



Benutzerhandbuch für Version 1.16.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Benutzerhandbuch für Version 1.16.0

Copyright © 2024 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

Was ist SimSpace Weaver?	1
Die wichtigsten Konzepte	1
Funktionsweise von SimSpace Weaver	2
Wie benutzt du SimSpace Weaver	5
Simulationsschema	6
Arbeiter und Ressourceneinheiten	6
Simulationsuhr	7
Partitionen	7
Bundesstaat Fabric	7
Entitäten	8
Apps	8
Beispielanwendungsfälle	11
Einrichtung	13
Richten Sie Ihr Konto ein	13
So melden Sie sich für ein AWS-Konto an	13
Einen Administratorbenutzer erstellen	14
Fügen Sie zu verwendende Berechtigungen hinzu SimSpace Weaver	15
Einrichten der lokalen Umgebung	16
AL2 ein Docker	17
AL2 ein WSL	18
Verwendung lizenzierter Software	22
Erste Schritte	23
Schnellstart:	23
Schritt 1: Erstellen eines Projekts	24
Schritt 2: Protokollierung einschalten	27
Schritt 3: Ausführen des Schnellstartskripts	29
Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer	32
Schritt 5: Sehen Sie sich Ihre Simulation an	40
Schritt 6: Anhalten und Bereinigen Ihrer Simulation	47
Ausführliches Tutorial	54
Schritt 1: Erstellen eines Projekts	54
Schritt 2: Protokollierung einschalten	58
Schritt 3: Hochladen Ihres Simulationsschemas	60
Schritt 4: Erstellen Erstellen: Erstellen: Erstellen	62

Schritt 5: Hochladen von Apps	64
Schritt 6: Starten Ihrer Simulation	67
Schritt 7: Simulationsdetails abrufen	71
Schritt 8: Starten benutzerdefinierter Apps	77
Schritt 9: Starten der Uhr	82
Schritt 10: Überprüfen der Protokolle	84
Schritt 11: Sehen Sie sich Ihre Simulation an	87
Schritt 12: Anhalten und Bereinigen Ihrer Simulation	95
Arbeiten mit SimSpace Weaver	102
Konfigurieren Ihrer Simulation	102
Simulationskonfigurationsparameter	104
SDK-Version	105
Eigenschaften der Simulation	105
Workers	106
Uhr	107
Partitionierungsstrategien	110
Domains	111
Maximale Dauer	121
Maximaler Wert	122
Standardwert	122
Mindestwert	122
Starten einer Simulation mithilfe von SimSpace Weaver App-SDK-Skripten	122
Starten einer Simulation mit der Konsole	123
Der Status einer Simulation, die ihre maximale Dauer erreicht	123
Entwicklung Apps	123
Spatial Apps	124
Benutzerdefinierte Apps	125
Entwicklung von Client-Anwendungen	126
Lokale Entwicklung	127
Entwicklung	128
Ausführen	129
Anzeigen	131
Stoppen	131
Debuggen	132
Unterschiede in 1.15.3	133
SimSpace Weaver -App-SDK	139

API-Methoden geben einen zurück Result	140
Interaktion mit dem App-SDK auf oberster Ebene	140
Simulationsmanagement	141
Subscriptions (Abonnements)	144
Entitäten	145
Entitätsereignisse	157
Result und Fehlerbehandlung	164
Generika und Domänentypen	166
Verschiedene Anwendungs-SDK-Operationen	166
SimSpace WeaverDemo-Framework	169
Arbeiten mit Kontingenten	170
Holen Sie sich die Limits für eine App	171
Ermitteln Sie die Menge der von einer App verwendeten Ressourcen	171
Zurücksetzen von Kennzahlen	172
Überschreiten eines Limits	173
Es wird kein Speicher mehr	173
Bewährte Methoden	173
Debuggen von Simulationen	174
Verwenden SimSpace Weaver Local und Ansehen der Konsolenausgabe	174
Sehen Sie sich Ihre Protokolle in Amazon CloudWatch Logs an	175
APIDescribe-Aufrufe verwenden	175
Verbinden eines Clients	176
Debuggen lokaler Simulationen	176
Benutzerdefinierte Container	177
Erstellen eines benutzerdefinierten Containers	178
Ändern eines Projekts zur Verwendung eines benutzerdefinierten Containers	179
Häufig gestellte Fragen	182
Fehlerbehebung	183
Verwenden von Python	183
Ein Python-Projekt erstellen	184
Eine Python-Simulation starten	187
Der Python-Beispielclient	188
Schreiben Sie Ihre eigenen Build-Skripte	189
Häufig gestellte Fragen	189
Fehlerbehebung	189
Unterstützung für andere Engines	191

Unity	192
Unreal Engine	193
Verwendung lizenzierter Software	193
Verwaltung von Ressourcen mit AWS CloudFormation	194
Snapshots	196
Snapshots	197
Anwendungsfälle für Snapshots	198
SimSpace WeaverApp-SDK	199
SimSpace Weaver-Konsole	204
AWS CLI	206
SimSpace Weaver-APIs	209
Häufig gestellte Fragen	209
Messaging	210
Anwendungsfälle für Messaging	211
Verwenden der Messaging-APIs	211
Wann sollte Messaging verwendet werden?	219
Tipps bei der Arbeit mit Messaging	222
Messaging-Fehler und Fehlerbehebung	224
Bewährte Methoden	227
Einrichten von Abrechnungsalarmen	227
Verwenden von SimSpace Weaver Local	227
Simulationen stoppen, die Sie nicht benötigen	228
Löschen von Ressourcen, die Sie nicht benötigen	228
Backups haben	228
Sicherheit	230
Datenschutz	231
Verschlüsselung im Ruhezustand	232
Verschlüsselung während der Übertragung	232
Datenschutz für den Datenverkehr zwischen Netzwerken	233
Identitäts- und Zugriffsverwaltung	233
Zielgruppe	233
Authentifizierung mit Identitäten	234
Verwalten des Zugriffs mit Richtlinien	238
Featuresweise von AWS SimSpace Weaver mit IAM	241
Beispiele für identitätsbasierte Richtlinien	249
Berechtigungen, die für Sie SimSpace Weaver erstellt	253

Serviceübergreifende Confused-Deputy-Prävention	255
Fehlerbehebung	258
Protokollierung und Überwachung von Sicherheitsereignissen	261
Compliance-Validierung	262
Ausfallsicherheit	263
Sicherheit der Infrastruktur	264
Sicherheitsmodell für Netzwerkkonnektivität	264
Konfigurations- und Schwachstellenanalyse	265
Bewährte Methoden für die Gewährleistung der Sicherheit	266
Verschlüsseln der Kommunikation zwischen Ihren Apps und ihren Clients	266
Sichern Sie Ihren Simulationsstatus regelmäßig	266
Verwalten Ihrer Apps und SDKs	266
Protokollierung und Überwachung	268
Anmeldungen CloudWatch	269
Zugreifen auf Ihre SimSpace Weaver Protokolle	269
SimSpace Weaver-Protokolle	269
Überwachung mit CloudWatch	272
SimSpace Weaver -Metriken auf Kontoebene	273
CloudTrail -Protokolle	273
SimSpace Weaver -Informationen in CloudTrail	274
Grundlagen zu SimSpace Weaver-Protokolldateieinträgen	275
Endpunkte und Service Quotas	277
Service-Endpunkte	277
Service Quotas	278
Messaging-Kontingente	281
Taktraten	282
Service-Kontingente für SimSpace Weaver Local	282
Fehlerbehebung	284
AssumeRoleAccessDenied	284
InvalidBucketName	286
ServiceQuotaExceededException	287
TooManyBuckets	287
Die Erlaubnis wurde während des Simulationsstarts verweigert	288
Probleme im Zusammenhang mit der Zeit bei der Verwendung Docker	289
Der Konsolentclient kann keine Verbindung herstellen	289
Nein <code>simspaceweaver</code> in der AWS CLI	291

Schemareferenz	293
Beispiele für ein vollständiges Schema	293
Schemaformat	295
SDK-Version	296
Simulationseigenschaften	296
Worker	298
Uhr	299
Partitionierungsstrategien	299
Domains	301
Platzierungseinschränkungen	312
API-Referenzen	313
SimSpace Weaver-Versionen	314
Aktuelle Version	314
So finden Sie Ihre aktuelle Version	314
Herunterladen der neuesten Version	314
Fehlerbehebung bei Downloads des App-SDK	315
Installieren der neuesten Version	316
Serviceversionen	316
1.15.1	332
Aktualisieren Sie ein vorhandenes Python-Projekt auf 1.15.1	332
Fehlerbehebung für Version 1.15.1	333
Häufig gestellte Fragen zu Version 1.15.1	333
Dokumentverlauf	335
Glossar	344
.....	cccxlx

Was ist AWS SimSpace Weaver?

AWS SimSpace Weaver ist ein Service, den Sie zum Erstellen und Ausführen von großen räumlichen Simulationen in der verwenden können AWS Cloud. Sie können beispielsweise Crowd-Simulationen, 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 Management der Simulationsdaten und die Netzwerkkommunikation zwischen den Amazon EC2 EC2-Instances, auf denen Ihre Simulation ausgeführt wird.

Die wichtigsten Konzepte für SimSpace Weaver

Eine Simulation oder ein Spiel wird durch den Computer begrenzt, auf dem sie ausgeführt wird. Wenn Sie die Größe und Komplexität Ihrer virtuellen Welt erhöhen, beginnt sich die Verarbeitungsleistung zu verschlechtern. Berechnungen dauern länger, den Systemen geht der Arbeitsspeicher aus und die Client-Frameraten sinken. Für Simulationen, die keine Echtzeitleistung benötigen, kann dies nur nervig sein. Oder es könnte sich um eine geschäftskritische Situation handeln, in der erhöhte Verarbeitungsverzögerungen zu erhöhten Kosten führen. Wenn Ihre Simulation oder Ihr Spiel Leistung in Echtzeit benötigt, ist eine Leistungsminderung 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 oft Skalierungsprobleme, indem sie Kopien ihrer virtuellen Welt auf verschiedenen Servern erstellen und die Benutzer auf diese verteilen.

SimSpace Weaver löst das Skalierungsproblem, indem Ihre virtuelle Welt räumlich aufgeteilt und die Teile auf einen Cluster von Recheninstanzen verteilt werden, die in der ausgeführt AWS Cloud werden. Die Recheninstanzen arbeiten zusammen, um die gesamte Simulationswelt parallel zu verarbeiten. Ihre Simulationswelt erscheint als ein einziger integrierter Raum für alles darin 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

- [Funktionsweise von SimSpace Weaver](#)

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

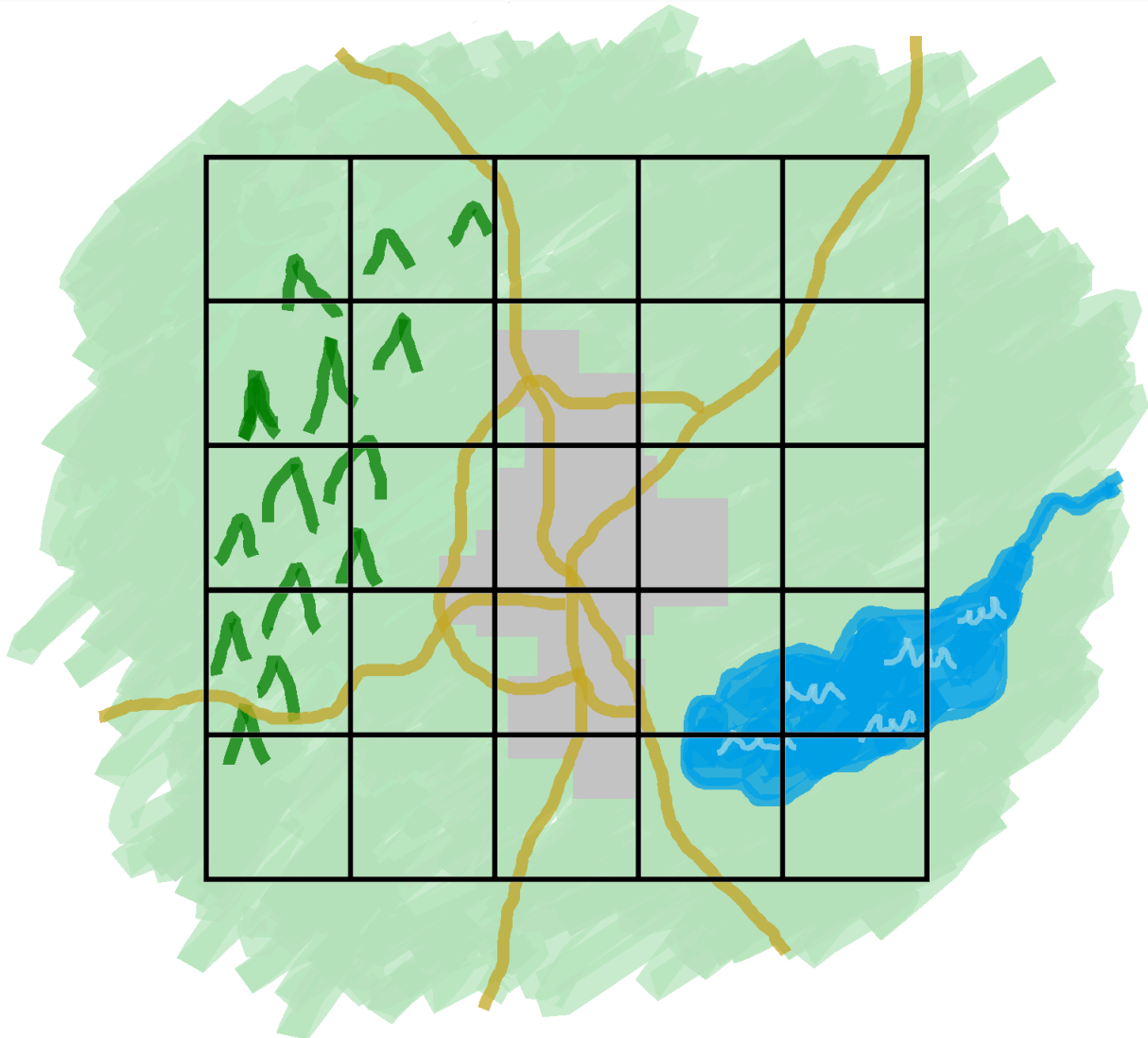
Funktionsweise von SimSpace Weaver

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

Sie definieren die Grenzen Ihrer Simulationswelt und unterteilen sie in ein Raster. Anstatt Simulationslogik zu erstellen, die auf dem gesamten Raster funktioniert, erstellen Sie Simulationslogik, die auf einer Zelle des Rasters arbeitet. In SimSpace Weaver ist eine räumliche App ein Programm, das Sie schreiben und das die Simulationslogik für eine Zelle Ihres Rasters implementiert. Dies beinhaltet die Logik für alle Entitäten in dieser Zelle. Der Besitzbereich einer räumlichen App ist die Rasterzelle, die von der räumlichen App gesteuert wird.

Note

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



Deine Simulationswelt ist in ein Raster unterteilt

Sie unterteilen Ihre Simulationswelt in ein Raster. Jede räumliche App implementiert Simulationslogik für eine einzelne Zelle in diesem Raster.

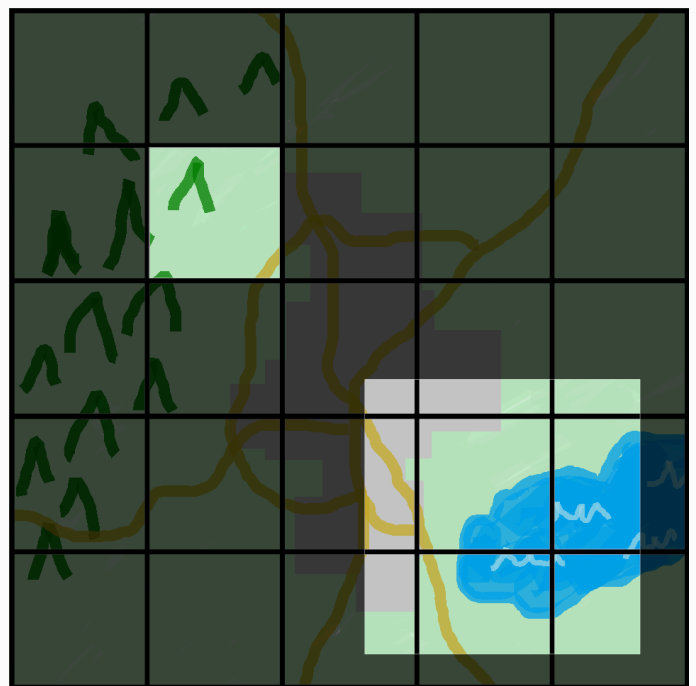
SimSpace Weaver führt eine Instanz Ihres räumlichen App-Codes für jede Zelle Ihres Rasters aus. Alle Spatial-App-Instanzen werden parallel ausgeführt. SimSpace Weaver unterteilt Ihre Gesamtsimulation im Wesentlichen in mehrere kleinere Simulationen. Jede der kleineren Simulationen behandelt einen Teil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

(Amazon EC2) -Instances (Amazon EC2) -Instances (Amazon EC2) -Instances (Amazon EC2) -Instances (AWS Cloud Ein einzelner Mitarbeiter kann mehrere räumliche Apps ausführen.

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

Subscriptions (Abonnements)

Die Sicht einer räumlichen App auf die Welt ist ihr eigener Besitzbereich. Um herauszufinden, was in einem anderen Teil der Simulationswelt passiert, erstellt die räumliche App ein Abonnement. Der Abonnementbereich ist ein Teilbereich des gesamten Bereichs Simulation World. Ein Abonnementbereich kann Teile mehrerer Eigentumsbereiche umfassen, einschließlich des eigenen Besitzbereichs der räumlichen App. SimSpace Weaver benachrichtigt die Spatial-App über alle Entitätsereignisse (z. B. Eingabe, Verlassen, Erstellen, Aktualisieren und Löschen), die innerhalb des Abonnementbereichs auftreten.



Die Sicht einer räumlichen App auf die Welt

Die Sicht einer räumlichen App auf die Welt ist ihr Besitzbereich, der eine Zelle im Weltraster darstellt.

Ansicht einer räumlichen App mit einem zusätzlichen Abonnementbereich

Eine räumliche App verwendet ein Abonnement, um herauszufinden, was in einem anderen Teil der Simulationswelt passiert. Der Abonnementbereich kann mehrere Rasterzellen und Zellenteile enthalten.

Beispielsweise benötigt eine App, die physisch interagierende Entitäten simuliert, möglicherweise Informationen über Entitäten, die sich knapp außerhalb der räumlichen Grenzen ihres Eigentumsbereichs befinden. Um dies zu erreichen, kann die App Bereiche abonnieren, die an ihr Eigentumsgebiet grenzen. Nach dem Erstellen des Abonnements erhält die App Benachrichtigungen über Entitätsereignisse in diesen Bereichen und kann die Entitäten lesen. Ein anderes Beispiel ist ein autonomes Fahrzeug, das alle Einheiten sehen muss, die sich 200 Meter vor ihm befinden, unabhängig davon, welcher App das Gebiet gehört. Die App für das Fahrzeug kann ein Abonnement mit einem Filter als achsenbündiger Begrenzungsrahmen erstellen (AABB), der den sichtbaren Bereich abdeckt.

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 kontrollieren 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. Diese Apps haben keine Eigentumsbereiche, da sie nicht räumlich sind. Nur mit einem Abonnement können sie herausfinden, was in der Simulationswelt passiert.

Wie benutzt du SimSpace Weaver

Wenn Sie es verwenden SimSpace Weaver, sind dies die wichtigsten Schritte, die Sie befolgen:

1. Schreiben und erstellen Sie 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 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. Starte deine Simulation.
6. Starten und beenden Sie Ihre benutzerdefinierten Apps nach Bedarf.
7. Connect Sie Kunden mit Ihren benutzerdefinierten Apps oder Service-Apps, um die Simulation anzusehen oder mit ihr zu interagieren.
8. Überprüfen Sie Ihre Simulationsprotokolle in AmazonCloudWatch Logs.
9. Stoppen Sie Ihre Simulation.
10. Bereinigen der 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 über das Simulationsschema finden Sie unter [SimSpace Weaver Simulationsschemareferenz](#).

Arbeiter und Ressourceneinheiten

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

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

Note

Wir haben eine Rechenressourceneinheit zuvor als Slot bezeichnet. Möglicherweise finden Sie diesen vorherigen 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 Zecken bezeichnet werden. Die Uhr kündigt allen Arbeitern die Startzeit jeder Zecke 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 Simulationszustandsdaten.

Eine Partition für eine räumliche App (auch als räumliche App-Partition oder räumliche Partition bezeichnet) enthält alle Entitäten im Besitzbereich einer räumlichen App. SimSpace Weaver platziert Entitäten auf der Grundlage der räumlichen Position jeder Entität in räumliche App-Partitionen. Das bedeutet, dass SimSpace Weaver versucht wird, Entitäten, die sich räumlich nahe beieinander befinden, auf demselben Worker zu platzieren. Dadurch wird die Menge an Wissen minimiert, die eine App von Entitäten benötigt, die sie nicht besitzt, um die Entitäten zu simulieren, die ihr gehören.

Bundesstaat Fabric

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

Die State Fabric verwendet ein benutzerdefiniertes Binärformat, das eine Entität als Satz von Anfangsdaten und ein Aktualisierungsprotokoll für jedes Datenfeld dieser Entität beschreibt.

Mit diesem Format können Sie auf den Zustand einer Entität zu einem früheren Zeitpunkt in der Simulationszeit zugreifen und ihn einem Punkt in Echtzeit zuordnen. Der Puffer hat eine endliche Größe, und es ist nicht möglich, die Zeit über das hinaus zurückzudrehen, was im Puffer enthalten ist. SimSpace Weaver verwendet für jedes Feld einen Zeiger auf den aktuellen Offset im Aktualisierungsprotokoll und aktualisiert einen Zeiger im Rahmen einer Feldaktualisierung. SimSpace Weaver ordnet diese Aktualisierungsprotokolle mithilfe von Shared Memory dem Prozessbereich einer App zu.

Dieses Objektformat hat einen geringen Aufwand und keine Serialisierungskosten. SimSpace Weaver verwendet dieses Objektformat auch, um Indexfelder zu analysieren und zu identifizieren (z. B. die Entitätsposition).

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 Orientierung), in denen Sie als persistente Daten speichern können SimSpace Weaver. Entitäten existieren innerhalb von Partitionen.

Apps

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

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

SimSpace Weaver weist jeder App das Eigentum an 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 Anwendungsfall und Lebenszyklus.

