierten App mithilfe der App APIs. Weitere Informationen zu der finden SimSpace Weaver APIs Sie unter [SimSpace Weaver API-Referenzen](#).

benutzerdefinierte Domäne Eine [Domain](#), die [benutzerdefinierte Apps](#) enthält.

benutzerdefinierte Partition Die [Partition](#) einer [benutzerdefinierten App](#).

D

Frist Eine [tatsächliche Uhrzeit](#), bis zu der ein Vorgang (z. B. die Bearbeitung eines [Häkchens](#)) abgeschlossen sein sollte.

Domäne Eine Gruppe von [App-Instanzen](#), die denselben ausführbaren Code (App-Binärdatei) ausführen und dieselben Startoptionen haben.

E

Endpunkt (Dienst) Ein vollqualifizierter Domänenname (FQDN), den Programme (wie der AWS Command Line Interface) verwenden, um eine Verbindung mit dem SimSpace Weaver Dienst herzustellen.

Endpunkt (Simulation) Eine IP-Adresse und Portnummer, über die Clients eine Verbindung zu einer Simulation herstellen. Sie können Endpunkte in [benutzerdefinierten Apps und Service-Apps](#) konfigurieren.

Entität Kundendatenobjekte (oder deren Definitionen). Entitäten können statisch (bleiben an einem Ort) oder dynamisch (bewegen sich durch den Simulationsraum) sein. Zum Beispiel Personen und Gebäude in einer Simulation.

I

Index (Simulation) Eine Beschreibung der räumlichen Eigenschaften einer Simulation, einschließlich ihrer räumlichen Grenzen und ihres Koordinatensystems.

L

Lebenszyklus (einer App) Eine Beschreibung der erwarteten logischen Schritte, die eine [App](#) während einer Simulation durchläuft. Lebenszyklen werden entweder verwaltet (SimSpace Weaver startet und stoppt die App) oder nicht verwaltet (Sie starten und beenden die App).

laden (Entitätsfelddaten) Liest [Entitätsfelddaten](#) aus dem [State Fabric](#).

P

Partition Ein Segment des gemeinsam genutzten Speichers auf einem [Worker](#). Jede Partition enthält eine separate Teilmenge von [Entitäten](#) innerhalb einer [Domäne](#). Jeder [App](#) ist eine Partition zugewiesen. Eine App besitzt alle Entitäten in ihrer Partition. Wenn eine App eine Entität erstellt, erstellt sie sie in ihrer Partition. Wenn Entitäten von einer Partition auf eine andere Partition verschoben werden, wird der Besitz von der App der Quellpartition auf die App der Zielpartition übertragen.

R

Ressourceneinheit Siehe [???](#).

S

Schema Ein YAML- oder JSON-Dokument, das die Konfiguration einer Simulation beschreibt. SimSpace Weaver verwendet ein Schema, um eine [Simulationsressource](#) zu erstellen.

Service-App Eine Art von [App](#), die Sie verwenden, um den Status der Simulation zu lesen und mit ihm zu interagieren. Service-Apps können Entitäten in der Simulation erstellen, müssen sie jedoch in die räumliche [Domäne](#)

übertragen. SimSpace Weaver verwaltet den [Lebenszyklus](#) einer Service-App und startet eine (oder mehrere, wie in Ihrem [Simulationsschema](#) angegeben) für jeden [Worker](#) in Ihrer Simulation.

Dienstdomäne	Eine Domain , die Service-Apps enthält.
Servicepartition	Die Partition einer Service-App .
Simulation (Ressource)	Eine Abstraktion eines Rechenclusters, auf dem ein simulierter virtueller Raum ausgeführt wird. Sie können mehrere Simulationen durchführen. Sie konfigurieren eine Simulation mithilfe eines Schemas .
räumliche App	Eine Art von App , die die Kern-Simulationslogik kapselt. Jede Geo-App besitzt 1 (und nur 1) Partition.
räumliche Domäne	Eine Domäne , die räumliche Apps enthält.
räumliche Partition	Die Partition einer räumlichen App .
State Fabric	SimSpace Weaver ist eine In-Memory-Datenbank. Die State Fabric speichert den Status der Simulationen, einschließlich Entitäten und interner SimSpace Weaver Daten.
speichern (Entitätsfelddaten)	Schreiben Sie Entitätsfelddaten in die State Fabric .
Abonnement	Eine lang andauernde Anfrage für eine bestimmte App-Instanz , Daten aus einem Abonnementbereich zu empfangen. Die abonnierende App verwendet ein Abonnement, um Änderungen an Entitäten innerhalb des Abonnementbereichs zu ermitteln.
Abonnementbereich	Eine zweidimensionale Region des Simulationsraums. Ein Abonnement bezieht sich auf einen Abonnementbereich. Ein Abonnementbereich kann sich über mehr als eine Partition erstrecken und auch Teile von Partitionen enthalten. Ein Abonnementbereich ist innerhalb seiner definierten Grenzen kontinuierlich.

T

ankreuzen	Ein diskreter Wert für die Zeit (entweder Wanduhr oder Simulationszeit). Apps können schneller iterieren als die Dauer der Ticks, es wird jedoch
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------

erwartet, dass sie bestimmte Ticks innerhalb bestimmter Fristen schreiben. Alle Operationen für alle Apps für einen bestimmten Tick müssen abgeschlossen sein, bevor der nächste Tick beginnen kann.

Tick-Rate

Siehe Taktrate.

Zeit (tatsächlich)

Die aktuelle Zeit aus der Perspektive der Realität. SimSpace Weaver verwendet einen 64-Bit-POSIX-Zeitstempel, der der Anzahl der Nanosekunden seit der Epoche entspricht Unix (January 1, 1970, 00:00:00 UTC).

Zeit (Simulation)

Die aktuelle Uhrzeit aus Sicht der Simulation. SimSpace Weaver verwendet einen logischen 64-Bit-Integer-Tick-Zähler, der möglicherweise nicht direkt der tatsächlichen Zeit entspricht.

W

Arbeiter

Eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance, die Simulationscode ausführt.