Note

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

Spatial

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

SimSpace Weaver verwaltet den Lebenszyklus von räumlichen Apps. Sie geben eine Anordnung von räumlichen App-Partitionen in Ihrem Simulationsschema an. Wenn Sie Ihre Simulation in SimSpace Weaver starten, wird für jede räumliche App-Partition eine räumliche App gestartet. Wenn Sie die Simulation beenden, werden Ihre räumlichen Apps in 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 auf die Partition einer räumlichen App zu verschieben. Dadurch wird das Eigentum an der Entität auf die räumliche 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 an eine räumliche App übertragen, um die Entität in die Simulation aufzunehmen und zu aktualisieren. Sie können einer benutzerdefinierten App einen Netzwerkendpunkt in 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 stoppen (mithilfe von SimSpace Weaver API-Aufrufen). Nachdem Sie eine benutzerdefinierte App-Instanz auf einem Worker gestartet haben, wird SimSpace Weaver die Instanz nicht an einen anderen Worker übertragen.

Apps für den Dienst

Sie können eine Service-App verwenden, wenn Sie einen schreibgeschützten Prozess benötigen, der auf jedem Worker ausgeführt wird. Sie können beispielsweise eine Service-App verwenden, wenn Sie eine umfangreiche Simulation haben und einen Viewing Client benötigen, der sich durch die Simulation bewegt 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 des zugewiesenen Workers filtern und nur die relevanten Entitäten an ihre verbundenen Clients senden. Ihr Viewing Client kann dann eine Verbindung zu verschiedenen Service-Apps herstellen, während er sich durch den Simulationsraum bewegt. Sie konfigurieren Service-Apps in Ihrem Simulationsschema. SimSpace Weaver startet und stoppt Ihre Service-Apps für Sie.

App-Zusammenfassung

In der folgenden Tabelle sind die Eigenschaften der verschiedenen SimSpace Weaver App-Typen zusammengefasst.

	Spatial	Benutzerdefinierte Apps	Apps für den Dienst
Entitäten lesen	Ja	Ja	Ja
Entitäten aktualisieren	Ja	Nein	Nein
Entitäten erstellen	Ja	Ja*	Ja*
Lebenszyklus	Verwaltet (SimSpace Weaver kontrolliert es.)	Unverwaltet (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.

	Spatial	Benutzerdefinierte Apps	Apps für den Dienst
Kunden können sich verbinden	Nein	Ja	Ja

* Wenn eine benutzerdefinierte App oder Service-App eine Entität erstellt, muss die App das Eigentum an der Entität auf eine räumliche App übertragen, damit die räumliche 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 nach den Arten von Apps, die sie enthalten: räumliche Domänen, benutzerdefinierte Domänen und Dienstdomänen. Sie konfigurieren Ihre Apps innerhalb von Domains.

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 lernen. Eine App verarbeitet Entitätsereignisse aus einem Abonnement, bevor sie Daten für Entitäten in Partitionen liest, die ihr nicht gehören.

Eine Partition kann auf demselben Worker wie die App existieren (dies wird als lokale Partition bezeichnet), aber eine andere App kann die Partition besitzen. Eine Partition kann auch auf einem anderen Worker existieren (dies wird als Remote-Partition bezeichnet). Wenn es sich bei dem Abonnement um eine Remote-Partition handelt, erstellt der Worker eine lokale Kopie der Remote-Partition durch einen Prozess, der als Replikation bezeichnet wird. Der Worker liest dann die lokale Kopie (replizierte Remote-Partition). Wenn eine andere App auf dem Worker von dieser Partition auf demselben Häkchen lesen muss, liest der Worker dieselbe lokale Kopie.

Beispielanwendungsfälle SimSpace Weaver

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

Erstellen Sie Simulationen großer Menschenmengen

Sie können SimSpace Weaver verwenden, um Menschenmengen in realen Umgebungen zu simulieren. SimSpace Weaver ermöglicht es Ihnen, Ihre Simulationen auf Millionen dynamischer Objekte 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, zur Gestaltung der Verkehrsführung und zur 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.

Einrichten für SimSpace Weaver

Um SimSpace Weaver für die erste Verwendung eingerichtet zu werden, müssen Sie Ihr AWS-Konto und Ihre lokale Umgebung einrichten. Wenn Sie fertig sind, sind Sie bereit für die [Tutorial — Erste Schritte](#).

Einrichtungsaufgaben ein

1. [Richten Sie Ihr AWS-Konto zur Verwendung ein SimSpace Weaver.](#)
2. [Einrichten der lokalen Umgebung für SimSpace Weaver.](#)

Richten Sie Ihr AWS-Konto zur Verwendung ein SimSpace Weaver

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

So melden Sie sich für ein AWS-Konto an

Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen durch.

Anmeldung für ein AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Tasten eingeben.

Wenn Sie sich für ein AWS-Konto anmelden, wird ein 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 eine Bestätigungs-E-Mail, sobald die Anmeldung abgeschlossen ist. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Einen Administratorbenutzer erstellen

Wenn Sie sich für AWS-Konto registriert haben, sichern Sie Root-Benutzer des AWS-Kontos, aktivieren Sie AWS IAM Identity Center erstellen Sie einen Administratorbenutzer, damit Sie nicht den Root-Benutzer für alltägliche Aufgaben verwenden.

Schützen Ihres Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei [AWS Management Console](#) als Kontobesitzer an, indem Sie Stammbenutzer 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-AnmeldungBenutzerhandbuch zu .

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

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

Erstellen eines Administratorbenutzers

1. IAM Identity Center aktivieren.

Eine genaue Anleitung finden Sie unter [Aktivierung von AWS IAM Identity Center](#) im AWS IAM Identity Center-Benutzerhandbuch.

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

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

Als Administratorbenutzer anmelden

- 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 unter [Anmelden beim AWS-Zugangsportale](#) im AWS-Anmeldung Benutzerhandbuch zu.

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 Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter [Erstellen einer Rolle für einen IAM-Benutzer](#) 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 Erteilung 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"
        }
    }
}
]
}

```

Einrichten der lokalen Umgebung für SimSpace Weaver

SimSpace Weaver Simulationen werden in containerisierten Amazon Linux 2 (AL2) Umgebungen ausgeführt. Sie benötigen eine AL2-Umgebung, um Ihre Apps zu kompilieren und mit dem SimSpace

Weaver App-SDK zu verknüpfen. Die lokale Standardentwicklungsumgebung ist ein AL2-Container in Docker. Wenn Sie sich dafür entscheiden Docker, es nicht zu verwenden, bieten wir alternative Anweisungen zum Ausführen einer AL2-Umgebung Windows Subsystem for Linux (WSL). Sie können auch Ihre eigene Methode verwenden, um eine lokale AL2-Umgebung zu erstellen. Weitere Möglichkeiten, AL2 lokal auszuführen, finden Sie in der [Amazon EC2 EC2-Dokumentation](#).

Important

Docker on Microsoft Windows ist die Standardentwicklungsumgebung. Der Einfachheit halber schlagen wir andere Möglichkeiten vor, Ihre lokale Entwicklungsumgebung einzurichten. Diese sind jedoch nicht Standard und werden nicht unterstützt.

Themen

- [Einrichten Amazon Linux 2 \(AL2\) in Docker](#)
- [Einrichten Amazon Linux 2 \(AL2\) in Windows Subsystem for Linux \(WSL\)](#)

Einrichten Amazon Linux 2 (AL2) in Docker

Dieser Abschnitt enthält Anweisungen zum Einrichten Ihrer lokalen AL2-Umgebung in Docker. Anweisungen zur Einrichtung von AL2 in Windows Subsystem for Linux (WSL) unter [Einrichten Amazon Linux 2 \(AL2\) in Windows Subsystem for Linux \(WSL\)](#)

Voraussetzungen

- Microsoft Windows 10 oder höher
- [Microsoft Visual Studio 2019](#) oder später, wenn der [Desktop development with C++](#) Workload installiert ist
- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)

Um AL2 einzurichten in Docker

1. Wenn Sie Ihre AWS Anmeldeinformationen für noch nicht konfiguriert haben AWS CLI, folgen Sie diesen Anweisungen: [Konfiguration der AWS-CLI](#). Wenn Sie das nur verwenden, können Sie SimSpace Weaver Sie das so konfigurieren AWS CLI, dass Ihre SimSpace Weaver Anmeldeinformationen standardmäßig verwendet werden.
2. [Laden Sie das verteilbare SimSpace Weaver App-SDK-Paket](#) herunter. Es enthält Folgendes:
 - Binärdateien und Bibliotheken für die SimSpace Weaver App-Entwicklung
 - 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 ein Docker Bild zu erstellen.

```
docker-create-image.bat
```

Note

Wenn Sie während dieses Schritts eine Fehlermeldung erhalten, stellen Sie sicher, dass der Docker Vorgang ausgeführt wird.

Einrichten Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL)

Dieser Abschnitt enthält Anweisungen zum Einrichten Ihrer lokalen AL2-Umgebung in Windows Subsystem for Linux (WSL) Anweisungen zur Einrichtung von AL2 in finden Sie Docker unter.

[Einrichten Amazon Linux 2 \(AL2\) in Docker](#)

Important

In diesem Abschnitt wird eine Lösung beschrieben, die eine Version von AL2 verwendet, die nicht Eigentum von Amazon ist, nicht von Amazon entwickelt wurde oder von Amazon nicht unterstützt wird. Diese Lösung wird nur zu Ihrer Bequemlichkeit bereitgestellt, falls Sie sie nicht verwenden Docker möchten. Amazon und AWS ich übernehmen keine Haftung, wenn Sie sich für diese Lösung entscheiden.

Voraussetzungen

- [Hyper-V auf Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- [Open-Source-AL2-Distribution eines Drittanbieters für WSL \(Download-Version 2.0.20200722.0-update.2\) \(siehe Anweisungen\)](#)

Important

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

Um AL2 einzurichten in WSL

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

```
ws1 -d Amazon2
```

Important

Verwenden Sie beim WSL Ausführen die Linux-Anweisungen anstelle der Windows-Anweisungen in diesem Handbuch (wenn Sie die Wahl haben). Wenn es keine Linux-Anweisungen gibt, verwenden Sie die folgenden Ersetzungen:

- Verwenden Sie `tools/linux` anstelle von `tools\windows`
- Verwenden Sie `.sh` Skripte anstelle von `.bat` Skripten

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

```
yum update -y
```

⚠ Important

Wenn bei diesem Schritt ein Timeout auftritt, müssen Sie möglicherweise zu diesen Verfahren wechseln WSL1 und sie erneut versuchen. 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. Laden Sie das herunter und installieren Sie es 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 zu verwendenden 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 die verteilbare SimSpace Weaver App-SDK-Datei herunter. Sie enthält Folgendes:

- Binärdateien und Bibliotheken für die SimSpace Weaver App-Entwicklung
- 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 zusätzliche Setup-Skript aus.

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

 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 dem Inhalt Ihrer Wahl zu erstellen. Im Zusammenhang mit Ihrer Nutzung von SimSpace Weaver sind Sie dafür verantwortlich, die Lizenzbedingungen aller Software oder Inhalte, die Sie in Ihren Simulationen verwenden, zu beschaffen, zu pflegen und einzuhalten. Vergewissern Sie sich, dass Ihre Lizenzvereinbarung sieht vor, dass Ihre Lizenzvereinbarung sieht vor, dass Ihre Software und Ihre Inhalte in einer virtuell gehosteten Umgebung ausgeführt werden dürfen.

Erste Schritte mit SimSpace Weaver

Dieser Abschnitt enthält Anleitungen zur Verfügung, die Ihnen den Einstieg erleichtern SimSpace Weaver. Diese Tutorials führen Sie in den allgemeinen Arbeitsablauf für die Erstellung von Simulationen mit ein SimSpace Weaver. In beiden Tutorials lernen Sie, wie Sie Simulationen erstellen, bereitstellen und ausführen SimSpace Weaver. Wir empfehlen, dass Sie mit dem Schnellstart-Tutorial beginnen, um eine Simulation in wenigen Minuten zum Laufen zu bringen. Lesen Sie anschließend das ausführliche Tutorial, um mehr über die einzelnen Schritte des Prozesses zu erfahren.

In diesen Tutorials wird eine Beispielanwendung (PathfindingSample) verwendet, die in der SimSpace Weaver App-SDK-.zip-Datei enthalten ist, die Sie während des [Einrichtungsvorgangs 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: Erstellen und Ausführen einer Simulation in wenigen Minuten](#)
- [Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung](#)

Schnellstart-Tutorial: Erstellen und Ausführen einer Simulation in wenigen Minuten

Dieses Tutorial führt Sie durch den Prozess zum Erstellen und Ausführen einer Simulation SimSpace Weaver in wenigen Minuten. Wir empfehlen Ihnen, mit diesem Tutorial zu beginnen und anschließend das ausführliche Tutorial durchzugehen.

Voraussetzungen

Bevor Sie beginnen, vergewissern Sie sich, dass Sie die in beschriebenen Schritte ausgeführt haben [Einrichten für SimSpace Weaver](#).

Schritte

- [Schritt 1: Erstellen eines Projekts](#)
- [Schritt 2: Protokollierung aktivieren \(optional\)](#)
- [Schritt 3: Ausführen des Schnellstartskripts](#)
- [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#)
- [Schritt 5: Sehen Sie sich Ihre Simulation an](#)
- [Schritt 6: Anhalten und Bereinigen Ihrer Simulation](#)

Schritt 1: Erstellen eines Projekts

Die verteilbare SimSpace Weaver App-SDK enthält ein Skript, das ein Projekt aus dem gebündelten PathfindingSample Projekt erstellt. Sie müssen das Skript von seinem Speicherort in Ihrem Dateisystem aus ausführen. Das Skript erstellt einen *project-name* innerhalb eines *Pfads* unter Verwendung der Werte, die Sie in der Befehlszeile angeben.

Docker

Ein Projekt erstellen

1. Wechseln Sie an einer Windows Eingabeaufforderung in Ihren Projektordner.

```
cd sdk-folder
```

2. Führen Sie das `create-project.bat`-Skript aus.

```
.\create-project.bat --name project-name --path path
```

Die vollständige Befehls- und Parameterliste folgt.

```
.\create-project.bat --name project-name --path path --app-sdk-version version-number --template template-name --overwriteproject
```


⚠ Important

Verwenden Sie einen Projektnamen nicht länger als 20 Zeichen. Eine Überschreitung dieses Limits kann zu Fehlern führen.

ℹ Note

Wenn Sie eine Aufforderung erhalten, ein Laufwerk mit zu teilenDocker, wählen Sie Yes die Freigabe aus.

⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `---profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch.

WSL**⚠ Important**

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Ein Projekt erstellen

1. Wechseln Sie bei einer Linux Shell-Eingabeaufforderung zu Ihrem Projektordner.

```
cd sdk-folder
```

2. Führen Sie das `create-project.sh`-Skript aus.

```
./create-project.sh --name project-name --path path
```

Die vollständige Befehls- und Parameterliste folgt.

```
./create-project.sh --name project-name --path path --profile cli-profile-name  
--app-sdk-version version-number --template template-name --overwriteproject
```

Important

Verwenden Sie einen Projektnamen nicht länger als 20 Zeichen. Eine Überschreitung dieses Limits kann zu Fehlern führen.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Parameter

name

Der Name Ihres Projekts.

path

Der Speicherort des Projekts in Ihrem Dateisystem. Die Projektstruktur kann manchmal lange Dateipfade haben, die die Pfadlängenbeschränkung in Ihrem Betriebssystem überschreiten können. Wir empfehlen Ihnen, einen Pfadnamen zu verwenden, der so kurz wie möglich ist.

profile

Der Name des AWS CLI Profils, das das Skript für die Authentifizierung verwenden soll. Weitere Informationen finden Sie unter [Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch. Dieser Parameter ist nur für das SimSpace Weaver App-SDK Version 1.12.1 oder höher verfügbar. Weitere Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#).

app-sdk-version

(optional) Die Version des SimSpace Weaver App-SDK, die Ihr Projekt verwendet. Sie erstellen und verknüpfen Ihre Apps mit dieser Version. Wenn das Skript die Version am Speicherort der verteilbaren Datei nicht findet oder Sie die Versionsnummer nicht angeben, lädt das Skript automatisch die neueste Version herunter.

template

(optional) Eine Projektvorlage, mit der das Skript Ihr Projekt erstellt. Wenn Sie keine Vorlage angeben, verwendet das Skript die `PathfindingSample`. Zulässige Werte:

- **PathfindingSample** – eine Beispielanwendung, die einen einzelnen [Worker](#) verwendet.
- **MultiWorkerPathfindingSample** – eine Version der Beispielanwendung, die mehrere [Worker](#) verwendet.

overwriteproject

(optional) Verwenden Sie diese Option, um einen vorhandenen Projektordner mit demselben Projektnamen und Pfad zu überschreiben.

Schritt 2: Protokollierung aktivieren (optional)

Die Protokollierung ist für das PathfindingSample Projekt standardmäßig deaktiviert. In dieser Anleitung wird davon ausgegangen, dass die Protokollierung aktiviert ist. Sie können sich dafür entscheiden, die Protokollierung zu aktivieren, dies ist jedoch optional.

Important

Das PathfindingSample generiert große Mengen an Protokolldaten. Wenn Sie sich dafür entscheiden, die Protokollierung zu aktivieren, werden Ihnen Abrechnungsgebühren für die Protokolldaten berechnet. Sie erhalten weiterhin Abrechnungsgebühren für die Protokolldaten, solange sie existieren. Wir empfehlen Ihnen dringend, diese Simulation zu beenden und die Bereinigungsschritte am Ende des Tutorials so bald wie möglich durchzuführen, wenn Sie die Protokollierung aktivieren.

Um die Protokollierung zu aktivieren

1. Öffnen Sie die folgende Datei in einem Texteditor:

Docker

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

Note

Ihr *Projektordner path**project-name* verwendet die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

WSL

Important

Wir stellen Ihnen diese Anweisungen der Einfachheit halber zur Verfügung. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) bestimmt und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

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

Note

Ihr *Projektordner path/project-name* verwendet die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

- Suchen Sie den `simulation_properties`: Abschnitt am Anfang der Datei:

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

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

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

- 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>"
```

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

Schritt 3: Ausführen des Schnellstartskripts

Die Beispielanwendung enthält ein Schnellstartskript. Das Skript erstellt, erstellt, lädt und startet Ihre Simulation und ihre Apps.

Docker

So führen Sie das Schnellstartskript aus


- Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:


```
cd project-folder\tools\windows
```

2. Führen Sie das Schnellstartskript für dieses Projekt aus.

```
.\quick-start-project-name-cli.bat
```

 **Important**

Ab Version 1.12.3 startet das `quick-start`Skript Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den `--maximum-duration` Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).

 **Important**

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch.

WSL

⚠ Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So führen Sie das Schnellstartskript aus

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Führen Sie das Schnellstartskript für dieses Projekt aus.

```
./quick-start-project-name-cli.sh
```

⚠ Important

Ab Version 1.12.3 startet das `quick-start` Skript Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den `--maximum-duration` Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).

⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen

zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Das Skript startet eine Schleife und stoppt automatisch, nachdem alle Komponenten sindSTARTED. Suchen Sie nach einer Ausgabe, die der folgenden ähnelt:

```
[2022-10-04T22:15:28] [INFO] Describe Simulation Results:
[2022-10-04T22:15:28] [INFO] {
  "Status": "STARTED",
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
```

Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer

Sie müssen die IP-Adresse und Portnummer Ihrer Ansichts-App (benutzerdefinierte App) abrufen, damit Sie eine Verbindung mit der Simulation herstellen können. Im folgenden Verfahren wird davon ausgegangen, dass Sie nichts über Ihre Simulation wissen (z. B. den Simulationsnamen). Sie können dieses Verfahren jederzeit verwenden, um die IP-Adresse und Portnummer für eine benutzerdefinierte App oder Service-App zu finden. Die folgende Beispielausgabe bezieht sich auf ein Projekt mit dem Namen MyProject.

Docker

So rufen Sie Ihre IP-Adresse und Portnummer ab

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr `project-folder path\project-name` verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie die `ListSimulations -API`, um den Namen Ihrer Simulation abzurufen.

```
.\weaver-project-name-cli.bat list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

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"
    }
  ]
}

```

3. Verwenden Sie die `DescribeSimulation` -API, um eine Liste der Domains in Ihrer Simulation abzurufen.

```

.\weaver-project-name-cli.bat describe-simulation --simulation simulation-name

```

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

Beispielausgabe:

```

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

```

4. Verwenden Sie die `ListApps` -API, um eine Liste der benutzerdefinierten Apps in einer Domain abzurufen. Der Domänenname für die Ansichts-App (benutzerdefinierte) im Beispielprojekt lautet `MyViewDomain`. Suchen Sie in der Ausgabe nach dem App-Namen.

```
.\weaver-project-name-cli.bat 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"  
    }  
  ]  
}
```

5. Verwenden Sie die DescribeApp-API, um die IP-Adresse und die Portnummer abzurufen. Für das Beispielprojekt lautet der Domänenname MyViewDomain und der App-Name ViewApp.

```
.\weaver-project-name-cli.bat describe-app --simulation simulation-name --  
domain domain-name --app app-name
```

Die IP-Adresse und die Portnummer befinden sich im -EndpointInfoBlock 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, die Clients zur Verbindung mit Ihrer App zur SimSpace Weaver Verfügung stellt. SimSpace Weaver ordnet den `Declared` Port dem `Actual` Port zu.

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So rufen Sie Ihre IP-Adresse und Portnummer ab

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie die ListSimulations -API, um den Namen Ihrer Simulation abzurufen.

```
./weaver-project-name-cli.sh list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

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"
    }
  ]
}
```

```
}
```

3. Verwenden Sie die `DescribeSimulation` -API, um eine Liste der Domains in Ihrer Simulation abzurufen.

```
./weaver-project-name-cli.sh describe-simulation --simulation simulation-name
```

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

Beispielausgabe:

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

4. Verwenden Sie die `ListApps` -API, um eine Liste der benutzerdefinierten Apps in einer Domain abzurufen. Der Domänenname für die Ansichts-App (benutzerdefinierte) im Beispielprojekt lautet `MyViewDomain`. Suchen Sie in der Ausgabe nach dem App-Namen.

```
./weaver-project-name-cli.sh 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"
    }
  ]
}
```

5. Verwenden Sie die `DescribeApp`-API, um die IP-Adresse und die Portnummer abzurufen. Für das Beispielprojekt lautet der Domänenname `MyViewDomain` und der App-Name `ViewApp`.

```
./weaver-project-name-cli.sh describe-app --simulation simulation-name --
domain domain-name --app app-name
```

Die IP-Adresse und die Portnummer befinden sich im `-EndpointInfoBlock` 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, die Clients zur Verbindung mit Ihrer App zur SimSpace Weaver Verfügung stellt. SimSpace Weaver ordnet den `Declared` Port dem `Actual` Port zu.

Schritt 5: 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 keinen lokalen Support für die Unreal Engine Entwicklung haben. In den Anweisungen für den Unreal Engine Client wird davon ausgegangen, dass Sie ihn verwendenWindows.

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.

Windows console client

Voraussetzungen

- Microsoft Windows 10 oder höher
- [Microsoft Visual Studio 2019](#) oder später, wenn der [Desktop development with C++ Workload](#) installiert ist
- [CMake3](#)
- [Git](#)


Um mit dem Beispielkonsolen-Client eine Verbindung zur Beispielanwendung herzustellen

1. Gehen Sie in einem Befehlszeilenfenster zum Ordner für den Konsolen-Client (im App-SDK-Ordner).


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

2. Wird verwendetCMake3, um eine Visual Studio Lösung in diesem Ordner zu erstellen.

```
cmake .
```

 Note

Stellen Sie sicher, dass Sie am Ende das Leerzeichen und den Punkt angeben.

 Important

Lassen Sie das Befehlszeilenfenster für weitere Schritte geöffnet.

3. Öffnen Sie in Visual Studio das `PathfindingSampleConsoleClient.sln`, das Sie im vorherigen Schritt erstellt haben.
4. Wählen Sie die `RelWithDebInfoBuild`-Konfiguration aus.
5. Wählen Sie `Build > Build Solution`.
6. Gehen Sie in Ihrem vorherigen Befehlszeilenfenster zum Build-Ausgabeordner im Ordner für den Konsolen-Client.

```
cd RelWithDebInfo
```

7. Führen Sie den Client mit der IP-Adresse und Portnummer Ihres `ausViewApp`.

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

In Ihrem Befehlszeilenfenster sollten Zahlen für Ereignisse zum Aktualisieren, Löschen und Erstellen von Entitäten angezeigt werden, ähnlich der folgenden Beispielausgabe.

Note

Die IP-Adressen und Portnummern in der folgenden Beispielausgabe sind Platzhalter. Geben Sie dem Konsolen-Client Ihre IP-Adresse und Portnummer ViewApp an. Geben Sie die Actual Portnummer an, wenn Sie eine Verbindung zu einem herstellen möchten ViewApp, der in der ausgeführt wird AWS Cloud. Geben Sie die IP-Adresse und die Portnummer 127.0.0. 1:7000 an, wenn Sie eine Verbindung zu einem herstellen ViewApp, der auf Ihrem lokalen System ausgeführt wird. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

```
##PathfindingSample#ViewApp Message Reader##

Added argument url:tcp://198.51.100.135:4321
Some subscription arguments are missing, restoring defaults.

*****
Sample usage without a MoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50
Sample usage with CircleMoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --circle-center-y 500 --circle-speed 0.001

*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
```

```
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

Note

Anleitungen zur Problembeseitigung finden Sie unter [PathfindingSample Der Konsolentient kann keine Verbindung herstellen](#)

8. Drücken Sie CTRL+C, um den Konsolen-Client zu beenden.

Linux console client

Important

Wir stellen Ihnen diese Anweisungen der Einfachheit halber zur Verfügung. In einigen Linux Umgebungen funktionieren sie möglicherweise nicht. Diese Verfahren werden nicht unterstützt.

Bei diesem Verfahren wird davon ausgegangen, dass Sie ausschließlich in einer Linux Umgebung arbeiten. Sie können Ihre Simulation auch mit integrierten Clients anzeigenWindows.

Voraussetzungen

- CMake3
- CCompiler (bereits enthalten) Amazon Linux 2
- Git

Um mit dem Beispielkonsolen-Client eine Verbindung zur Beispielanwendung herzustellen

1. Wechseln Sie an einer LinuxShell-Eingabeaufforderung zum Ordner für den Konsolen-Client (im App-SDK-Ordner).

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

2. Erstellen Sie einen Build-Ordner.

```
mkdir build
```

3. Gehe zum Build-Ordner.

```
cd build
```

4. Wird verwendetCMake3, um den Client zu erstellen.

```
cmake3 ../ && cmake3 --build .
```

Note

Stellen Sie sicher, dass Sie am Ende das Leerzeichen und den Punkt angeben.

5. Führen Sie den Client mit der IP-Adresse und der Portnummer Ihres ausViewApp.

```
./ConsoleClient --url tcp://ip-address:port-number
```

In Ihrem Befehlszeilenfenster sollten Zahlen für Ereignisse zum Aktualisieren, Löschen und Erstellen von Entitäten angezeigt werden, ähnlich der folgenden Beispielausgabe.

Note

Die IP-Adressen und Portnummern in der folgenden Beispielausgabe sind Platzhalter. Geben Sie dem Konsolen-Client Ihre IP-Adresse und Portnummer ViewApp an. Geben Sie die Actual Portnummer an, wenn Sie eine Verbindung zu einem herstellen möchtenViewApp, der in der ausgeführt wirdAWS Cloud. Geben Sie die IP-Adresse und die Portnummer 127.0.0. 1:7000 an, wenn Sie eine Verbindung zu einem herstellenViewApp, der auf Ihrem lokalen System ausgeführt wird. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321
```

```
Some subscription arguments are missing, restoring defaults.
```

```
*****
```

```
Sample usage without a MoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50
Sample usage with CircleMoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --
circle-center-y 500 --circle-speed 0.001

*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

Note

Anleitungen zur Problembhebung finden Sie unter. [PathfindingSample Der Konsolencient kann keine Verbindung herstellen](#)

6. Drücken Sie CTRL+C, um den Konsolen-Client zu beenden.

Unreal Engine on Windows

Voraussetzungen

- Unreal Engine 5 Entwicklungsumgebung
- Microsoft .NET Framework 4.8 Developer Pack
- Windows Konsolen-Client (siehe die Registerkarte „Windows Konsolen-Client“ auf dieser Seite)

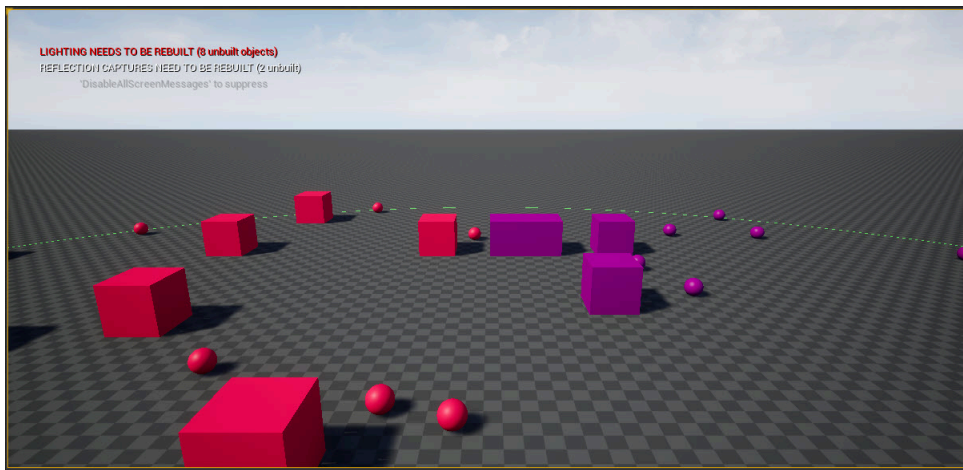
⚠ Important

Andere Versionen von Unreal Engine und .NET werden nicht unterstützt und können Probleme verursachen.

Um mit dem Unreal Beispielclient eine Verbindung zur Beispielanwendung herzustellen

1. Der Unreal Engine Client verwendet die NNG Bibliothek vom Konsolen-Client aus. Sie müssen den Konsolenclient für erstellenWindows, falls Sie ihn noch nicht erstellt haben. Weitere Informationen finden Sie auf der Registerkarte „WindowsKonsolen-Client“ auf dieser Seite.
2. Gehen Sie in einem Dateimanager-Fenster zu `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Öffnen Sie `UnrealClient.uproject`.
4. Wenn der Editor Sie fragt, ob Sie die `UnrealClient` Module neu erstellen möchten, wählen Sie `yes`.
5. Öffnen Sie in einem Texteditor `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`.
6. Aktualisieren Sie die URL mit der IP-Adresse und der Portnummer für Ihre View-App: `tcp://ip-address:port-number` (sie sollte so aussehen `tcp://198.51.100.135:1234`).
7. Wählen Sie im UnrealEditor `play`.

Ihr Unreal Editor sollte eine Visualisierung der Simulation anzeigen, die dem folgenden Screenshot ähnelt.



Note

Abhängig von der Leistungsfähigkeit Ihres lokalen Entwicklungssystems kann es einige Minuten dauern, bis der Unreal Editor die Simulation anzeigt. Während dieser Zeit scheint das System einzufrieren.

Verwenden Sie die D Tasten W, A, S, , , , , , , um sich im Unreal Client zu bewegen. Halten Sie die Maustaste gedrückt und ziehen Sie die Maus, um sich zu drehen.

Sie können die Taste [(linke eckige Klammer) drücken, um den Abonnementbereich zu verkleinern. Sie können die Taste] (rechte eckige Klammer) drücken, um den Abonnementbereich zu vergrößern. Die Größe des Abonnementbereichs bestimmt die Anzahl der Entitäten, die im Client angezeigt werden.

Sie können die C Taste drücken, um eine Entität in der Simulation zu erstellen. Der Client sendet einen `CreateEntity` Befehl an die View-App. Die View-App erstellt dann die Entität und überträgt sie in die räumliche Domäne.

Sie können den Code für `ViewAppDriver::HandleEntityCreationRequests` in `project-folder\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp`, um zu sehen, wie die App diesen Prozess implementiert.

Schritt 6: Anhalten und Bereinigen Ihrer Simulation

Es ist wichtig, Ihre Simulationen zu bereinigen, wenn Sie sie nicht mehr benötigen. SimSpace Weaver Simulationsressourcen werden auf Ihre Service Quotas (Limits) angerechnet, auch

wenn Ihre Simulation gestoppt wird. Sie erhalten weiterhin Abrechnungsgebühren für laufende Simulationen. Möglicherweise erhalten Sie auch Abrechnungsgebühren für die Datenspeicherung in unterstützenden Services wie Amazon CloudWatch Logs und Amazon Simple Storage Service. Weitere Informationen zu SimSpace Weaver Servicekontingenten finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Befolgen Sie die Verfahren in diesem Abschnitt, wenn Sie bereit sind, Ihre Simulation zu bereinigen.

⚠ Important

Sie können eine angehaltene Simulation nicht neu starten.

⚠ Important

Sie können eine gelöschte Simulation nicht wiederherstellen.

Simulationsressourcen in bereinigen SimSpace Weaver

Sie müssen Ihre Simulation beenden, bevor Sie sie löschen können. Durch das Löschen einer Simulation werden SimSpace Weaver nur Ressourcen in entfernt. Sie müssen separate Schritte ausführen, um Ressourcen zu löschen, die Ihre Simulation in anderen Services erstellt oder verwendet (siehe den folgenden Abschnitt).

Docker

So bereinigen Sie Ihre Simulation

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr `project-folder path\project-name` verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Suchen Sie die Namen Ihrer Simulationen.

```
.\weaver-project-name-cli.bat list-simulations
```


⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface-Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

3. Halten Sie eine Simulation an.

```
.\weaver-project-name-cli.bat stop-simulation --simulation simulation-name
```

4. Löschen Sie eine angehaltene Simulation.

```
.\weaver-project-name-cli.bat delete-simulation --simulation simulation-name
```

WSL**⚠ Important**

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So bereinigen Sie Ihre Simulation

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Suchen Sie die Namen Ihrer Simulationen.

```
./weaver-project-name-cli.sh list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

3. Halten Sie eine Simulation an.

```
./weaver-project-name-cli.sh stop-simulation --simulation simulation-name
```

4. Löschen Sie eine angehaltene Simulation.

```
./weaver-project-name-cli.sh delete-simulation --simulation simulation-name
```

AWS Management Console

So bereinigen Sie Ihre Simulation

1. Öffnen Sie die -SimSpace WeaverKonsole unter [SimSpace Weaver der -Konsole](#).
2. Simulations.
3. Wählen Sie in der Simulations Liste die Option neben dem Namen der Simulation aus, die Sie löschen möchten.
4. Wenn der Status der von Ihnen ausgewählten Simulation istSTARTED:
 - a. Wählen Sie das DropdownActions-Menü aus.
 - b. Wählen Sie Stop.
 - c. Geben Sie zur Bestätigung Ihren Simulationsnamen ein.
 - d. Wählen Sie Stop.
 - e. Warten Sie, bis der Status Ihrer Simulation istSTOPPED.
5. Wählen Sie das DropdownActions-Menü aus.
6. Wählen Sie Delete.
7. Wählen Sie zur Bestätigung Delete.

Simulationsressourcen in unterstützenden Services bereinigen

Zur Unterstützung Ihrer Simulation SimSpace Weaver erstellt Ressourcen in anderen -Services. SimSpace Weaverlöscht diese Ressourcen nicht, wenn Sie Ihre Simulation löschen. Sie können diese zusätzlichen Ressourcen löschen, wenn Sie sie nicht benötigen.

Important

Möglicherweise fallen Gebühren für jede dieser Ressourcen an, die Sie nicht löschen.

So löschen Sie Support-Ressourcen für Ihr Projekt

1. Wenn Sie mit Ihrem Projekt fertig sind, löschen Sie den zugehörigen AWS CloudFormationStack. Weitere Informationen zum Arbeiten mit AWS CloudFormationfinden Sie unter [Löschen eines Stacks auf der -AWS CloudFormationKonsole](#) im AWS CloudFormation -Benutzerhandbuch.
 - weaver-*project-name* -stack

⚠ Important

Simulationen, die Sie mit demselben Projekt gestartet haben, teilen sich Ressourcen wie die App-Rolle. Wenn Sie den AWS CloudFormationStack löschen, löschen Sie die App-Rolle. Löschen Sie Ihren AWS CloudFormationStack nicht, wenn Sie andere Simulationen haben, die dieselben Ressourcen gemeinsam nutzen.

ℹ Note

Ihr AWS CloudFormationStack wird wahrscheinlich melden, DELETE_FAILED da er AmazonAmazon S3-Buckets nicht löschen kann, die nicht leer sind. Im folgenden Schritt löschen Sie Ihre Amazon S3-Buckets.

2. Wenn Sie mit Ihrem Projekt fertig sind, löschen Sie dessen Amazon S3-Bucket(s). Weitere Informationen zum Arbeiten mit Amazon S3-Buckets finden Sie unter [Löschen eines Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.

- `weaver-lowercase-project-name-account-number-region`

Das Projekt mit dem Namen MyProject im Konto 111122223333 in der us-west-2 Region hat beispielsweise den folgenden Bucket:

- `weaver-myproject-111122223333-us-west-2`

ℹ Note

Sie müssen den Inhalt eines Amazon S3-Buckets löschen, bevor Sie den Bucket löschen können.

ℹ Note

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

- `weaver-lowercase-project-name-account-number -app-zips-region`
- `weaver-lowercase-project-name-account-number -schemas-region`

3. Wenn Sie die Protokollierung für Ihre Simulation aktiviert haben, löschen Sie die CloudWatch Protokollgruppe. Weitere Informationen zum Arbeiten mit CloudWatch Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollstreams](#) im Amazon- CloudWatch Logs- Benutzerhandbuch.

Der Name der Protokollgruppe für Ihre Simulation wird in ihrem Schema (Konfigurationsdatei) angegeben: `project-folder\tools\project-name.yaml`

Der Name der Protokollgruppe ist der Wert von `log_destination_resource_name`. Der folgende Schemaausschnitt zeigt, dass die Protokollgruppe für die Beispielanwendung `istMySimulationLogs`.

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

Warning

Wenn Sie mehrere Simulationen starten, die dieselbe Protokollgruppe angeben, werden die Protokolldaten für alle diese Simulationen an dieselbe Protokollgruppe weitergeleitet. Wenn Sie die Protokollgruppe löschen, löschen Sie die Protokolldaten für alle Simulationen, die diese Protokollgruppe verwenden. Wenn Sie eine Protokollgruppe für eine laufende Simulation löschen, schlägt die Simulation fehl.

Important

Wenn das Schema Ihrer Simulation `log_destination_service: "logs"` und eine `log_destination_resource_name` angibt, Sie die Protokollgruppe `log_destination_resource_name` jedoch nicht in

CloudWatch Logs finden können, stellen Sie sicher, dass Sie dieselbe überprüfenAWS-Region, in der Ihre Simulation ausgeführt wurde.

Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung

Dieses Tutorial führt Sie durch dasselbe allgemeine Verfahren, das im Schnellstart-Tutorial vorgestellt wurde, jedoch ausführlicher. Das Schnellstart-Tutorial vereinfachte viele Schritte und versteckte Details innerhalb der Automatisierung. Dieses Tutorial zeigt und erklärt diese Details.

Wir empfehlen, dass Sie das [Schnellstart-Tutorial](#) abschließen, bevor Sie dieses Tutorial durchgehen.

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in beschriebenen Schritte ausgeführt haben [Einrichten für SimSpace Weaver](#).

Schritte

- [Schritt 1: Erstellen eines Projekts](#)
- [Schritt 2: Protokollierung aktivieren \(optional\)](#)
- [Schritt 3: Hochladen Ihres Simulationsschemas](#)
- [Schritt 4: Erstellen Erstellen: Erstellen: Erstellen](#)
- [Schritt 5: Hochladen von Apps](#)
- [Schritt 6: Starten Ihrer Simulation](#)
- [Schritt 7: Simulationsdetails abrufen](#)
- [Schritt 8: Starten benutzerdefinierter Apps](#)
- [Schritt 9: Starten der Uhr](#)
- [Schritt 10: Überprüfen der Protokolle](#)
- [Schritt 11: Sehen Sie sich Ihre Simulation an](#)
- [Schritt 12: Anhalten und Bereinigen Ihrer Simulation](#)

Schritt 1: Erstellen eines Projekts

Die verteilbare SimSpace Weaver App-SDK enthält ein Skript, das ein Projekt aus dem gebündelten PathfindingSample Projekt erstellt. Sie müssen das Skript von seinem Speicherort in Ihrem Dateisystem aus ausführen. Das Skript erstellt einen *project-name* innerhalb eines *Path* unter Verwendung der Werte, die Sie in der Befehlszeile angeben.

Docker

Ein Projekt erstellen

1. Wechseln Sie an einer Windows Eingabeaufforderung in Ihren Projektordner.

```
cd sdk-folder
```

2. Führen Sie das `create-project.bat`-Skript aus.

```
.\create-project.bat --name project-name --path path
```

Die vollständige Befehls- und Parameterliste folgt.

```
.\create-project.bat --name project-name --path path --app-sdk-version version-number --template template-name --overwriteproject
```

Important

Verwenden Sie einen Projektnamen nicht länger als 20 Zeichen. Eine Überschreitung dieses Limits kann zu Fehlern führen.

Note

Wenn Sie eine Aufforderung erhalten, ein Laufwerk mit zu teilenDocker, wählen Sie Yes die Freigabe aus.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK

Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch.

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Ein Projekt erstellen

1. Wechseln Sie bei einer Linux Shell-Eingabeaufforderung zu Ihrem Projektordner.

```
cd sdk-folder
```

2. Führen Sie das `create-project.sh`-Skript aus.

```
./create-project.sh --name project-name --path path
```

Die vollständige Befehls- und Parameterliste folgt.

```
./create-project.sh --name project-name --path path --profile cli-profile-name  
--app-sdk-version version-number --template template-name --overwriteproject
```


⚠ Important

Verwenden Sie einen Projektnamen nicht länger als 20 Zeichen. Eine Überschreitung dieses Limits kann zu Fehlern führen.

⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Parameter**name**

Der Name Ihres Projekts.

path

Der Speicherort des Projekts in Ihrem Dateisystem. Die Projektstruktur kann manchmal lange Dateipfade haben, die die Pfadlängenbeschränkung in Ihrem Betriebssystem überschreiten können. Wir empfehlen Ihnen, einen Pfadnamen zu verwenden, der so kurz wie möglich ist.

profile

Der Name des AWS CLI Profils, das das Skript für die Authentifizierung verwenden soll. Weitere Informationen finden Sie unter [Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#)

im AWS Command Line Interface -Benutzerhandbuch. Dieser Parameter ist nur für das SimSpace Weaver App-SDK Version 1.12.1 oder höher verfügbar. Weitere Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#).

app-sdk-version

(optional) Die Version des SimSpace Weaver App-SDK, die Ihr Projekt verwendet. Sie erstellen und verknüpfen Ihre Apps mit dieser Version. Wenn das Skript die Version am Speicherort der verteilbaren Datei nicht findet oder Sie die Versionsnummer nicht angeben, lädt das Skript automatisch die neueste Version herunter.

template

(optional) Eine Projektvorlage, mit der das Skript Ihr Projekt erstellt. Wenn Sie keine Vorlage angeben, verwendet das Skript die `PathfindingSample`. Zulässige Werte:

- **PathfindingSample** – eine Beispielanwendung, die einen einzelnen [Worker](#) verwendet.
- **MultiWorkerPathfindingSample** – eine Version der Beispielanwendung, die mehrere [Worker](#) verwendet.

overwriteproject

(optional) Verwenden Sie diese Option, um einen vorhandenen Projektordner mit demselben Projektnamen und Pfad zu überschreiben.

Schritt 2: Protokollierung aktivieren (optional)

Die Protokollierung ist für das `PathfindingSample` Projekt standardmäßig deaktiviert. In dieser Anleitung wird davon ausgegangen, dass die Protokollierung aktiviert ist. Sie können sich dafür entscheiden, die Protokollierung zu aktivieren, dies ist jedoch optional.

Important


Das `PathfindingSample` generiert große Mengen an Protokolldaten. Wenn Sie sich dafür entscheiden, die Protokollierung zu aktivieren, werden Ihnen Abrechnungsgebühren für die Protokolldaten berechnet. Sie erhalten weiterhin Abrechnungsgebühren für die Protokolldaten, solange sie existieren. Wir empfehlen Ihnen dringend, diese Simulation zu beenden und die Bereinigungsschritte am Ende des Tutorials so bald wie möglich durchzuführen, wenn Sie die Protokollierung aktivieren.

Um die Protokollierung zu aktivieren

1. Öffnen Sie die folgende Datei in einem Texteditor:


Docker

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

 Note


Ihr *Projektordner path**project-name* verwendet die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

WSL

 Important

Wir stellen Ihnen diese Anweisungen der Einfachheit halber zur Verfügung. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) bestimmt und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

project-folder/tools/*project-name*-schema.yaml

 Note

Ihr *Projektordner path*/*project-name* verwendet die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

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

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

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

```
log_destination_service: "logs"
```

```
log_destination_resource_name: "MySimulationLogs"
```

4. 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>"
```

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

Schritt 3: Hochladen Ihres Simulationsschemas

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

Docker

Die Beispielanwendung verfügt über ein vorkonfiguriertes Schema. Sie finden die Schemadatei der Beispielanwendung im Ordner Tools für Ihr Projekt:

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

So laden Sie Ihr Schema hoch

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr `project-folder path\project-name` verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das Hilfsskript, um das Schema hochzuladen.

```
.\upload-schema-project-name.bat
```

⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

WSL

⚠ Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Die Beispielanwendung verfügt über ein vorkonfiguriertes Schema. Sie finden die Schemadatei der Beispielanwendung im Ordner Tools für Ihr Projekt:

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

So laden Sie Ihr Schema hoch

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr `project-folder path/project-name` verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das Hilfsskript, um das Schema hochzuladen.

```
./upload-schema-project-name.sh
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Schritt 4: Erstellen Erstellen: Erstellen: Erstellen

Sie sind jetzt bereit, die räumlichen und benutzerdefinierten Apps für das Beispielprojekt zu erstellen. Das Beispielprojekt enthält ein Hilfsskript, das diese Apps für Sie erstellt.

Docker

Das Skript startet einen Docker-Container mit dem Docker-Image, das Sie beim [Einrichten Ihrer lokalen Umgebung](#) erstellt haben. Das Skript führt Ihren Build in einer Amazon Linux-Umgebung im Docker-Container aus. Es schreibt die Build-Artefakte und ihre Abhängigkeiten in Ihren `project-folder\build` Ordner in Windows.

Um dein Projekt zu erstellen

1. Wenn Sie noch nicht dort sind, gehen Sie zum Tools-Ordner für Ihr Projekt und Ihre Plattform. Sie `path\project-name` verwenden `project-folder` die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

Geben Sie in der Windows-Befehlszeile Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das Hilfsskript, um das Projekt zu erstellen.

```
.\build-project-name.bat
```

WSL

Important

Wir werden hier auch zu Informationsmöglichkeiten zusammengefasst. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Das Skript schreibt die Build-Artefakte und ihre Abhängigkeiten in Ihren `project-folder/build` Ordner.

Um dein Projekt zu erstellen

1. Wenn Sie noch nicht dort sind, wechseln Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. YOUR `project-folder/path/project-name` verwendet die Werte, die Sie bei der Erstellung des Projekts angegeben haben.

Geben Sie an einer Linux-Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das Hilfsskript, um das Projekt zu erstellen.

```
./build-project-name.sh
```

Schritt 5: Hochladen von Apps

Das Build-Skript verpackt Ihre Apps als ZIP-Dateien. Sie müssen diese ZIP-Dateien in bestimmte Buckets in Amazon Simple Storage Service hochladen, um Ihre SimSpace Weaver Simulation in der Cloud auszuführen. Das SimSpace Weaver App-SDK stellt ein Hilfsskript für den Upload bereit.

Docker

So laden Sie Ihre Apps hoch

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das Hilfsskript, um Ihre Apps hochzuladen.

```
.\upload-app-project-name.bat
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für](#)

[Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So laden Sie Ihre Apps hoch

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das Hilfsskript, um Ihre Apps hochzuladen.

```
./upload-app-project-name.sh
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden

Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Überprüfen Ihrer Amazon S3-Ressourcen

Sie können Ihre Amazon 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-Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Für die Beispielanwendung verwenden Ihr Schema (das Sie in einem vorherigen Schritt hochgeladen haben) und Ihre App-Ressourcen die folgenden Namensformate:

- Schema-Bucket: `simspaceweaver-project-name-lowercase-account-number-schemas-region`
 - Schemadatei: `project-name-schema.yaml`
- App-Bucket: `simspaceweaver-project-name-lowercase-account-number-app-zips-region`
 - Räumliche App: `project-nameSpatial.zip`
 - Anzeigen der (benutzerdefinierten) App: `project-nameView.zip`

Zum Beispiel unter Berücksichtigung der folgenden Projekteigenschaften:

- Projektname: MyProject
- AWS Kontonummer: 111122223333
- AWS-Region: us-west-2

Die Schema- und App-Ressourcen hätten die folgenden Namen:

- Schema-Bucket: `simspaceweaver-myproject-111122223333-schemas-us-west-2`
 - Schemadatei: `MyProject-schema.yaml`
- App-Bucket: `simspaceweaver-myproject-111122223333-apps-zips-us-west-2`
 - Räumliche App: `MyProjectSpatial.zip`

- Anzeigen (benutzerdefinierter) App: MyProjectView.zip

Schritt 6: Starten Ihrer Simulation

Das SimSpace Weaver App-SDK bietet ein Hilfsskript, um Ihre Simulation zu starten. Das Skript ist ein Wrapper um einen StartSimulation API-Aufruf. Es stellt die folgenden Parameter für den API-Aufruf bereit:

- Der Name des Simulationsschemas (das Sie in einem früheren Schritt hochgeladen haben)
- Der Simulationsname
- Der SimSpace Weaver Service-Endpunkt

Docker

So starten Sie Ihre Simulation

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:


```
cd project-folder\tools\windows
```

2. Verwenden Sie das Hilfsskript, um Ihre Simulation hochzuladen und zu starten.


```
.\start-simulation-project-name.bat
```

Important

Ab Version 1.12.3 startet das start-simulationSkript Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den --maximum-duration Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).


 Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

 Note

`run-project-name.bat` ist ein alternatives Hilfsskript, das auch die Simulationsuhr startet. In diesem Tutorial starten Sie die Uhr separat in einem späteren Schritt. Ab Version 1.12.3 startet das `runSkript` Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den `--maximum-duration` Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).

WSL

 Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So starten Sie Ihre Simulation


1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:


```
cd project-folder/tools/linux
```

2. Verwenden Sie das Hilfsskript, um Ihre Simulation hochzuladen und zu starten.

```
./start-simulation-project-name.sh
```


 Important

Ab Version 1.12.3 startet das `start-simulation`-Skript Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den `--maximum-duration` Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).

 Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen

zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

 Note

`run-project-name.sh` ist ein alternatives Hilfsskript, das auch die Simulationsuhr startet. In diesem Tutorial starten Sie die Uhr separat in einem späteren Schritt. Ab Version 1.12.3 startet das `run` Skript Ihre Simulation mit einer maximalen Dauer von 1 Stunde. Sie können den `--maximum-duration` Parameter verwenden, um eine andere maximale Dauer anzugeben. In Version 1.12.2 oder niedriger können Sie keine maximale Dauer für das Skript angeben und Simulationen haben eine maximale Dauer von 14 Tagen. Weitere Informationen zur maximalen Dauer einer Simulation finden Sie unter [Maximale Dauer einer Simulation](#).

Das Skript wird in Schleife geschaltet, bis der Simulationsstatus entweder `STARTED` oder `istFAILED`. Es kann einige Minuten dauern, bis eine Simulation gestartet wird. Wenn Ihre Simulation erfolgreich gestartet wird, wird eine Ausgabe ähnlich der folgenden angezeigt:

```
[2022-10-04T22:15:28] [INFO] Describe Simulation Results:
[2022-10-04T22:15:28] [INFO] {
  "Status": "STARTED",
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
```

```
"BucketName": "weaver-myproject-111122223333-us-west-2"  
},
```

Note

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

- `weaver-lowercase-project-name-account-number-app-zips-region`
- `weaver-lowercase-project-name-account-number-schemas-region`

Schritt 7: Simulationsdetails abrufen

Das SimSpace Weaver App-SDK bietet ein Hilfsskript, das das umschließt AWS CLI. Das Skript vereinfacht Aufrufe an , AWS CLI indem es den SimSpace Weaver Service-Endpunkt bereitstellt. Sie verwenden dieses Hilfsskript, um die SimSpace Weaver APIs aufzurufen. Die DescribeSimulation API enthält Details zu Ihrer Simulation, einschließlich ihres Status. Eine Simulation kann sich in einem der folgenden Zustände befinden:

Status des Simulationszyklus

1. **STARTING** – Anfangsstatus nach dem Aufruf von StartSimulation
2. **STARTED** – alle räumlichen Apps werden gestartet und sind fehlerfrei
3. **STOPPING** – Anfangsstatus nach dem Aufruf von StopSimulation
4. **STOPPED** – Alle Rechenressourcen werden gestoppt
5. **DELETING** – Anfangsstatus nach dem Aufruf von DeleteSimulation
6. **DELETED** – Alle Ressourcen, die der Simulation zugewiesen sind, werden gelöscht
7. **FAILED** – Die Simulation hatte einen kritischen Fehler/Fehler und wurde gestoppt
8. **SNAPSHOT_IN_PROGRESS** – Ein [Snapshot](#) wird gerade erstellt

Docker

So rufen Sie Ihre Simulationsdetails ab

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr `project-folder path\project-name` verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das CLI-Hilfsskript, um die ListSimulations-API aufzurufen.

```
.\weaver-project-name-cli.bat list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Das Skript sollte Details zu Ihren 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"
  }

```

3. Rufen Sie auf `DescribeSimulation`, um Ihre Simulationsdetails zu erhalten. Ersetzen Sie *simulation-name* durch die Name Ihrer Simulation aus der Ausgabe des vorherigen Schritts.

```

.\weaver-project-name-cli.bat describe-simulation --simulation simulation-name

```

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

```

{
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "ExecutionId": "1a2b3c4d-0ab1-1234-567a-12ab34cd5e6f",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "Status": "STARTED",
  "TargetStatus": "STARTED",
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
    "BucketName": "weaver-myproject-111122223333-us-west-2"
  },
  "SchemaError": "[]",
  "LoggingConfiguration": {
    "Destinations": [
      {
        "CloudWatchLogsLogGroup": {
          "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
        }
      }
    ]
  },
}

```

```
"LiveSimulationState": {
  "Domains": [
    {
      "Type": "",
      "Name": "MySpatialSimulation",
      "Lifecycle": "Unknown"
    },
    {
      "Type": "",
      "Name": "MyViewDomain",
      "Lifecycle": "ByRequest"
    }
  ],
  "Clocks": [
    {
      "Status": "STARTED",
      "TargetStatus": "STARTED"
    }
  ]
},
"MaximumDuration": "1H",
"StartError": "[]"
}
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So rufen Sie Ihre Simulationsdetails ab

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das CLI-Hilfsskript, um die ListSimulations-API aufzurufen.

```
./weaver-project-name-cli.sh list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Das Skript sollte Details zu Ihren 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"
}
```

3. Rufen Sie auf `DescribeSimulation`, um Ihre Simulationsdetails zu erhalten. Ersetzen Sie durch *simulation-name* die Name Ihrer Simulation aus der Ausgabe des vorherigen Schritts.

```
./weaver-project-name-cli.sh describe-simulation --simulation simulation-name
```

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

```
{
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "ExecutionId": "1a2b3c4d-0ab1-1234-567a-12ab34cd5e6f",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "Status": "STARTED",
  "TargetStatus": "STARTED",
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
    "BucketName": "weaver-myproject-111122223333-us-west-2"
  },
  "SchemaError": "[]",
  "LoggingConfiguration": {
    "Destinations": [
      {
        "CloudWatchLogsLogGroup": {
          "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
        }
      }
    ]
  },
  "LiveSimulationState": {
    "Domains": [
      {
        "Type": "",
        "Name": "MySpatialSimulation",
        "Lifecycle": "Unknown"
      },
      {
        "Type": "",
        "Name": "MyViewDomain",
```

```
        "Lifecycle": "ByRequest"
      }
    ],
    "Clocks": [
      {
        "Status": "STARTED",
        "TargetStatus": "STARTED"
      }
    ]
  },
  "MaximumDuration": "1H",
  "StartError": "[]"
}
```

Schritt 8: Starten benutzerdefinierter Apps

SimSpace Weaver verwaltet nicht den Lebenszyklus benutzerdefinierter Apps. Sie müssen Ihre benutzerdefinierten Apps starten. Es hat sich bewährt, Ihre benutzerdefinierten Apps zu starten, bevor Sie Ihre Simulationsuhr starten, aber Sie können benutzerdefinierte Apps starten, nachdem Sie die Uhr gestartet haben.

Sie können das CLI-Hilfsskript verwenden, um die StartApp -API zum Starten Ihrer benutzerdefinierten Apps aufzurufen.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripts verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen](#)

[für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

Docker

```
.\weaver-project-name-cli.bat start-app --simulation simulation-name --name app-name --domain domain-name
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
./weaver-project-name-cli.sh start-app --simulation simulation-name --name app-name --domain domain-name
```

Der StartApp API-Aufruf erstellt und startet eine neue Instance der benutzerdefinierten App mit dem von Ihnen angegebenen Namen. Wenn Sie den Namen einer bereits vorhandenen App angeben, erhalten Sie eine Fehlermeldung. Wenn Sie eine bestimmte App (Instance) neu starten möchten, müssen Sie diese App zunächst beenden und löschen.

Note

Der Status Ihrer Simulation muss lauten, STARTED bevor Sie benutzerdefinierte Apps starten können. Informationen zum Überprüfen des Status Ihrer Simulation finden Sie unter [Schritt 7: Simulationsdetails abrufen](#).

Die Beispielanwendung stellt die ViewApp benutzerdefinierte App bereit, um Ihre Simulation anzuzeigen. Diese App stellt Ihnen eine statische IP-Adresse und Portnummer zur Verfügung, um die Simulationsclients zu verbinden (dies werden Sie in einem späteren Schritt in diesem Tutorial tun). Sie können sich eine domain als eine Klasse von Apps vorstellen, die denselben

ausführbaren Code und dieselben Startoptionen haben. Der app name identifiziert die Instance der App. Weitere Informationen zu SimSpace Weaver Konzepten finden Sie unter [Die wichtigsten Konzepte für SimSpace Weaver](#).

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

Docker

```
.\weaver-project-name-cli.bat describe-app --simulation simulation-name --app app-name --domain domain-name
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
./weaver-project-name-cli.sh describe-app --simulation simulation-name --app app-name --domain domain-name
```

Docker

So starten Sie die Ansichts-App in diesem Tutorial

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das CLI-Hilfsskript, um StartApp für aufzurufen ViewApp.

```
.\weaver-project-name-cli.bat start-app --simulation simulation-name --name  
ViewApp --domain MyViewDomain
```

3. Rufen Sie auf `DescribeApp`, um den Status Ihrer benutzerdefinierten App zu überprüfen.

```
.\weaver-project-name-cli.bat describe-app --simulation simulation-name --app  
ViewApp --domain MyViewDomain
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So starten Sie die Ansichts-App in diesem Tutorial

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das CLI-Hilfsskript, um `StartApp` für `ViewApp` aufzurufen.

```
./weaver-project-name-cli.sh start-app --simulation simulation-name --name  
ViewApp --domain MyViewDomain
```

3. Rufen Sie auf `DescribeApp`, um den Status Ihrer benutzerdefinierten App zu überprüfen.

```
./weaver-project-name-cli.sh describe-app --simulation simulation-name --app  
ViewApp --domain MyViewDomain
```


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

```
{
  "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, die Clients zur Verbindung mit Ihrer App zur SimSpace Weaver Verfügung stellt. SimSpace Weaver ordnet den `Declared` Port dem `Actual` Port zu.

Note

Sie können das [Verfahren aus dem Schnellstart-Tutorial](#) verwenden, um die IP-Adresse und Portnummer jeder gestarteten benutzerdefinierten App unabhängig von diesem Workflow abzurufen.

Schritt 9: Starten der Uhr

Wenn Sie Ihre Simulation zum ersten Mal erstellen, hat sie eine Uhr, aber die Uhr wird nicht ausgeführt. Wenn Ihre Uhr nicht ausgeführt wird, aktualisiert Ihre Simulation ihren Status nicht. Nachdem Sie die Uhr gestartet haben, beginnt sie mit dem Senden von Häkchen an Ihre Apps. Jedes Häkchen, Ihre räumlichen Apps durchlaufen 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.

⚠ Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Docker

So starten Sie die Uhr

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Verwenden Sie das CLI-Hilfsskript, um die StartClock-API aufzurufen.

```
.\weaver-project-name-cli.bat start-clock --simulation simulation-name
```

Note

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

```
.\weaver-project-name-cli.bat list-simulations
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So starten Sie die Uhr

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Verwenden Sie das CLI-Hilfsskript, um die StartClock-API aufzurufen.

```
./weaver-project-name-cli.sh start-clock --simulation simulation-name
```

Note

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

```
./weaver-project-name-cli.sh list-simulations
```

Schritt 10: Überprüfen der Protokolle

SimSpace Weaver schreibt Simulationsverwaltungsnachrichten und die Konsolenausgabe aus Ihren Apps in Amazon CloudWatch Logs. Weitere Informationen zum Arbeiten mit Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollstreams](#) im Amazon- CloudWatch Logs-Benutzerhandbuch.

Jede Simulation, die Sie erstellen, hat eine eigene Protokollgruppe in - CloudWatch Protokollen. Der Name der Protokollgruppe wird im Simulationsschema angegeben. Im folgenden Schemaausschnitt `log_destination_service` ist der Wert von `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 finden, nachdem Sie sie gestartet haben.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Docker

```
project-folder\tools\windows\weaver-project-name-cli.bat describe-simulation --  
simulation simulation-name
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
project-folder/tools/linux/weaver-project-name-cli.sh describe-simulation --  
simulation simulation-name
```

Das folgende Beispiel zeigt den Teil der Ausgabe von DescribeSimulation, der die Protokollierungskonfiguration beschreibt. Der Name der Protokollgruppe wird am Ende der angezeigtLogGroupArn.

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

Jede Simulationsprotokollgruppe enthält mehrere Protokollstreams:

- Verwaltungsprotokollstream – Simulationsverwaltungsnachrichten, die vom SimSpace Weaver Service erstellt wurden.

```
/sim/management
```

- Protokollstream für Fehler – vom SimSpace Weaver Service 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
```

- Protokollstreams für räumliche Anwendungen (1 für jede räumliche App auf jedem Worker) – Konsolenausgabe, die von räumlichen Apps erstellt wird. Jede räumliche App schreibt in ihren eigenen Protokollstream. Die *spatial-app-id* besteht aus allen Zeichen nach dem abschließenden Schrägstrich am Ende der *worker-id*.

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

- Benutzerdefinierte App-Protokollstreams (1 für jede benutzerdefinierte App-Instance) – Konsolenausgabe, die von benutzerdefinierten Apps erstellt wird. Jede benutzerdefinierte App-Instance schreibt in ihren eigenen Protokollstream.

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

- Protokollstreams der Service-App (1 für jede Service-App-Instance) – Konsolenausgabe, die von Service-Apps erzeugt wird. Jede Service-App schreibt in einen eigenen Protokollstream. Der *service-app-id* besteht aus allen Zeichen nach dem abschließenden Schrägstrich am Ende des *service-app-name*.

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

Note

Die Beispielanwendung verfügt nicht über Service-Apps.

Schritt 11: 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 keinen lokalen Support für die Unreal Engine Entwicklung haben. In den Anweisungen für den Unreal Engine Client wird davon ausgegangen, dass Sie ihn verwendenWindows.

Der Konsolenclient zeigt eine Liste von Entitätseignissen an, sobald sie auftreten. Der Client erhält die Informationen zum Entitätseignis vonViewApp. 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 Entitätseignisse.

Windows console client

Voraussetzungen

- Microsoft Windows 10oder höher

- [Microsoft Visual Studio 2019](#) oder später, wenn der [Desktop development with C++](#) Workload installiert ist
- [CMake3](#)
- [Git](#)

Um mit dem Beispielkonsolen-Client eine Verbindung zur Beispielanwendung herzustellen

1. Gehen Sie in einem Befehlszeilenfenster zum Ordner für den Konsolen-Client (im App-SDK-Ordner).

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

2. Wird verwendet CMake3, um eine Visual Studio Lösung in diesem Ordner zu erstellen.

```
cmake .
```

Note

Stellen Sie sicher, dass Sie am Ende das Leerzeichen und den Punkt angeben.

Important

Lassen Sie das Befehlszeilenfenster für weitere Schritte geöffnet.

3. Öffnen Sie in Visual Studio das `PathfindingSampleConsoleClient.sln`, das Sie im vorherigen Schritt erstellt haben.
4. Wählen Sie die `RelWithDebInfoBuild`-Konfiguration aus.
5. Wählen Sie `Build > Build Solution`.
6. Gehen Sie in Ihrem vorherigen Befehlszeilenfenster zum Build-Ausgabeordner im Ordner für den Konsolen-Client.

```
cd RelWithDebInfo
```

7. Führen Sie den Client mit der IP-Adresse und Portnummer Ihres `ausViewApp`.


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

In Ihrem Befehlszeilenfenster sollten Zahlen für Ereignisse zum Aktualisieren, Löschen und Erstellen von Entitäten angezeigt werden, ähnlich der folgenden Beispielausgabe.

Note

Die IP-Adressen und Portnummern in der folgenden Beispielausgabe sind Platzhalter. Geben Sie dem Konsolen-Client Ihre IP-Adresse und Portnummer ViewApp an. Geben Sie die Actual Portnummer an, wenn Sie eine Verbindung zu einem herstellen möchten ViewApp, der in der ausgeführt wird AWS Cloud. Geben Sie die IP-Adresse und die Portnummer 127.0.0. 1:7000 an, wenn Sie eine Verbindung zu einem herstellen ViewApp, der auf Ihrem lokalen System ausgeführt wird. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321  
Some subscription arguments are missing, restoring defaults.
```

```
*****
```

```
Sample usage without a MoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50
```

```
Sample usage with CircleMoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --circle-center-y 500 --circle-speed 0.001
```

```
*****
```

```
Starting NNG client. NNG version: 1.2.4
```


```
Creating socket ...done.
```

```
Connecting to View App ... done.
```

```
Initiating connection to tcp:// 198.51.100.135:4321 ... done.
```

```
Receiving messages ...
```


```
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

 Note

Anleitungen zur Problembhebung finden Sie unter. [PathfindingSample Der Konsolenclient kann keine Verbindung herstellen](#)

8. Drücken Sie CTRL+C, um den Konsolen-Client zu beenden.

Linux console client

 Important

Wir stellen Ihnen diese Anweisungen der Einfachheit halber zur Verfügung. In einigen Linux Umgebungen funktionieren sie möglicherweise nicht. Diese Verfahren werden nicht unterstützt.

Bei diesem Verfahren wird davon ausgegangen, dass Sie ausschließlich in einer Linux Umgebung arbeiten. Sie können Ihre Simulation auch mit integrierten Clients anzeigenWindows.

Voraussetzungen

- CMake3
- CCompiler (bereits enthalten) Amazon Linux 2
- Git

Um mit dem Beispielkonsolen-Client eine Verbindung zur Beispielanwendung herzustellen

1. Wechseln Sie an einer LinuxShell-Eingabeaufforderung zum Ordner für den Konsolen-Client (im App-SDK-Ordner).

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

- Erstellen Sie einen Build-Ordner.


```
mkdir build
```

- Gehe zum Build-Ordner.

```
cd build
```

- Wird verwendet CMake3, um den Client zu erstellen.

```
cmake3 ../ && cmake3 --build .
```


 Note

Stellen Sie sicher, dass Sie am Ende das Leerzeichen und den Punkt angeben.

- Führen Sie den Client mit der IP-Adresse und der Portnummer Ihres ausViewApp.

```
./ConsoleClient --url tcp://ip-address:port-number
```

In Ihrem Befehlszeilenfenster sollten Zahlen für Ereignisse zum Aktualisieren, Löschen und Erstellen von Entitäten angezeigt werden, ähnlich der folgenden Beispielausgabe.

 Note

Die IP-Adressen und Portnummern in der folgenden Beispielausgabe sind Platzhalter. Geben Sie dem Konsolen-Client Ihre IP-Adresse und Portnummer ViewApp an. Geben Sie die Actual Portnummer an, wenn Sie eine Verbindung zu einem herstellen möchten ViewApp, der in der ausgeführt wird AWS Cloud. Geben Sie die IP-Adresse und die Portnummer 127.0.0. 1:7000 an, wenn Sie eine Verbindung zu einem herstellen ViewApp, der auf Ihrem lokalen System ausgeführt wird. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321  
Some subscription arguments are missing, restoring defaults.
```

```
*****
```

```
Sample usage without a MoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50
```

```
Sample usage with CircleMoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --circle-center-y 500 --circle-speed 0.001
```

```
*****
```

```
Starting NNG client. NNG version: 1.2.4
```

```
Creating socket ...done.
```

```
Connecting to View App ... done.
```

```
Initiating connection to tcp:// 198.51.100.135:4321 ... done.
```

```
Receiving messages ...
```

```
[2022-10-04 19:13:00.710] CreateEntity Count: 72
```

```
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
```

```
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
```

```
[2022-10-04 19:13:03.690] CreateEntity Count: 11
```

```
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
```

```
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
```

```
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

Note

Anleitungen zur Problembhebung finden Sie unter. [PathfindingSample Der Konsolenclient kann keine Verbindung herstellen](#)

6. Drücken Sie CTRL+C, um den Konsolen-Client zu beenden.

Unreal Engine on Windows

Voraussetzungen

- Unreal Engine 5 Entwicklungsumgebung
- Microsoft .NET Framework 4.8 Developer Pack
- WindowsKonsolen-Client (siehe die Registerkarte „WindowsKonsolen-Client“ auf dieser Seite)

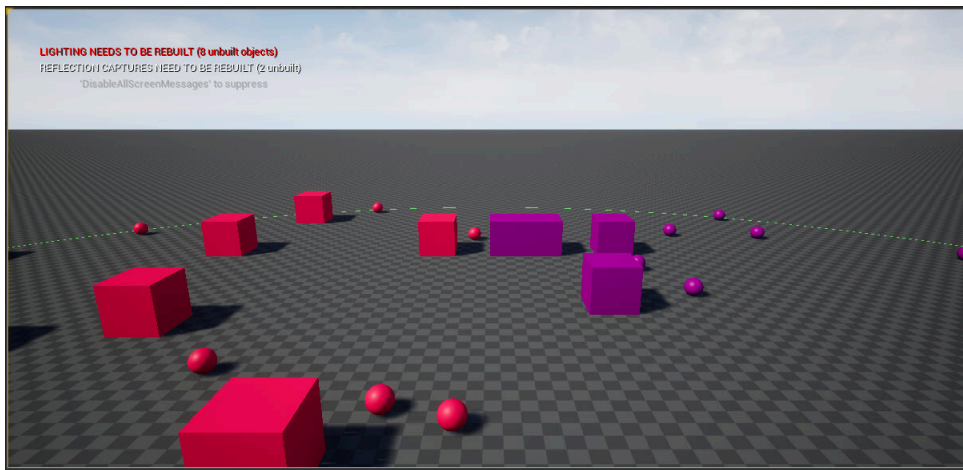
Important

Andere Versionen von Unreal Engine und .NET werden nicht unterstützt und können Probleme verursachen.

So stellen Sie mit dem Unreal Beispielclient eine Verbindung zur Beispielanwendung her

1. Der Unreal Engine Client verwendet die NNG Bibliothek vom Konsolen-Client aus. Sie müssen den Konsolenclient für erstellenWindows, falls Sie ihn noch nicht erstellt haben. Weitere Informationen finden Sie auf der Registerkarte „WindowsKonsolen-Client“ auf dieser Seite.
2. Gehen Sie in einem Dateimanager-Fenster zu `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Öffnen Sie `UnrealClient.uproject`.
4. Wenn der Editor Sie fragt, ob Sie die UnrealClient Module neu erstellen möchten, wählen Sie yes.
5. Öffnen Sie in einem Texteditor `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`.
6. Aktualisieren Sie die URL mit der IP-Adresse und der Portnummer für Ihre View-App: `tcp://ip-address:port-number` (sie sollte so aussehen `tcp://198.51.100.135:1234`).
7. Wählen Sie im UnrealEditor play.

Ihr Unreal Editor sollte eine Visualisierung der Simulation anzeigen, die dem folgenden Screenshot ähnelt.



Note

Abhängig von der Leistungsfähigkeit Ihres lokalen Entwicklungssystems kann es einige Minuten dauern, bis der Unreal Editor die Simulation anzeigt. Während dieser Zeit scheint das System einzufrieren.

Verwenden Sie die D Tasten W, A, S, , , , , , , , um sich im Unreal Client zu bewegen. Halten Sie die Maustaste gedrückt und ziehen Sie die Maus, um sich zu drehen.

Sie können die Taste [(linke eckige Klammer) drücken, um den Abonnementbereich zu verkleinern. Sie können die Taste] (rechte eckige Klammer) drücken, um den Abonnementbereich zu vergrößern. Die Größe des Abonnementbereichs bestimmt die Anzahl der Entitäten, die im Client angezeigt werden.

Sie können die C Taste drücken, um eine Entität in der Simulation zu erstellen. Der Client sendet einen `CreateEntity` Befehl an die View-App. Die View-App erstellt dann die Entität und überträgt sie in die räumliche Domäne.

Sie können den Code für `ViewAppDriver::HandleEntityCreationRequests` in `project-folder\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp`, um zu sehen, wie die App diesen Prozess implementiert.

Note

Wenn Sie die IP-Adresse und die Portnummer Ihrer View-App nicht kennen, können Sie das [Verfahren aus dem Schnellstart-Tutorial](#) verwenden, um diese Informationen abzurufen.

Schritt 12: Anhalten und Bereinigen Ihrer Simulation

Es ist wichtig, Ihre Simulationen zu bereinigen, wenn Sie sie nicht mehr benötigen. SimSpace Weaver Simulationsressourcen werden auf Ihre Service Quotas (Limits) angerechnet, auch wenn Ihre Simulation gestoppt wird. Sie erhalten weiterhin Abrechnungsgebühren für laufende Simulationen. Möglicherweise erhalten Sie auch Abrechnungsgebühren für die Datenspeicherung in unterstützenden Services wie Amazon CloudWatch Logs und Amazon Simple Storage Service. Weitere Informationen zu SimSpace Weaver Servicekontingenten finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Befolgen Sie die Verfahren in diesem Abschnitt, wenn Sie bereit sind, Ihre Simulation zu bereinigen.

⚠ Important

Sie können eine angehaltene Simulation nicht neu starten.

⚠ Important

Sie können eine gelöschte Simulation nicht wiederherstellen.

Simulationsressourcen in bereinigen SimSpace Weaver

Sie müssen Ihre Simulation beenden, bevor Sie sie löschen können. Durch das Löschen einer Simulation werden SimSpace Weaver nur Ressourcen in entfernt. Sie müssen separate Schritte ausführen, um Ressourcen zu löschen, die Ihre Simulation in anderen Services erstellt oder verwendet (siehe den folgenden Abschnitt).

Docker

So bereinigen Sie Ihre Simulation

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path\project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Windows Eingabeaufforderung Folgendes ein:

```
cd project-folder\tools\windows
```

2. Suchen Sie die Namen Ihrer Simulationen.

```
.\weaver-project-name-cli.bat list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profils mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch.

3. Halten Sie eine Simulation an.

```
.\weaver-project-name-cli.bat stop-simulation --simulation simulation-name
```

4. Löschen Sie eine angehaltene Simulation.


```
.\weaver-project-name-cli.bat delete-simulation --simulation simulation-name
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

So bereinigen Sie Ihre Simulation

1. Wenn Sie noch nicht vorhanden sind, gehen Sie zum Ordner Tools für Ihr Projekt und Ihre Plattform. Ihr *project-folder path/project-name* verwendet die Werte, die Sie beim Erstellen des Projekts angegeben haben.

Geben Sie an einer Linux Shell-Eingabeaufforderung Folgendes ein:

```
cd project-folder/tools/linux
```

2. Suchen Sie die Namen Ihrer Simulationen.

```
./weaver-project-name-cli.sh list-simulations
```

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von

im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

3. Halten Sie eine Simulation an.

```
./weaver-project-name-cli.sh stop-simulation --simulation simulation-name
```

4. Löschen Sie eine angehaltene Simulation.

```
./weaver-project-name-cli.sh delete-simulation --simulation simulation-name
```

AWS Management Console

So bereinigen Sie Ihre Simulation

1. Öffnen Sie die -SimSpace WeaverKonsole unter [SimSpace Weaver der -Konsole](#).
2. Simulations.
3. Wählen Sie in der Simulations Liste die Option neben dem Namen der Simulation aus, die Sie löschen möchten.
4. Wenn der Status der ausgewählten Simulation istSTARTED:
 - a. Wählen Sie das DropdownActions-Menü aus.
 - b. Wählen Sie Stop.
 - c. Geben Sie zur Bestätigung Ihren Simulationsnamen ein.
 - d. Wählen Sie Stop.
 - e. Warten Sie, bis der Status Ihrer Simulation istSTOPPED.
5. Wählen Sie das DropdownActions-Menü aus.
6. Wählen Sie Delete.
7. Wählen Sie zur Bestätigung Delete.

Simulationsressourcen in unterstützenden Services bereinigen

Zur Unterstützung Ihrer Simulation SimSpace Weaver erstellt Ressourcen in anderen Services. SimSpace Weaverlöscht diese Ressourcen nicht, wenn Sie Ihre Simulation löschen. Sie können diese zusätzlichen Ressourcen löschen, wenn Sie sie nicht benötigen.

⚠ Important

Möglicherweise fallen Gebühren für eine dieser Ressourcen an, die Sie nicht löschen.

So löschen Sie Support-Ressourcen für Ihr Projekt

1. Wenn Sie mit Ihrem Projekt fertig sind, löschen Sie den zugehörigen AWS CloudFormationStack. Weitere Informationen zum Arbeiten mit AWS CloudFormation finden Sie unter [Löschen eines Stacks auf der -AWS CloudFormationKonsole](#) im AWS CloudFormation -Benutzerhandbuch.

- `weaver-project-name-stack`

⚠ Important

Simulationen, die Sie mit demselben Projekt gestartet haben, teilen sich Ressourcen wie die App-Rolle. Wenn Sie den AWS CloudFormationStack löschen, löschen Sie die App-Rolle. Löschen Sie Ihren AWS CloudFormationStack nicht, wenn Sie andere Simulationen haben, die dieselben Ressourcen gemeinsam nutzen.

i Note

Ihr AWS CloudFormationStack wird wahrscheinlich melden, DELETE_FAILED da er Amazon S3-Buckets, die nicht leer sind, nicht löschen kann. Im folgenden Schritt löschen Sie Ihre Amazon S3-Buckets.

2. Wenn Sie mit Ihrem Projekt fertig sind, löschen Sie dessen Amazon S3-Bucket(s). Weitere Informationen zum Arbeiten mit Amazon S3-Buckets finden Sie unter [Löschen eines Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.

- `weaver-lowercase-project-name-account-number-region`

Das Projekt mit dem Namen MyProject im Konto 111122223333 in der us-west-2 Region hat beispielsweise den folgenden Bucket:

- `weaver-myproject-111122223333-us-west-2`

Note

Sie müssen den Inhalt eines Amazon S3-Buckets löschen, bevor Sie den Bucket löschen können.

Note

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

- `weaver-lowercase-project-name-account-number -app-zips-region`
- `weaver-lowercase-project-name-account-number -schemas-region`

3. Wenn Sie die Protokollierung für Ihre Simulation aktiviert haben, löschen Sie die CloudWatch Protokollgruppe . Weitere Informationen zum Arbeiten mit CloudWatch Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollstreams](#) im Amazon- CloudWatch Logs- Benutzerhandbuch.

Der Name der Protokollgruppe für Ihre Simulation wird in ihrem Schema (Konfigurationsdatei) angegeben: `project-folder\tools\project-name.yaml`

Der Name der Protokollgruppe ist der Wert von `log_destination_resource_name`. Der folgende Schemaausschnitt zeigt, dass die Protokollgruppe für die Beispielanwendung `istMySimulationLogs`.

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

⚠ Warning

Wenn Sie mehrere Simulationen starten, die dieselbe Protokollgruppe angeben, werden die Protokolldaten für alle diese Simulationen an dieselbe Protokollgruppe weitergeleitet. Wenn Sie die Protokollgruppe löschen, löschen Sie die Protokolldaten für alle Simulationen, die diese Protokollgruppe verwenden. Wenn Sie eine Protokollgruppe für eine laufende Simulation löschen, schlägt die Simulation fehl.

⚠ Important

Wenn das Schema Ihrer Simulation `log_destination_service: "logs"` und eine angibt, Sie die Protokollgruppe `log_destination_resource_name` jedoch nicht in CloudWatch Logs finden können, stellen Sie sicher, dass Sie dieselbe überprüfenAWS-Region, in der Ihre Simulation ausgeführt wurde.

Arbeiten mit SimSpace Weaver

Dieses Kapitel enthält Informationen und Anleitungen, die Sie beim Erstellen Ihrer eigenen Anwendungen unterstützen SimSpace Weaver.

Themen

- [Konfigurieren Ihrer Simulation](#)
- [Maximale Dauer einer Simulation](#)
- [Entwicklung Apps](#)
- [Entwicklung von Client-Anwendungen](#)
- [Lokale Entwicklung](#)
- [AWS SimSpace Weaver -App-SDK](#)
- [AWS SimSpace Weaver Demo-Framework](#)
- [Arbeiten mit Service Quotas](#)
- [Debuggen von Simulationen](#)
- [Benutzerdefinierte Container](#)
- [Verwenden von Python](#)
- [Unterstützung für andere Engines](#)
- [Verwendung von lizenzierter Software mit AWS SimSpace Weaver](#)
- [Verwaltung Ihrer Ressourcen mit AWS CloudFormation](#)
- [Snapshots](#)
- [Messaging](#)

Konfigurieren Ihrer Simulation

Ein Simulationsschema (oder Schema) ist eine YAML-formatierte Textdatei, die die Konfiguration für eine Simulation angibt. Sie können dasselbe Schema verwenden, um mehrere Simulationen zu starten. Die Schemadatei befindet sich im Projektordner für Ihre Simulation. Sie können die Datei mit einem beliebigen Texteditor bearbeiten. Liest Ihr Schema SimSpace Weaver nur, wenn es die Simulation startet. Alle Änderungen, die Sie an einer Schemadatei vornehmen, wirken sich nur auf neue Simulationen aus, die Sie nach den Änderungen starten.

Docker

Um Ihre Simulation zu konfigurieren, bearbeiten Sie Ihre Simulationsschemadatei:

```
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 im Rahmen seines Prozesses hoch, um Ihre Simulation zu erstellen:

```
project-folder\tools\windows\quick-start-project-name-cli.bat
```

Sie können auch das Upload-Schema-Hilfsskript für Ihr Projekt verwenden, wenn Sie das Schnellstartskript nicht zum Erstellen Ihrer Simulation verwenden:

```
project-folder\tools\windows\upload-schema-project-name.bat
```

WSL

Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit bestimmt Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Sie konfigurieren Ihre Simulation, indem Sie ihre Simulationsschemadatei bearbeiten:

```
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 im Rahmen seines Prozesses hoch, um Ihre Simulation zu erstellen:

```
project-folder/tools/linux/quick-start-project-name-cli.sh
```

Sie können auch das Upload-Schema-Hilfsskript für Ihr Projekt verwenden, wenn Sie das Schnellstartskript nicht zum Erstellen Ihrer Simulation verwenden:

```
project-folder/tools/linux/upload-schema-project-name.sh
```

Important


Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei](#) im AWS Command Line Interface -Benutzerhandbuch.

Simulationskonfigurationsparameter

Das Simulationsschema enthält Bootstrapping-Informationen, darunter:

- Simulationseigenschaften – SDK-Version und Rechenkonfiguration (Typ und Anzahl der [Worker](#))
- Uhrzeiten – Tickrate und Toleranzen
- Strategien zur räumlichen Partitionierung – räumliche Topologie (z. B. ein Raster), Grenzen und Platzierungsgruppen (räumliche Partitionsgruppierung auf Auftragnehmern)
- Domains 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 von Ihnen angegebenen Tickrate voranzubringen.

 Note

Das create-project-Skript im SimSpace Weaver App-SDK generiert automatisch ein Simulationsschema für Sie, basierend auf der Beispielanwendung.

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

Themen

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

SDK-Version

Das `sdk_version` Feld gibt die Version von anSimSpace Weaver, für die das Schema formatiert ist. Gültige Werte: 1.16, 1.15, 1.14, 1.13, 1.12

 Important

Der Wert von `sdk_version` enthält 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`. Dies bedeutet, dass der Wert von `log_destination_resource_name` ist der Name einer Protokollgruppe in `AmazonCloudWatchLogs`.

Note

Die Protokollierung ist optional. Wenn Sie keine Protokollzeileigenschaften konfigurieren, erzeugt Ihre Simulation keine Protokolle.

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

Workers

Dieser Abschnitt gibt den Typ und die Anzahl der Arbeiter an, die Sie für Ihre Simulation benötigen. `SimSpace Weaver` verwendet eigene Worker-Typen, die den Amazon EC2-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 mit der Simulation beginnen.

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, stellen Sie die `desiredAnzahl` der Recheninstanzen auf einen Wert größer als 1. Es gibt eine maximale Anzahl von Apps für jeden Mitarbeiter. Weitere Informationen finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#). SimSpace Weaver verwendet nur mehr als einen Worker, wenn die Anzahl der Apps auf einem Worker dieses Limit überschreitet. SimSpace Weaver kann eine App auf jedem der verfügbaren Worker platzieren. Die Platzierung der App bei einem bestimmten Mitarbeiter ist nicht garantiert.

Der folgende Schemaausschnitt zeigt eine Konfiguration für eine Simulation, die 2 Worker anfordert. SimSpace Weaver wird versuchen, 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

Das `clock`-Abschnitt spezifiziert die Eigenschaften der Simulationsuhr. Derzeit können Sie nur das konfigurieren `tick-rate` (die Anzahl der Ticks pro Sekunde, die die Uhr an Apps sendet). Die `tick-rate` 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 `tick-rate` wird auch als `tick-rate` bezeichnet.

Die gültigen Werte für `tick_rate` hängen ab von `sdk_version` in Ihrem Schema angegeben.

Gültige Werte für die Tick-Rate

- Versionen vor "1.14":
 - 10

- 15
- 30
- Ausführung "1.14" oder später:
 - "10"
 - "15"
 - "30"
 - "unlimited"

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

Important

- Für Schemas mit `sdk_version` früher als "1.14" der Wert von `tick_rate` ist ein Ganzzahl, wie 30.
- Für Schemas mit `sdk_version` von "1.14" oder später, der Wert von `tick_rate` ist ein Schnur, wie "30". Der Wert muss die doppelten Anführungszeichen enthalten.

Wenn Sie eine Version konvertieren "1.12" oder "1.13" von Schema zu Version "1.14" oder später müssen Sie den Wert von `tick_rate` ein doppelten Anführungszeichen.

Unbegrenzte Tickrate

Sie können die Einstellung `tick_rate` zu "unlimited" damit Ihre Simulation so schnell ausgeführt werden kann, wie Ihr Code ausgeführt werden kann. Mit einer unbegrenzten Tickrate SimSpace Weaversendet den nächsten Tick sofort, nachdem alle Apps die Commits für den aktuellen Tick abgeschlossen haben.

Important

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

Unbegrenzte Tickrate 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 AWS Cloud. Sie spezifizieren immer noch `tick_rate: "unlimited"` in deinem Schema. Mehr über SimSpace Weaver Local erfahren Sie unter [Lokale Entwicklung](#).

Häufig gestellte Fragen zur Uhr

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

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

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 von `DescribeSimulationAPI`. 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.

- Dies `dk_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]}"]"
```

- Dies `dk_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\"}]"
```

- Dersdk_versionist"1.14"oder später und der Wert vontick_rateist 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]\"}]"
```

- Dersdk_versionist"1.14"oder später und der Wert vontick_rateist 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

Derpartitioning_strategiesIn diesem Abschnitt werden die Konfigurationseigenschaften für die Partitionen von Geo-Apps spezifiziert. Sie geben Ihren eigenen Namen für eine Partitionierungsstrategie an (eine Reihe von Eigenschaften in diesem Abschnitt) und verwenden ihn in Ihrer räumlichen App-Konfiguration.

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

DerpropertyDie Eigenschaft gibt den Typ des Koordinatensystems an, das Ihre Simulation verwendet. Der WertGridgibt ein zweidimensionales (2D) Gitter an.

Für ein `GridTopology`, der Simulationsraum wird als achsenorientierter Begrenzungsrahmen modelliert (AABB). Sie geben die Koordinatengrenzen für jede Achse Ihres AABB in `der_aabb_bounds` Eigentum. Alle Entitäten, die in Ihrer Simulation räumlich existieren, müssen eine Position innerhalb des AABB haben.

Gruppen zur Rasterplatzierung

Ein Platzierungsgruppe ist eine Sammlung von räumlichen App-Partitionen, die Sie möchten SimSpace Weaver auf denselben Arbeiter legen. Sie geben die Anzahl und Anordnung der Platzierungsgruppen (in einem Raster) in `der_grid_placement_groups` Eigentum. SimSpace Weaver wird versuchen, die Partitionen gleichmäßig auf die Platzierungsgruppen zu verteilen. Die Besitzbereiche räumlicher Apps mit Partitionen in derselben Platzierungsgruppe werden räumlich benachbart sein.

Wir empfehlen, dass $x * y$ der gewünschten Anzahl von Arbeitern entspricht. Wenn es nicht gleich ist, SimSpace Weaver wird versuchen, Ihre Praktikumsgruppen auf die verfügbaren Mitarbeiter auszurichten.

Wenn Sie keine Platzierungsgruppenkonfiguration angeben, SimSpace Weaver wird einen für Sie berechnen.

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`**— räumlicher Bereich
- **`launch_apps_per_worker`** (nicht in der Beispielanwendung enthalten) — Servicedomäne

Important

SimSpace Weaver unterstützt bis zu 5 Domänen für jede Simulation. Dies schließt alle räumlichen, benutzerdefinierten und dienstspezifischen Domänen ein.

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

- Weber-*lowercase-project-name-Kontonummer*-App-Reißverschlüsse-*Region*
- Weber-*lowercase-project-name-Kontonummer*-Schemata-*Region*

Themen

- [Konfiguration der App](#)
- [Konfiguration räumlicher Domänen](#)
- [Netzwerkendpunkte](#)
- [Konfiguration von Servicedomänen](#)

Konfiguration der App

Sie geben die Konfiguration einer App an (`app_config`) als Teil der Konfiguration für seine Domain. Alle Arten von Domains 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 WeaverApp SDK-Projekte der Version 1.12.x verwenden separate Buckets für die App-ZIP-Dateien und das Schema:

- `Weber-lowercase-project-name-Kontonummer`-App-Reißverschlüsse-*Region*
- `Weber-lowercase-project-name-Kontonummer`-Schemata-*Region*

Der `package`-Eigenschaft gibt den S3-URI einer Zip-Datei in einem S3-Bucket an. Die Zip-Datei enthält die ausführbare Datei der App (auch als Binär) und alle anderen Ressourcen, die es benötigt (wie Bibliotheken). Jede Instanz der ausführbaren App läuft in einem Docker-Container auf einem Arbeiter.

Der `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`
- Spezifizieren Sie: `launch_command: ["MyTestApp", "-option1", "value1"]`

Der `required_resource_units`-Eigenschaft gibt die Anzahl der Rechenressourceneinheiten an, die SimSpace Weaver dieser App zuordnen. Eine Rechenressourceneinheit ist eine feste Menge an Verarbeitungskapazität (vCPU) und Erinnerung (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. Auf jedem Worker gibt es eine begrenzte Anzahl von Rechenressourceneinheiten. Weitere Informationen finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Konfiguration räumlicher 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 App-ZIP-Dateien und das Schema:

- `Weber-lowercase-project-name-Kontonummer-App-Reißverschlüsse-Region`
- `Weber-lowercase-project-name-Kontonummer-Schemata-Region`

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

SimSpace Weaver startet 1 Instanz der räumlichen App für jede Zelle im Partitionsraster. Zum Beispiel, wenn eine räumliche Domäne hat `grid_partition` Wert `x: 2` und `y: 2`, es gibt $2 * 2 = 4$ Partitionen in der räumlichen Domäne. SimSpace Weaver startet 4 Instanzen der in der räumlichen Domäne konfigurierten App und weist jeder App-Instanz 1 Partition zu.

Themen

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

Ressourcenbedarf für räumliche Bereiche

Sie können jedem Worker bis zu 17 Rechenressourceneinheiten zuweisen. Sie geben die Anzahl der Rechenressourceneinheiten an, die jede räumliche App verwendet, in der `app_config` Abschnitt Ihres räumlichen Bereichs.

Example Schemaausschnitt, der die Rechenressourceneinheiten für eine räumliche 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 Raster (in Ihrem `grid_partition,x*y`) anhand der Anzahl der Rechenressourceneinheiten, die den räumlichen Apps zugewiesen sind.

Für das vorherige Beispiel ist die Domain `MySpatialDomain` spezifiziert:

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

Das Grid für `MySpatialDomain` hat $2 * 2 = 4$ Zellen. Die räumliche Domäne erfordert $4 * 1 = 4$ Rechenressourceneinheiten.

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

Mehrere räumliche Domänen

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

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 für jeden Entitätstyp verschiedene Domänen erstellen 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. Mehr über SimSpace Weaver Local erfahren Sie unter [Lokale Entwicklung](#).

Important

SimSpace Weaver unterstützt bis zu 5 Domänen für jede Simulation. Dies schließt alle räumlichen, benutzerdefinierten und dienstspezifischen Domänen ein.

Konfiguration mehrerer räumlicher Domänen

Um mehr als eine räumliche Domäne zu konfigurieren, fügen Sie die anderen räumlichen Domänendefinitionen als separate benannte Abschnitte in Ihr Schema ein. Jede Domain muss

angeben `launch_apps_by_partitioning_strategy` Schlüssel. 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 den Schlüssel der obersten Ebene hinzu `placement_constraints` zu Ihrem Schema, um anzugeben, welche Domänen `SimSpace Weavers` sollte zusammen platziert werden. Das Erforderliche `on_workers` Der Schlüssel muss sich auf einen benannten `workers` Konfiguration im Schema.

Example Schemaausschnitt, der zusammengelegte 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 Arbeiter ist.
 - Stellen Sie sicher, dass die Werte der Platzierungsgruppe gemeinsame Teiler für die Rasterabmessungen der Domänen sind, die Sie zusammen platzieren.
- Wenn nicht benutzen Platzierungsgruppen:
 - Stellen Sie sicher, dass eine Achse Ihrer räumlichen Domänenraster einen gemeinsamen Divisor 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 bestehenden Simulation einen weiteren räumlichen Bereich hinzufügen?

- Für eine laufende Simulation— Sie können die Konfiguration für eine laufende Simulation nicht ändern. Ändern Sie die Domainkonfiguration im Schema, laden Sie das Schema und die App-Zips hoch und starten Sie eine neue Simulation.
- Für eine neue Simulation— Fügen Sie die Domainkonfiguration zum Schema hinzu, laden Sie das Schema und die App-Zips 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 Domainkonfiguration 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  
  [\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, nach der Domänen zusammen für Arbeitnehmer vergeben werden. Dies passiert, wenn Sie mehrere räumliche Domänen angeben, es aber keinen gemeinsamen Divisor oder Vielfaches zwischen Domänenrastern gibt (z. B. zwischen einem 2x4-Raster und einem 3x5-Raster).

Netzwerkendpunkte

Benutzerdefinierte Apps und Service-Apps können Netzwerkendpunkte haben, mit denen externe Clients eine Verbindung herstellen können. Sie geben eine Liste von Portnummern als Wert für `ingress_ports` in `derendpoint_config`. Diese Portnummern sind sowohl TCP als auch UDP. Die benutzerdefinierte App oder die Service-App sollte an die Portnummern gebunden sein, die Sie in `angebeningress_ports`. SimSpace Weaver weist zur Laufzeit dynamisch Portnummern zu und ordnet diese Ports den dynamischen Ports zu. Du kannst die `anrufendescrbe-appAPI`,

nachdem Ihre Apps begonnen haben, die dynamischen (tatsächlichen) Portnummern zu finden. Weitere Informationen finden Sie unter [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#) aus dem 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 WeaverApp SDK-Projekte der Version 1.12.x verwenden separate Buckets für die App-ZIP-Dateien und das Schema:

- Weber-*lowercase-project-name-Kontonummer*-App-Reißverschlüsse-*Region*
- Weber-*lowercase-project-name-Kontonummer*-Schemata-*Region*

Note

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

Konfiguration von Servicedomänen

Das Vorhandensein von `launch_apps_per_worker` in einer Domänenkonfiguration bedeutet dies, dass es sich um eine Dienstdomäne mit Service-Apps handelt. SimSpace Weaver startet und stoppt Service-Apps für Sie. Wann SimSpace Weaver startet und stoppt eine App, es wird davon ausgegangen, dass die App ein verwalteter Lebenszyklus. SimSpace Weaver unterstützt derzeit das Starten von 1 oder 2 Service-Apps auf jedem einzelnen Mitarbeiter.

Example Beispiel für eine Domain, die so konfiguriert ist, dass auf jedem Worker eine Service-App gestartet wird

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://example-bucket/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 auf jedem Worker 2 Service-Apps gestartet werden

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 2
    app_config:
      package: "s3://example-bucket/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 die maximale Dauer angibt, für die die Simulation 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` oder `1H` bedeutet 1 Stunde. SimSpace Weaver hört auf, Ihre Simulation zu erstellen, wenn sie diese Grenze 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 äquivalent zu `0` ist. 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` Zustand über, sobald sie den `STARTED` Status erreicht.

Starten einer Simulation mithilfe von SimSpace Weaver App-SDK-Skripten

Sie können einen Wert für den `maximum-duration` Parameter angeben, wenn Sie eines der folgenden Skripts verwenden, um eine Simulation zu starten:

- `quick-start-project-name-cli.bat --maximum-duration value`
- `start-simulation-project-name.bat --maximum-duration value`
- `run-project-name.bat --maximum-duration value`

Jedes Skript übergibt den Wert von `maximum-duration` an die `StartSimulation` API.

Important

Wenn Sie keinen Wert für `maximum-duration` angeben, wird der [Standardwert](#) (`14D`) verwendet.

Starten einer Simulation mit der Konsole

Sie können einen Wert für die maximale Dauer angeben, wenn Sie eine Simulation in der [SimSpace WeaverKonsole](#) starten. Geben Sie den Wert in das Feld Maximale Dauer des Formulars mit den 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, lautet der Status der Simulation STOPPING (sofern sie gerade läuft) oder STOPPED. In der [SimSpace WeaverKonsole](#) ist der Zielstatus der Simulation immer noch STARTED, da dies der letzte von einem Benutzer angeforderte Status war.

Entwicklung Apps

SimSpace Weaver Die Entwicklung erfordert eine Amazon Linux 2 (AL2) Umgebung zum Erstellen von Apps, da Ihre Simulationen Amazon Linux in der ausgeführt AWS Cloud werden. Wenn Sie das verwenden Windows, können Sie Skripts im SimSpace Weaver App-SDK verwenden, um einen Docker Container zu erstellen und zu starten, der AL2 mit den Abhängigkeiten ausgeführt wird, die Sie zum Erstellen von SimSpace Weaver Apps benötigen. Sie können eine AL2 Umgebung auch mit Windows Subsystem for Linux (WSL) einem nativen AL2 System starten oder verwenden. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

Note

Unabhängig davon, wie Sie Ihre lokale Entwicklungsumgebung konfigurieren, werden Ihre Apps in Docker Containern ausgeführt, wenn Sie sie zur Ausführung in der hochladen 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 den Loop, wenn die Simulation beendet wird.
 - b. Verarbeiten Sie Ereignisse im Abonnement und in Bezug auf Eigentümerschaft.
 - c. Aktualisieren Sie die Simulation.
 - d. Bestätigen Sie die `Transaction`, um das Update zu beenden.
3. Zerstören Sie die Anwendung.

Spatial Apps

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

Das SimSpace Weaver App-SDK bietet eine Beispielanwendung. Den Quellcode für die räumliche App der Beispielanwendung finden Sie im folgenden Ordner:

Docker

```
project-folder\src\PathfindingSample\SpatialApp
```

WSL

Important

Wir stellen diese Anweisungen zu Ihrer Unterstützung bereit. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
project-folder/src/PathfindingSample/SpatialApp
```

Benutzerdefinierte Apps

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

Benutzerdefinierte Apps können

- Erstellen von Entitäten
- Abonnieren von anderen Partitionen
- Commit-Änderungen

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 bestimmte 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 den Loop, wenn die Simulation beendet wird.
 - b. Änderungen des Prozesszustands.
 - 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 in der Simulation existieren kann. SimSpace Weaver verwendet den räumlichen Standort der Entität, um die Entität in der entsprechenden räumlichen App-Partition zu platzieren. Die benutzerdefinierte App, die die Entität erstellt hat, kann die Entität nicht aktualisieren oder löschen, nachdem sie in eine räumliche Domäne übertragen wurde.

Das SimSpace Weaver App-SDK bietet eine Beispielanwendung. Sie können die in der Beispielanwendung enthaltenen benutzerdefinierten Apps als Modelle für Ihre eigenen benutzerdefinierten Apps verwenden. Den Quellcode für die View-App (eine benutzerdefinierte App) der Beispielanwendung finden Sie im folgenden Ordner:

Docker

```
project-folder\src\PathfindingSample\ViewApp
```

WSL

Important

Wir stellen diese Anweisungen zu Ihrer Unterstützung bereit. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
project-folder/src/PathfindingSample/ViewApp
```

Entwicklung von Client-Anwendungen

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

- Injizieren Sie Verkehrsinformationen in Echtzeit in eine Simulation im Stadtmaßstab.
- Erstellen Sie human-in-the-loop-Simulationen, bei denen ein menschlicher Bediener einen Aspekt der Simulation steuert.
- Ermöglichen Sie es Benutzern, mit der Simulation zu interagieren, z. B. für eine Trainingssimulation.

Die benutzerdefinierten Apps in diesen Beispielen dienen als Schnittstelle zwischen dem Simulationsstatus und der Außenwelt. 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 zeigt nur eine IP-Adresse und eine Portnummer für jede Ihrer benutzerdefinierten Apps an, damit Clients eine Verbindung zu ihnen herstellen können.

Das SimSpace Weaver App-SDK stellt Clients für seine Beispielanwendung zur Verfügung. 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 werden zu Informationsmöglichkeiten zusammengefasst. Sie sind für die Verwendung mit Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

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

Weitere Informationen zum Erstellen und Verwenden der Beispielanwendungsclients finden Sie [Schritt 5: Sehen Sie sich Ihre Simulation an](#) im Schnellstart-Tutorial in diesem Handbuch.

Lokale Entwicklung

Sie können Ihre SimSpace Weaver Anwendungen lokal bereitstellen, um sie schnell zu testen und zu debuggen. SimSpace Weaver Local wird Microsoft Windows nur für das Aufbauen auf unterstützt.

Important

Hinweise zur Entwicklung mit Unity und Unreal Engine finden Sie unter [Unterstützung für andere Engines](#).

⚠ Important

Wenn Sie in Version 1.15.3 mit C++, Unreal Engine oder PythonUnity arbeiten, finden Sie unter [Lokale Entwicklungsunterschiede in Version 1.15.3](#)

Voraussetzungen

- Microsoft Windows 10 oder höher
- [Microsoft Visual Studio 2019](#) oder später, wenn der [Desktop development with C++ Workload](#) installiert ist

Themen

- [Erstellen Sie Ihre Simulation für SimSpace Weaver Local](#)
- [Führen Sie Ihre Simulation durch mit SimSpace Weaver Local](#)
- [Sehen Sie sich Ihre lokale Simulation an](#)
- [Stoppen Sie Ihre lokale Simulation](#)
- [Debuggen lokaler Simulationen](#)
- [Lokale Entwicklungsunterschiede in Version 1.15.3](#)

Erstellen Sie Ihre Simulation für SimSpace Weaver Local

Um die Bedienung zu erlernen SimSpace Weaver Local, können Sie dieselbe Pathfinding-Beispielanwendung verwenden, die Sie während der [Erste Schritte mit SimSpace Weaver](#) Tutorials in der Cloud ausgeführt haben, diesmal jedoch auf Ihrer lokalen Hardware.

Um die Beispielanwendung für zu erstellen SimSpace Weaver Local

1. Gehen Sie in einer Befehlszeile zu `project-folder\tools\local`.
2. Führen Sie `generate_visual_studio_project.bat`.
3. `project-folder\buildlocal\PathfindingSampleLocal.sln` in Visual Studio öffnen.
4. Stellen Sie Ihre Build-Konfiguration auf ein RelWithDebInfo.
5. Wählen Sie Build > Build Solution.

Visual Studio platziert deine Build-Artefakte in:

- *project-folder*\buildlocal\out\RelWithDebInfo.

In diesem Ordner sollten Sie die folgenden ausführbaren Dateien sehen:

- PathfindingSampleLocalSpatial.exe
- PathfindingSampleLocalView.exe

Führen Sie Ihre Simulation durch mit SimSpace Weaver Local

Sie können SimSpace Weaver Local damit eine beliebige Kombination von bis zu 24 räumlichen oder benutzerdefinierten Apps auf Ihrem lokalen Computer ausführen. Die Simulationsuhr startet, nachdem alle im Schema definierten räumlichen Apps gestartet wurden.

Um Ihre Apps auszuführen mit SimSpace Weaver Local

1. Gehen Sie in einem Dateiauswahlfenster zu *project-folder*\buildlocal\out\RelWithDebInfo.
2. SimSpace Weaver Local Apps benötigen eine Schemadatei, die `schema.yaml` im Arbeitsverzeichnis der Apps benannt ist. Wenn die erforderlichen Informationen aus dem Schema nicht gelesen werden können, werden die Apps beendet.


Das Schema für SimSpace Weaver Local muss nicht identisch mit sein *project-folder*\tools*project-name*-schema.yaml, aber Sie können es als Ausgangspunkt verwenden.

Wählen Sie eine der folgenden Optionen:

- Kopieren Sie das Schema nach *project-folder*\buildlocal\out\RelWithDebInfo\schema.yaml.
- Setzen Sie die Umgebungsvariable `WEAVERLOCAL_SCHEMA_PATH` auf den Namen einer Schemadatei mit einem anderen Pfad oder Dateinamen.

Example Beispiel

```
set WEAVERLOCAL_SCHEMA_PATH=c:\projects\MyProject\tools\MyProject-schema.yaml
```

 Note

Wenn Sie Ihre Umgebungsvariable von der Befehlszeile aus festlegen, kann auf diese Umgebungsvariable (mit diesem Wert) nur in dieser Befehlszeile (Konsolenfenster) zugegriffen werden.

3. Das Schema für die Beispielanwendung definiert ein 2x2-Raster, das 4 Partitionen erstellt. Sie werden ein Skript ausführen, das 4 Instanzen der Geo-App startet, die der im Schema angegebenen Anzahl von Geo-Apps entsprechen. Das Skript startet außerdem eine View-App. Nachdem alle Geo-Apps gestartet wurden und ihnen eine Partition zugewiesen wurde, beginnt die Simulation automatisch zu ticken.


Um Ihre Apps zu starten

- a. Wechseln Sie in einer Befehlszeile zum lokalen Tools-Ordner für Ihr Projekt.


```
cd project-folder\tools\local
```

- b. Führen Sie das Skript aus, um die Apps zu starten.

```
launch_simulation_locally.bat
```

 Note

Wenn Sie `WEAVERLOCAL_SCHEMA_PATH` den Namen einer Schemadatei angeben, müssen Sie Ihre Spatial-Apps über die Befehlszeile in derselben Sitzung (Fenster) starten, in der Sie die Umgebungsvariable festgelegt haben.

 Important

Wenn Sie ein Windows-Sicherheits-Popup erhalten, wählen Sie diese Option, `Allow Access` damit Sie eine Verbindung zur View-App herstellen können, um die Simulation zu visualisieren.

Note

Sie können Ihre Spatial- und View-Apps auch manuell starten. Dazu müssen Sie 4 Instanzen der Spatial-App und eine View-App manuell starten.

- Räumliche App: `start PathfindingSampleLocalSpatial.exe`
- App ansehen: `start PathfindingSampleLocalView.exe`

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 [Schritt 5: Sehen Sie sich Ihre Simulation an](#) im Schnellstart-Tutorial.

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

Stoppen Sie Ihre lokale Simulation

Ihre lokale Simulation wird weiterhin ausgeführt, wenn Ihre lokalen Geo-Apps aktiv sind. Wenn Sie eines der Spatial-App-Fenster schließen, beenden Sie die gesamte Simulation. Schließen Sie alle anderen Fenster, um den Rest der Simulation zu bereinigen.

Sie können jedes App-Fenster manuell schließen, oder Sie können das folgende Skript verwenden, um alle Fenster automatisch zu schließen:

- `project-folder\tools\local\terminate_local_simulation.bat`

Note

Auch wenn das Schließen eines räumlichen App-Fensters die Simulation beendet, sollten Sie darauf achten, die anderen App-Fenster zu schließen. Sie können keine weitere lokale Simulation erfolgreich starten, wenn noch Fenster aus einer früheren Simulation geöffnet sind.

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.

Lokale Entwicklungsunterschiede in Version 1.15.3

In diesem Abschnitt werden Änderungen in der Entwicklung SimSpace Weaver Local ab Version 1.15.3 beschrieben. Diese Änderungen wirken sich auf Workflows für SimSpace Weaver Local Projekte in C++, PythonUnity, und aus. Unreal Engine

Themen

- [Änderungen an Dateien](#)
- [Aktualisieren Sie ein vorhandenes C++-Projekt auf SimSpace Weaver Local 1.15.3](#)
- [Führen Sie ein neues Python-Projekt in SimSpace Weaver Local 1.15.3 aus](#)
- [Aktualisieren Sie ein vorhandenes Unity-Projekt auf SimSpace Weaver Local 1.15.3](#)
- [Aktualisieren Sie ein vorhandenes Unreal Engine-Projekt auf 1.15.3 SimSpace Weaver Local](#)
- [Häufig gestellte Fragen zu 1.15.3 SimSpace Weaver Local](#)

Änderungen an Dateien

- SimSpaceWeaverAppSdkLocalBibliotheksdateien sind jetzt benannt `weaver_app_sdk_cxx_v1_full_local`.
 - Sie finden diese Dateien in `sdk-folder\SimSpaceWeaverAppSdk-1.15.3\lib\weaverlocal\windows`.
- Ihre Projekte sollten nicht mehr auf diese verweisen oder diese enthalten `SimSpaceWeaverAppSDK-1.15.1\include\aws\weaverruntime\local_ffl`.
- Die Python-Skripte für benötigen nicht SimSpace Weaver Local mehr `cmake`.
- Es gibt neue oder umbenannte Python-Skripte für SimSpace Weaver Local:
 - `build-local`— Erstellt die Python-App für den Start. Es legt Ihren gesamten Python-Code und die SimSpace Weaver Local Bibliotheksdateien in das `buildlocal` Verzeichnis.
 - `local-config`— Definiert die Umgebungsvariablen für die lokalen Skripte.
 - `start-python-locally`— Richtet die Apps ein PythonPath und startet Python.
 - Die folgenden Skripte entsprechen den Cloud-Skripten mit ähnlichen Namen:
 - `quick-start-local`

- `start-simulation-local`
- `stop-simulation-local`

Aktualisieren Sie ein vorhandenes C++-Projekt auf SimSpace Weaver Local 1.15.3

Voraussetzungen

- [App-SDK der Version 1.15.3 SimSpace Weaver](#)
- Ein vorhandenes C++-Projekt für SimSpace Weaver Local Version 1.15.2 oder früher

Um Ihr Projekt zu aktualisieren

1. Stellen Sie sicher, dass Ihr Projekt den erforderlichen Header enthält:

```
#include <aws/weaverruntime/ffi/weaver_app_sdk_cxx_ffi_v1/src/lib.rs.h>
```

Note

Dies ist normalerweise enthalten in `aws\weaverruntime\detail.h`

2. Stellen Sie sicher, dass Ihr Projekt den folgenden veralteten Header nicht enthält:

```
#include <aws/weaverruntime/local_ffi/Bridge.h>
```

Note

Dieser ist normalerweise enthalten in `aws\weaverruntime\detail.h`

3. Ersetzen Sie in Ihrer CMake-Datei oder Ihrem Build-Skript die `SimSpaceWeaverAppSdkLocal` statischen Bibliotheksnamen durch den neuen `weaver_app_sdk_cxx_v1_full_local` Namen.
4. Wenn Sie Version 1.15.3 noch nicht ausgeführt haben, führen Sie sie jetzt `sdk-folder\docker-create-image.bat` aus. Sie müssen dies nur einmal tun.
5. Folgen Sie den [üblichen Verfahren](#), um Ihr Projekt zu erstellen und auszuführen.

Fehlerbehebung

Sie erhalten Linker-Fehler (ungelöstes externes Symbol)

Ihr Compiler gibt Linker-Fehler wie die folgenden aus (aus Gründen der Lesbarkeit wurden Zeilenumbrüche hinzugefügt).

```
Error    LNK2019    unresolved external symbol
           "class outcome_v2_92ee5284::basic_result<class
           Aws::WeaverRuntime::Application,
           enum Aws::WeaverRuntime::ffi::weaver_app_sdk_cxx_ffi_v1::ErrorCode
```

Um das Problem zu beheben

- Stellen Sie sicher, dass Ihr Projekt nicht beinhaltet `<aws/weaverruntime/local_ffi/Bridge.h>`.

Sie erhalten einen Typinkompatibilitätsfehler

Möglicherweise wird ein Fehler wie der folgende angezeigt (aus Gründen der Lesbarkeit wurden Zeilenumbrüche hinzugefügt).

```
the object has type qualifiers that are not compatible with the member function
object type is: const rust::cxxbridge1::String
```

Um das Problem zu beheben

1. Aktualisieren Sie Ihre SimSpace Weaver Header-Dateien mit den Versionen 1.15.3.
2. Stellen Sie sicher, dass jede Verwendung von der folgenden `rust::cxxbridge1::String` ähnelt:

```
rust::cxxbridge1::String domain_name = domain.name.value;
if (domain.type_ == Api::DomainType::Spatial && Name.Compare(domain_name.c_str())
    == 0)
```

Führen Sie ein neues Python-Projekt in SimSpace Weaver Local 1.15.3 aus

Voraussetzungen

- [App-SDK der Version 1.15.3 SimSpace Weaver](#)
- Ein neues Projekt der SimSpace Weaver Local Version 1.15.3 PythonBubblesSample

Um dein Projekt auszuführen

1. Gehen Sie in einem Befehlszeilenfenster zu `project-folder\tools\local\windows`.
2. Führen Sie Folgendes aus: `quick-start-local.bat`

4 Spatial-Apps, 1 View-App und 1 Client sollten lokal gestartet werden. Der Client sollte Blasen anzeigen.

Fehlerbehebung

Sie erhalten eine Fehlermeldung, die besagt, dass sie **GLIBCXX** nicht gefunden wurde

Der folgende Fehler weist darauf hin, dass das SimSpace Weaver Python-App-SDK nicht importiert werden konnte. Eine wahrscheinliche Ursache ist, dass Ihre C++-Bibliotheken veraltet sind.

```
ImportError: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.29' not found
```

Um das Problem zu beheben

1. Aktualisieren Sie die Quelle Ihrer `libstdc++` Bibliotheken (wie `gcc` oder `msvc`) auf eine Version, die die angegebene Version von enthält `GLIBCXX` (im angegebenen Beispiel ist die Version `3.4.29`).
2. Stellen Sie sicher, dass die Systemumgebungsvariable auf den richtigen Pfad zu gesetzt `LD_LIBRARY_PATH` ist. `lib64`

Das Schnellstart-Skript schlägt fehl

Das Schnellstartskript schlägt mit einer Meldung wie der folgenden fehl.

```
python: can't open file 'C:\usr\project\buildlocal\bIn\bubbles_tkinter_client.py':
```



```
[Errno 2] No such file or directory
```

Um das Problem zu beheben

1. Gehe zu *project-folder*\tools\local\windows und bearbeite `local-config.bat`.
2. Stellen Sie sicher, dass die folgenden Umgebungsvariablen auf die richtigen Pfade für Ihr lokales System gesetzt sind:
 - `TOOLS_DIR` ist auf dein gesetzt *project-folder*\tools
 - `TOOLS_DIR_WINDOWS` ist auf dein eingestellt *project-folder*\tools\local\windows
 - `PROJECT_ROOT` ist auf dein eingestellt *project-folder*
 - `BUILD_DIR` ist auf dein eingestellt *project-folder*\buildlocal
 - `APP_SDK_DIR` ist auf dein eingestellt *sdk-folder*

Aktualisieren Sie ein vorhandenes Unity-Projekt auf SimSpace Weaver Local 1.15.3

Voraussetzungen

- [App-SDK der Version 1.15.3 SimSpace Weaver](#)
- Ein vorhandenes Unity-Projekt für SimSpace Weaver Local Version 1.15.2 oder früher

Um Ihr Projekt zu aktualisieren

1. Entfernen Sie in Unity das vorhandene AWS SimSpace Weaver-Paket.
 - a. Öffnen Sie Ihr vorhandenes Unity-Projekt.
 - b. Wählen Sie im Editorfenster „Fenster“ > „Paketmanager“.
 - c. Wählen Sie unter Pakete — Unity Technologies die Option AWS SimSpace Weaver und dann Entfernen aus.
2. Führen Sie in Ihrer Version 1.15.3 Folgendes *sdk-folder* aus `download-unity-package.bat`
3. Folgen Sie den Anweisungen unter `Unity_SDK_for_AWS_SimSpace_Weaver.pdf`, um das neu heruntergeladene `SimSpaceWeaverUnityPackage.zip` AWS SimSpace Weaver-Paket im Unity-Editor hinzuzufügen.

Aktualisieren Sie ein vorhandenes Unreal Engine-Projekt auf 1.15.3 SimSpace Weaver Local

Voraussetzungen

- [App-SDK der Version 1.15.3 SimSpace Weaver](#)
- Ein vorhandenes Unreal Engine-Projekt für SimSpace Weaver Local Version 1.15.2 oder früher

Um dein Projekt zu aktualisieren

1. Schließen Sie alle Unreal-Projekt- und Code-Editor-Fenster.
2. Gehen Sie in einem Befehlszeilenfenster zu Ihrer Version *sdk-folder* 1.15.3.
3. Führen Sie `update-unreal-project.bat --path project-folder --name project-name`.

Note

Dadurch wird Ihr vorhandenes Plugin durch das neue Plugin ersetzt. Alle Änderungen werden gelöscht.

4. Folgen Sie den Anweisungen unter `AWS_SimSpace_Weaver_Unreal_Guide.pdf`, um Ihr Projekt für SimSpace Weaver Local zu erstellen.

Fehlerbehebung

Fehler beim Löschen vorhandener Plugin-Dateien

Möglicherweise erhalten Sie ähnliche Fehler wie die folgenden.

```
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverAppSdk.dll': Operation not permitted
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverAppSdkLocal.dll': Operation not permitted
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverCppMetrics.dll': Operation not permitted
```

Um das Problem zu beheben

1. Stellen Sie sicher, dass das Fenster des Unreal-Projekteditors und die Code-Editoren geschlossen sind.
2. Führen Sie `update-unreal-project.bat`.

Häufig gestellte Fragen zu 1.15.3 SimSpace Weaver Local

- F1: Wie ändere ich meine SimSpace Weaver Local Umgebungsvariablen für mein Python-Projekt?
 - Bearbeiten Sie `project-folder\tools\windows\local-config` für Ihr Python-Projekt.

AWS SimSpace Weaver -App-SDK

Das SimSpace Weaver App-SDK stellt APIs bereit, mit denen Sie die Entitäten in Ihrer Simulation steuern und auf SimSpace Weaver Ereignisse reagieren können. Sie enthält den folgenden Namespace:

- API – Kerndefinitionen der API und ihrer Verwendung

Verknüpfen Sie mit der folgenden Bibliothek:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

Die Bibliothek ist für dynamische Links verfügbar, wenn Sie Ihre Apps in der `ausführenAWS` Cloud. Sie müssen es nicht mit Ihren Apps hochladen.

Note

Die SimSpace Weaver App-SDK-APIs steuern Daten in Ihrer Simulation. Diese APIs sind getrennt von den SimSpace Weaver Service-APIs, die Ihre SimSpace Weaver Servicere Ressourcen (wie Simulationen, Apps und Uhren) in `steuernAWS`. Weitere Informationen finden Sie unter [SimSpace Weaver API-Referenzen](#).

Themen

- [API-Methoden geben einen zurück Result](#)
- [Interaktion mit dem App-SDK auf oberster Ebene](#)
- [Simulationsmanagement](#)
- [Subscriptions \(Abonnements\)](#)
- [Entitäten](#)
- [Entitätsereignisse](#)
- [Result und Fehlerbehandlung](#)
- [Generika und Domänentypen](#)
- [Verschiedene Anwendungs-SDK-Operationen](#)

API-Methoden geben einen zurück Result

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

Example Beispiel

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

Diese Methode:

- Gibt zurück `Transaction`, wenn erfolgreich `BeginUpdate()` ausgeführt wird.
- Gibt zurück `Aws::WeaverRuntime::ErrorCode`, wenn `BeginUpdate()` fehlschlägt.

Interaktion mit dem App-SDK auf oberster Ebene

Lebenszyklus

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

Partitionen

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

Simulationsmanagement

In diesem Abschnitt werden Lösungen für gängige Simulationsverwaltungsaufgaben beschrieben.

Themen

- [Starten einer Simulation](#)
- [Aktualisieren einer Simulation](#)
- [Beenden einer Simulation](#)

Starten einer 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 einer Simulation

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

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)` – teilt Ihnen mit, ob blockieren `BeginUpdate()` wird oder nicht.

Verwenden Sie `Result<void> Commit(Transaction& txn)`, um die Änderungen zu bestätigen.

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();
}
```

Beenden einer Simulation

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

Andere Apps stellen fest, dass die Simulation heruntergefahren wird, wenn sie `ErrorCode::ShuttingDown` von ihren Aufrufen an `BeginUpdateWillBlock()` oder `erhaltenBeginUpdate()`. Wenn eine App empfängt `ErrorCode::ShuttingDown`, kann sie aufrufen, 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)` nach `aufApi::Commit()`. Das Zerstören einer Anwendung während einer Aktualisierung kann zu undefiniertem Verhalten führen.

⚠ Important

Sie müssen aufrufen, `DestroyApplication()` bevor das Programm beendet wird, um sicherzustellen, dass die Anwendung als erfolgreich beendet gemeldet wird.

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

Subscriptions (Abonnements)

Sie erstellen ein Abonnement mit einem Abonnementbereich und einer Domain-ID. Die Domänen-ID stellt die Domäne dar, die Eigentümer dieses Abonnementbereichs ist. Eine `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 die von `Api::SubscriptionHandle` zurückgegebene `transaction` verwenden `CreateSubscriptionBoundingBox2F32()`, um das Abonnement zu ändern. Sie übergeben sie als Argument an die folgenden Funktionen:

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

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

Entitäten

Sie rufen die `-Store` und `-Load` APIs über die `Api::Entity` des von `Result<Api::Entity>` zurückgegebenen oder über ein Eigentumsänderungsereignis auf `CreateEntity()`, wenn eine Entität den Abonnementbereich der App betritt (weitere Informationen finden Sie unter [Entitätsereignisse](#)). Wir empfehlen Ihnen, Ihre `Api::Entity` Objekte zu verfolgen, damit Sie sie mit diesen APIs verwenden können.

Themen

- [Erstellen von Entitäten](#)
- [Übertragen einer Entität in eine räumliche Domäne](#)
- [Schreiben und Lesen von Entitätsfelddaten](#)
- [Speichern der Position einer Entität](#)
- [Laden der Position einer Entität](#)

Erstellen von Entitäten

Verwenden Sie `CreateEntity()`, um eine Entität zu erstellen. Sie definieren die Bedeutung der `Api::TypeId`, die 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.

Übertragen einer Entität in eine räumliche Domäne

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

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

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

Parameter

txn

Die aktuelle Transaction.

entity

Das Ziel Entity für die Änderung von Domain.

domainId

Die DomainId des Ziels Domain für die Entity.

Diese API gibt zurückSuccess, wenn die Entitätsdomäne erfolgreich geändert wurde.

Schreiben und Lesen von Entitätsfelddaten

Alle Entitätsdatenfelder sind Blob-Typen. Sie können bis zu 1 024 Byte an Daten in eine Entität schreiben. Wir empfehlen Ihnen, Blobs so klein wie möglich zu halten, da größere Größen die Leistung verringern. Wenn Sie in einen Blob schreiben, übergeben Sie SimSpace Weaver einen Zeiger auf die Daten und ihre Länge. Wenn Sie aus einem Blob lesen, SimSpace Weaver stellt Ihnen einen Zeiger und eine Länge zum Lesen bereit. Alle Lesevorgänge müssen abgeschlossen sein, bevor die App aufruftCommit(). Von einem Leseaufruf zurückgegebene Zeiger werden ungültig, wenn die App aufruftCommit().

Important

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

Themen

- [Speichern der Felddaten einer Entität](#)
- [Laden der Felddaten einer Entität](#)
- [Laden der Felddaten entfernter Entitäten](#)

Speichern der Felddaten einer Entität

Die folgenden Beispiele zeigen, wie Sie die Felddaten einer Entität, die der App gehört, speichern (in den Zustandstext schreiben) können. 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 stellt den Datentyp der übergebenen Daten dar.

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

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 für die Verwendung eines struct zum Speichern der Daten

```

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));
}

```

Laden der Felddaten einer Entität

Die folgenden Beispiele zeigen, wie Sie die Felddaten einer Entität laden (aus dem Zustandsmaterial lesen). 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 `Api::TypeId` von `erhaltenApi::BuiltinTypeId`. Wenn es keine angemessene Konvertierung gibt, können Sie verwenden `Api::BuiltinTypeId::Dynamic`.

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 für die Verwendung eines struct zum Speichern der Daten

```

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);
}

```

Laden der Felddaten entfernter Entitäten

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

Example Beispiel für einen 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;
}
}
}

/* ... */
}

```

Speichern der Position einer Entität

Sie können die Position einer Entität mithilfe einer Ganzzahldatenstruktur speichern (in den Zustandstext 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 müssen `Api::BuiltinTypeId::Vector3F32` 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 für die Verwendung eines struct zur Darstellung der Position

```
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));
}

```

Laden der Position einer Entität

Sie können die Position einer Entität mithilfe einer Ganzzahldatenstruktur laden (aus dem Zustandsmaterial lesen). In diesen Beispielen wird die folgende Funktion verwendet:

Note

Sie müssen `Api::BuiltinTypeId::Vector3F32` 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 für die Verwendung eines struct zur Darstellung der Position

```

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);
}
```

Entitätsereignisse

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 Callback-gesteuerte Ereignisverarbeitung für Entitäten 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 Entitätsereignisverarbeitung erstellen.

Themen

- [Durch Ereignisse für -eigene Entitäten iterieren](#)

- [Iterieren von Ereignissen für abonnierte Entitäten](#)
- [Iterieren Sie durch Ereignisse zur Änderung der Eigentümerschaft für Entitäten](#)

Durch Ereignisse für -eigene Entitäten iterieren

Verwenden Sie `OwnershipChanges()`, um eine Liste der Ereignisse für -eigene Entitäten (Entitäten im Besitzbereich der App) abzurufen. Die Funktion hat die folgende Signatur:

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

Führen Sie dann die Entitäten mit einer Schleife durch, 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 im 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 im Bereich und wurde geändert.
- Reject – Die App konnte die Entität nicht aus dem Bereich entfernen.

Note

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

Iterieren von Ereignissen für abonnierte Entitäten

Verwenden Sie `AllSubscriptionEvents()`, um eine Liste der Ereignisse für abonnierte Entitäten (Entitäten im Abonnementbereich der App) abzurufen. Die Funktion hat die folgende Signatur:

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

Führen Sie dann die Entitäten mit einer Schleife durch, 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 im 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 im Bereich und wurde geändert.
- Reject – Die App konnte die Entität nicht aus dem Bereich entfernen.

Note

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

Iterieren Sie durch Ereignisse zur Änderung der Eigentümerschaft für Entitäten

Um Ereignisse zu erhalten, bei denen eine Entität zwischen einem Eigentumsbereich und einem Abonnementbereich wechselt, vergleichen Sie die Änderungen zwischen dem aktuellen und dem vorherigen Entitätseigentum und Abonnementereignissen.

Sie können diese Ereignisse behandeln, indem Sie Folgendes lesen:

- `Api::SubscriptionChangeList`

- `Api::OwnershipEvents`

Anschließend können Sie die Änderungen mit zuvor gespeicherten Daten vergleichen.

Das folgende Beispiel zeigt, wie Sie Ereignisse zur Änderung der Entitätseigentümerschaft behandeln können. In diesem Beispiel wird davon ausgegangen, dass für Entitäten, die zwischen abonnierten Entitäten und -eigenen Entitäten (in beide Richtungen) wechseln, das Ereignis zum Entfernen/Hinzufügen des Eigentums zuerst eintritt, gefolgt vom Ereignis zum Entfernen/Hinzufügen des Abonnements im nächsten Häkchen.

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 entityIdByAction;

/**
 * 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:
            {
                entityIdByAction[action].push_back(
                    event.entity.descriptor->id);
                break;
            }
        case Api::ChangeListAction::None:
        case Api::ChangeListAction::Update:
        case Api::ChangeListAction::Reject:
            {
                break;
            }
    }
}

```

```
}

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

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

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

```

    }
}

m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;

return Success();
}

```

Result und Fehlerbehandlung

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

```

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 Makro der Steueranweisung

Innerhalb einer Funktion mit dem Rückgabebetyp können `Aws::WeaverRuntime::Result<T>` Sie das `WEAVERRUNTIME_TRY` Makro anstelle des vorherigen Codemusters verwenden. Das Makro führt die an es übergebene Funktion aus. Wenn die übergebene Funktion fehlschlägt, gibt das Makro

die umschließende Funktion einen Fehler zurück. Wenn die übergebene Funktion erfolgreich ist, fährt die Ausführung mit der nächsten Zeile fort. Das folgende Beispiel zeigt ein Umschreiben der vorherigen `DoBeginUpdate()` Funktion. Diese Version verwendet das `WEAVERRUNTIME_TRY` Makro anstelle der `if-else` Kontrollstruktur. Beachten Sie, dass der Rückgabebetyp der Funktion `istAws::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 `BeginUpdate()` fehlschlägt, kehrt das Makro frühzeitig mit einem Fehler `DoBeginUpdate()` zurück. Sie können das `WEAVERRUNTIME_EXPECT_ERROR` Makro verwenden, um die `Aws::WeaverRuntime::ErrorCode` von `abzurufenBeginUpdate()` abzurufen. Das folgende Beispiel zeigt, wie die `Update()` Funktion aufruft `DoBeginUpdate()` und den Fehlercode bei einem Fehler abrufen.

```
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 für eine Funktion `BeginUpdate()` verfügbar machen, die aufruft, `Update()` indem Sie den Rückgabotyp von `Update()` in `ändernAws::WeaverRuntime::Result<void>`. Sie können diesen Vorgang wiederholen, um den Fehlercode weiter unten im Aufruf-Stack zu senden.

Generika und Domänentypen

Das SimSpace Weaver App-SDK stellt die Datentypen `Api::Vector2F32` und mit einfacher Genauigkeit `Api::BoundingBox2F32` sowie die Datentypen mit doppelter Genauigkeit `Api::Vector2F64` und `bereitApi::BoundingBox2F64`. Bei diesen Datentypen handelt es sich um passive Datenstrukturen ohne Bequemlichkeitsmethoden. 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 minimale Version der AzCoremathematischen Bibliothek, die `Vector3` und enthält `Aabb`. Weitere Informationen finden Sie in den Header-Dateien in:

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

Verschiedene Anwendungs-SDK-Operationen

Themen

- [AllSubscriptionEvents und OwnershipChanges enthalten Ereignisse aus dem letzten Aufruf](#)
- [Freigabe von Lesesperren nach der Verarbeitung SubscriptionChangeList](#)
- [Erstellen einer eigenständigen App-Instance zum Testen](#)

AllSubscriptionEvents und OwnershipChanges enthalten Ereignisse aus dem letzten Aufruf

Die Rückgabewerte von Aufrufen an `Api::AllSubscriptionEvents()` und `Api::OwnershipChanges()` enthalten Ereignisse aus dem letzten Aufruf, nicht das letzte Häkchen. Im folgenden Beispiel `secondOwnershipChangeList` sind `secondSubscriptionEvents` und leer, da ihre Funktionen unmittelbar nach dem ersten Aufruf aufgerufen werden.

Wenn Sie 10 Häkchen warten und dann `Api::AllSubscriptionEvents()` und `aufrufenApi::OwnershipChanges()`, enthalten ihre Ergebnisse sowohl Ereignisse als auch Änderungen gegenüber den letzten 10 Häkchen (nicht das letzte Häkchen).

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 nicht `SubscriptionEvents()`.

Freigabe von Lesesperren nach der Verarbeitung `SubscriptionChangeList`

Wenn Sie mit einer Aktualisierung beginnen, gibt es gemeinsam genutzte Speichersegmente für die festgeschriebenen Daten in anderen Partitionen für das vorherige Häkchen. Diese gemeinsam genutzten Speichersegmente werden möglicherweise von Lesern gesperrt. Eine App kann erst dann ein vollständiges Commit ausführen, wenn alle Leser die Sperren freigegeben haben. Zur Optimierung sollte eine App aufrufen, `Api::ReleaseReadLeases()` um die Sperren nach der Verarbeitung von `Api::SubscriptionChangeList` Elementen freizugeben. Dies reduziert Konflikte zum Commit-Zeitpunkt. `Api::Commit()` veröffentlicht die Lese-Leases standardmäßig, aber es hat sich bewährt, sie nach der Verarbeitung von Abonnementaktualisierungen 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 einer eigenständigen App-Instance zum Testen

Sie können verwenden `Api::CreateStandaloneApplication()`, um eine eigenständige App zu 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 WeaverDemo-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 nützliche Funktionen, die die Entwicklung einfacher Apps optimieren
- Eine einfachere Möglichkeit, experimentelle Funktionen des SimSpace Weaver App SDK zu testen

Wir haben das SimSpace Weaver App-SDK mit Low-Level-Zugriff auf SimSpace Weaver APIs entwickelt, 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 auf APIs bietet, die die Verwendung SimSpace Weaver vereinfachen. Die Kosten für die Benutzerfreundlichkeit sind geringer als die direkte Verwendung des SimSpace Weaver App-SDK. Simulationen, die eine geringere Leistung tolerieren können (z. B. solche ohne Leistungsanforderungen in Echtzeit), könnten gute Kandidaten für den Einsatz des Demo-Frameworks sein. Wir empfehlen, dass Sie die native Funktionalität im SimSpace Weaver App-SDK für komplexe Anwendungen verwenden, da das Demo-Framework kein vollständiges Toolkit ist.

Das Demo-Framework beinhaltet

- Funktionierende Codebeispiele, die Folgendes unterstützen und demonstrieren:

- Verwaltung des App-Flows
- Callback-gesteuerte Verarbeitung von Entitätsereignissen
- Eine Reihe von Hilfsbibliotheken von Drittanbietern:
 - spdlog(eine Logging-Bibliothek)
 - Eine Minimalversion vonAZCore (einer Mathematikbibliothek), die nur Folgendes enthält:
 - Vector3
 - Aabb
 - cxxopts(eine Parser-Bibliothek für Befehlszeilenoptionen)
- Spezifische Hilfsfunktionen fürSimSpace Weaver

Das Demo-Framework besteht aus einer Bibliothek, Quelldateien undCMakeLists. Die Dateien sind im verteilbarenSimSpace Weaver App-SDK-Paket enthalten.

Arbeiten mit Service Quotas

In diesem Abschnitt wird beschrieben, wie Sie mit den Service Quotas für arbeitenSimSpace Weaver. Kontingente werden auch Limits genannt. Eine Liste der Service Quotas finden Sie unter[SimSpace Endpunkte und Kontingente von Weaver](#). Die APIs in diesem Abschnitt stammen aus der Gruppe der App-APIs. App-APIs unterscheiden sich von den Service-APIs. Die App-APIs sind Teil desSimSpace 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 Limits für eine App](#)
- [Ermitteln Sie die Menge der von einer App verwendeten Ressourcen](#)
- [Zurücksetzen von Kennzahlen](#)
- [Überschreiten eines Limits](#)
- [Es wird kein Speicher mehr](#)
- [Bewährte Methoden](#)

Holen Sie sich die Limits für eine App

Sie können die `RuntimeLimits` App-API verwenden, um die Grenzwerte für eine App abzufragen.

```
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);
```

Ermitteln Sie die Menge der von einer App verwendeten Ressourcen

Sie können die `RuntimeMetrics` App-API aufrufen, 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 eine zurückstruct, die die Metriken enthält. Eine Zählermetrik enthält einen fortlaufenden Gesamtwert und erhöht sich nur. Eine Messgröße enthält einen Wert, der steigen oder fallen kann. Die Anwendungslaufzeit aktualisiert einen Zähler jedes Mal, 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
}
```

Zurücksetzen von Kennzahlen

Die `ResetRuntimeMetrics` App-API setzt die Werte in der `zurückAppRuntimeMetrics`struct.

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

Im folgenden Beispiel wird gezeigt, wie Sie `ResetRuntimeMetrics` in Ihrer App aufrufen.

```
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));  
}
```

Überschreiten eines Limits

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

Es wird kein Speicher mehr

SimSpace Weaver verwendet einen Garbage Collector-Prozess, um freien 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 reservierte Speicherlimit der App überschreiten. SimSpace Weaver gibt einen internen Fehler mit einer Meldung zurück, die `OutOfMemory` (und zusätzliche Details) enthält. Weitere Informationen finden Sie unter [Schreibvorgänge im Zeitverlauf verteilen](#).

Bewährte Methoden

Die folgenden bewährten Methoden stellen allgemeine Richtlinien dar, um Ihre Apps so zu gestalten, dass keine Limits. Sie gelten möglicherweise nicht für Ihr spezifisches App-Design.

Überwachen Sie häufig und fahren Sie langsamer

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

Vermeiden Sie es, Abonnementlimits und Übertragungslimits zu überschreiten

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

Schreibvorgänge im Zeitverlauf verteilen

Die Anzahl und Größe der Aktualisierungen in einem Häkchen kann erhebliche Auswirkungen auf die Zeit und den Speicherplatz haben, die für die Durchführung einer Transaktion benötigt werden. Hohe Speicheranforderungen können dazu führen, dass der Arbeitsspeicher für die Anwendungslaufzeit knapp wird. Sie können Schreibvorgänge über einen längeren Zeitraum verteilen, um die durchschnittliche Gesamtgröße der Aktualisierungen pro Tick zu verringern. Dies kann dazu beitragen, die Leistung zu verbessern und eine Überschreitung von Grenzwerten zu vermeiden. Wir empfehlen, dass Sie nicht mehr als durchschnittlich 12 MB auf jedes Häkchen oder 1,5 KB für jede Entität schreiben.

Debuggen von Simulationen

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

Themen

- [Verwenden SimSpace Weaver Local und Ansehen der Konsolenausgabe](#)
- [Sehen Sie sich Ihre Protokolle in Amazon CloudWatch Logs an](#)
- [APIdescribe-Aufrufe verwenden](#)
- [Verbinden eines Clients](#)

Verwenden SimSpace Weaver Local und Ansehen der Konsolenausgabe

Wir empfehlen Ihnen, Ihre Simulationen zuerst lokal zu entwickeln und dann in der auszuführenAWS Cloud. Sie können die Konsolenausgabe direkt anzeigen, wenn Sie mit ausführenSimSpace Weaver Local. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

Sehen Sie sich Ihre Protokolle in Amazon CloudWatch Logs an

Wenn Sie Ihre Simulation in AWS Cloud der Konsolenausgabe Ihrer Apps ausführen, wird sie an Protokollstreams in Amazon CloudWatch Logs gesendet. Ihre Simulation schreibt auch andere Protokolldaten. Sie müssen die Protokollierung in Ihrem Simulationsschema aktivieren, wenn Ihre Simulation Protokolldaten schreiben soll. Weitere Informationen finden Sie unter [SimSpace Weaver - Protokolle in Amazon CloudWatch Logs](#).

Warning

Ihre Simulation kann große Mengen an Protokolldaten erzeugen. Die Protokolldaten können sehr schnell anwachsen. Sie sollten Ihre Protokolle genau überwachen und Ihre Simulationen stoppen, wenn Sie sie nicht mehr benötigen. Die Protokollierung kann zu hohen Kosten führen.

APIdescribe-Aufrufe verwenden

Sie können die folgenden Service-APIs verwenden, um Informationen zu Ihren Simulationen in der zu erhaltenAWS Cloud.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie das SimSpace Weaver App-SDK Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `--profile cli-profile-name` Parameter für AppSimSpace Weaver-SDK-Skripte angeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

- ListSimulations – eine Liste aller Ihrer Simulationen in der abrufenAWS Cloud.

Example Beispiel

```
tools\windows\weaver-MyProject-cli.bat list-simulations
```

- DescribeSimulation – Details zu einer Simulation abrufen.

Example Beispiel

```
tools\windows\weaver-MyProject-cli.bat describe-simulation --simulation MySimulation
```

- DescribeApp – Abrufen von Details zu einer App.

Example Beispiel

```
tools\windows\weaver-MyProject-cli.bat describe-app --simulation MySimulation --  
domain MyCustomDomain --app MyCustomApp
```

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

Verbinden eines Clients

Sie können einen Client mit einer laufenden benutzerdefinierten oder 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 Beispiel-Clients finden Sie unter [Schritt 5: Sehen Sie sich Ihre Simulation an](#).

Sie finden den Quellcode für die Beispiel-Clients 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.

Benutzerdefinierte Container

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

Sie können einen benutzerdefinierten Container verwenden, um Ihre Softwareabhängigkeiten zu verwalten und zusätzliche Softwarekomponenten hinzuzufügen, 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 App-ZIP-Datei einfügen.

Important

Wir unterstützen nur AL2-Docker-Images, die in Amazon-ECR-Repositoryys gehostet werden, entweder in 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 zur Amazon Elastic Container Registry](#).

Themen

- [Erstellen eines benutzerdefinierten Containers](#)
- [Ändern eines Projekts zur Verwendung eines benutzerdefinierten Containers](#)
- [Häufig gestellte Fragen zu benutzerdefinierten Containern](#)
- [Fehlerbehebung bei benutzerdefinierten Containern](#)

Erstellen eines benutzerdefinierten Containers

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

Voraussetzungen

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

So erstellen Sie einen benutzerdefinierten Container

1. Erstellen Sie Ihre Dockerfile.

Eine Dockerfile zum Ausführen von AWS SimSpace Weaver Apps beginnt mit dem Amazon Linux 2 Image in Amazon ECR.

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

2. Erstellen Sie Ihre Dockerfile.
3. Laden Sie Ihr Container-Image in Amazon ECR hoch.
 - [Verwenden Sie die AWS Management Console.](#)
 - [Verwenden Sie 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 von verwaltete Richtlinie an Ihre IAM-Identität anfügen und es erneut versuchen. Weitere Informationen zum Anfügen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im AWS Identity and Access Management -Benutzerhandbuch.

Ändern eines Projekts zur Verwendung eines benutzerdefinierten Containers

Bei diesen Anweisungen wird davon ausgegangen, dass Sie bereits wissen, wie Sie verwenden sollen, AWS SimSpace Weaver und dass Sie Ihre Anwendungsspeicher- und Entwicklungsworkflows effizienter gestalten möchten AWS Cloud.

Voraussetzungen

- Sie ändern ein vorhandenes SimSpace Weaver Projekt, das vom `create-project.bat` Skript erstellt wurde.
- Sie haben einen benutzerdefinierten Container in Amazon Elastic Container Registry (Amazon ECR). Weitere Informationen zum Erstellen eines benutzerdefinierten Containers finden Sie unter [Erstellen eines benutzerdefinierten Containers](#).

So ändern Sie Ihr Projekt zur Verwendung eines benutzerdefinierten Containers

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 Richtliniennamen `simspaceweaver-ecr`. 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 den Namen der Simulations-App-Rolle Ihres Projekts:
- i. Öffnen Sie in einem Texteditor die AWS CloudFormation Vorlage für Ihr Projekt:

```
project-folder\cloudformation\weaver-project-name-stack.yaml
```

- ii. Suchen Sie die `RoleName` Eigenschaft unter `WeaverAppRole`. 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. Fügen Sie die `simspaceweaver-ecr` Richtlinie an die Simulations-App-Rolle des Projekts an. Weitere Informationen zum Anfügen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#) im AWS Identity and Access Management - Benutzerhandbuch.

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 Domains anzugeben.
- Fügen Sie die `image` Eigenschaft in der `app_config` für eine Domäne hinzu, die Sie ein benutzerdefiniertes Container-Image verwenden möchten. Geben Sie den Amazon-ECR-Repository-URI als Wert an. Sie können für jede Domain ein anderes Image angeben.
- Wenn nicht für eine Domäne angegeben `image` ist und angegeben `default_image` ist, verwenden Apps in dieser Domäne das Standard-Image.
- Wenn nicht für eine Domäne angegeben `image` ist und nicht angegeben `default_image` ist, werden Apps in dieser Domäne in einem SimSpace Weaver Standardcontainer ausgeführt.

Example Schemaausschnitt, der benutzerdefinierte Containerereinstellungen enthält

```

sdk_version: "1.16.0"
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 und laden Sie Ihr Projekt 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 für eine laufende Simulation nicht 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 eines Projekts zur Verwendung eines benutzerdefinierten Containers](#).

Fehlerbehebung bei benutzerdefinierten Containern

Themen

- [AccessDeniedException beim Hochladen Ihres Images in Amazon Elastic Container Registry \(Amazon ECR\)](#)
- [Eine Simulation, die einen benutzerdefinierten Container verwendet, kann nicht gestartet werden](#)

AccessDeniedException beim Hochladen Ihres Images in Amazon Elastic Container Registry (Amazon ECR)

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 von verwaltete Richtlinie an Ihre IAM-Identität anfügen und es erneut versuchen. Weitere Informationen zum Anfügen 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. Weitere Informationen finden Sie im [detaillierten Tutorial](#).
- Testen Sie Ihre Simulation ohne benutzerdefinierten Container.
- Testen Sie Ihre Simulation lokal. Weitere Informationen finden Sie unter [Lokale Entwicklung](#).

Verwenden von Python

Sie können Python für Ihre verwenden `SimSpace Weaver` Apps und Client. Das Python-Softwareentwicklungskit (Python SDK) ist Teil des Standards `SimSpace Weaver` Verteilbares App-

SDK-Paket. 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 Unterstützung für Python erfordert SimSpace Weaver Version 1.15.0 oder höher.

Themen

- [Ein Python-Projekt erstellen](#)
- [Eine Python-Simulation starten](#)
- [Der Python-Beispielclient](#)
- [Schreiben Sie Ihre eigenen Build-Skripte](#)
- [Häufig gestellte Fragen zur Verwendung von Python](#)
- [Behebung von Problemen im Zusammenhang mit Python](#)

Ein Python-Projekt erstellen

Sie verwenden die `create-project.bat` Skript, um ein Python-Projekt zu erstellen, genauso wie Sie es tun würden, um ein Nicht-Python-Projekt zu erstellen. Sie können das `PythonBubblesSample` Vorlage als Ausgangspunkt für Ihr Python-Projekt. Siehe [Erstellen Sie ein Python-Projekt](#), unten.

Benutzerdefinierter Python-Container

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

Ein benutzerdefinierter Python-Container muss Folgendes enthalten:

- gcc
- openssl-devel
- bzip2-Entwickler
- libffi-entwickeln
- wget
- tar
- gzip
- Marke
- Python (Version 3.9)

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

Das `create-custom-container.bat` Das 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 zu der hinzufügen Dockerfile:

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

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

```
Flask==2.1.1
```

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

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

Das `Dockerfile` und `requirements.txt` sind in dem `tools` Ordner Ihres Projekts.

Important

Du musst `rennencreate-custom-container.bat` nach allen Änderungen an der `Dockerfile` oder `requirements.txt`.

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 verwenden, der Python enthält (z. B. das Container-Image, das von `create-custom-container.bat` Skript), müssen Sie Python und die erforderlichen Abhängigkeiten in jede App-ZIP-Datei aufnehmen, in die Sie hochladen `SimSpace Weaver`.

Erstellen Sie ein Python-Projekt

Das folgende Verfahren gilt für Microsoft Windows. Wenn Sie das Windows-Subsystem für Linux (WSL) verwenden, verwenden Sie `.sh` Versionen von `.bat` stattdessen Skripte. Sie müssen das Setup für Amazon Elastic Container Registry (Amazon ECR) abschließen, um dieses Verfahren verwenden zu können. Weitere Informationen finden Sie unter [Einrichtung mit Amazon ECR](#) in der Amazon ECR-Benutzerhandbuch.

Um ein Python-Projekt zu erstellen

1. In einem Eingabeaufforderungsfenster, gehe zu deinem `SimSpace Weaver` SDK-Ordner.

```
cd sdk-folder
```

2. Lauf `create-project.bat` mit dem `PythonBubblesSample` Schablone.

```
.\create-project.bat --name project-name --path project-folder-parent-path --  
template PythonBubblesSample
```

3. Gehe zur `tools` Ordner in Ihrem Projektordner. Ihr *Projekt-Ordner* ist `project-folder-parent-path\project-name`.

```
cd project-folder\tools
```

4. Erstellen Sie den benutzerdefinierten Container.

```
.\create-custom-container.bat
```

Eine Python-Simulation starten

Sie können Ihre Python-basierte Simulation auf die gleiche Weise wie eine reguläre `startSimSpace WeaverSimulation`, sowohl in `SimSpace Weaver Local` und in `SimSpace Weaver` in der `AWS Cloud`.

Weitere Informationen finden Sie hier:

SimSpace Weaver Local

- [Lokale Entwicklung](#)

AWS Cloud

- [Schritt 3: Ausführen des Schnellstartskripts](#) im Schnellstart-Tutorial
- [Ausführliches Tutorial: Erfahren Sie die Details beim Erstellen der Beispielanwendung](#)

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

Der Python-Beispielclient

Wenn Sie den verwenden `PythonBubblesSampleVorlage` zum Erstellen eines Projekts, dann enthält Ihr Projekt einen Python-Beispielclient. Sie können den Beispielclient verwenden, um Folgendes anzuzeigen `PythonBubblesSampleSimulation`. Sie können den Beispielclient auch als Ausgangspunkt verwenden, um Ihren eigenen Python-Client zu erstellen.

Das folgende Verfahren geht davon aus, dass Sie einen erstellt haben `PythonBubblesSampleProjekt` und hat seine Simulation gestartet.

Um den Python-Client zu starten

1. In einem Eingabeaufforderungsfenster, gehe zum `src\PythonBubblesSample\bin` Ordner Ihres Projekts.

```
cd project-folder\src\PythonBubblesSample\bin
```

2. Führen Sie den Python-Client aus.

```
python bubbles_tkinter_client.py --host ip-address --port port-number --  
simsizes max-entitites
```

Parameter

host

Die IP-Adresse Ihrer Simulation. Für eine Simulation, die im AWS Cloud, Sie finden die IP-Adresse Ihrer Simulation im [SimSpace WeaverKonsole](#) oder verwenden Sie das Verfahren in [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#) des Schnellstart-Tutorials. Verwenden Sie für eine lokale Simulation `127.0.0.1` als IP-Adresse.

port

Die Portnummer Ihrer Simulation. Für eine Simulation, die im AWS Cloud, das ist der `ActualPortnummer`. Die Portnummer Ihrer Simulation finden Sie im [SimSpace WeaverKonsole](#) oder verwenden Sie das Verfahren in [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#) des Schnellstart-Tutorials. Verwenden Sie für eine lokale Simulation `7000` als Portnummer.

simszize

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

Schreiben Sie Ihre eigenen Build-Skripte

Sie können Ihre eigenen Build-Skripte für Ihre Python-Simulation schreiben. Für einen erfolgreichen Build müssen die folgenden Schritte ausgeführt werden:

1. Kopieren Sie den Inhalt von `src/PythonBubblesSample/` in den `Build/out/` Verzeichnis.
2. Kopieren Sie den Inhalt von `${WEAVER_SDK_DIRECTORY}/lib/weaver/weaver_python_app_sdk_v1` in den `Build/out/lib/weaver_app_sdk_v1` Verzeichnis.
3. Kopieren `${WEAVER_SDK_DIRECTORY}/lib/weaver/libweaver_app_sdk_python_v1_39.so` in die `Build/out/lib/weaver_app_sdk_v1` Verzeichnis.
4. Benennen Sie `Build/out/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_39.so` um in `libweaver_app_sdk_python_v1.so`.
5. Komprimieren Sie den Inhalt des `Build/out/` Verzeichnis.
6. Wiederholen Sie den ZIP-Vorgang für jede App-Zip-Datei, die im Schema der Simulation angegeben ist. Für die `PythonBubblesSample`, das Schema erwartet ein `project-nameSpatial.zip` und ein `project-nameView.zip`.

Nach diesen Schritten können die ZIP-Dateien in den Amazon S3-Bucket des Projekts hochgeladen werden.

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 löst eineModuleNotFoundFehler](#)

Fehler bei der Erstellung eines benutzerdefinierten Containers

Wenn Sie eine Fehlermeldung erhaltenno `basic auth credentials`nachdem du gerannt bistcreate-custom-container.batdann könnte es ein Problem mit Ihren temporären Anmeldeinformationen für Amazon ECR geben. Führen Sie den folgenden Befehl mit IhremAWS-RegionID undAWSKontonummer:

```
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, dassAWS-RegionSie angeben, ist derselbe, den Sie für Ihre Simulation verwenden. Verwenden Sie einen derAWS-RegionenDasSimSpace Weaverunterstützt. Weitere Informationen finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Nachdem du das ausgeführt hastaws ecrBefehl, führe auscreate-custom-container.batnochmal.

Weitere Ressourcen zur Fehlerbehebung, die Sie überprüfen sollten

- [Fehlerbehebung bei benutzerdefinierten Containern](#)
- [Fehlerbehebung bei Amazon ECR](#)in derAmazon ECR-Benutzerhandbuch
- [Einrichtung mit Amazon ECR](#)in derAmazon ECR-Benutzerhandbuch

Ihre Python-Simulation kann nicht gestartet werden

Sie könnten eine sehenUnable to start appFehler im Verwaltungsprotokoll Ihrer Simulation. 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 -Protokolle in Amazon CloudWatch Logs](#).

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

Ein Python-Simulations- oder View-Client löst einen `ModuleNotFoundError` aus

Python wirft einen `ModuleNotFoundError`, wenn ein erforderliches Python-Paket nicht gefunden werden kann.

Wenn sich Ihre Simulation im AWS Cloud, stellen Sie sicher, dass in Ihrem benutzerdefinierten Container alle erforderlichen Abhängigkeiten aufgeführt sind `requirements.txt`. Denk daran zu `rennen create-custom-container.bat` nochmal, wenn du `editier requirements.txt`.

Wenn Sie den Fehler für die bekommen `PythonBubblesSampleKunde`, benutzen `pip` das angegebene Paket zu installieren:

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

Unterstützung für andere Engines

Sie können Ihre eigene benutzerdefinierte C++ Engine mit verwenden `SimSpace Weaver`. Derzeit entwickeln wir Unterstützung für die folgenden Engines. Für jede dieser Engines gibt es eine separate Dokumentation.

Wichtig

Integrationen mit den hier aufgeführten Engines sind experimentell. Sie stehen als Vorschau zur Verfügung.

Engines

- [Unity](#) (Mindestversion 2021.3.7f1)
- [Unreal Engine](#) (Mindestversion 5.0)

Unity

Sie müssen die Unity Entwicklungsumgebung bereits installiert haben, bevor Sie SimSpace Weaver Simulationen mit Unity erstellen. Laden Sie das separate Unity SDK für AWS SimSpace Weaver (Unity SDK) herunter und folgen Sie den Anweisungen in diesem Paket.

Important

Sie müssen die neueste Version des SimSpace Weaver App-SDK verwenden. Die neueste Version ist 1.16.0. Weitere Informationen finden Sie unter [AWS SimSpace Weaver-Versionen](#).

So laden Sie die herunter und verwenden sie Unity SDK

1. Gehen Sie an einer Windows-Eingabeaufforderung zu Ihrem *sdk-folder* .
2. Führen Sie das Download-Skript aus. Ersetzen Sie *region* durch die , AWS-Region in der Sie Ihre Simulation starten werden (z. B. us-west-2).

```
.\download-unity-package.bat --region region
```

Das Skript wird SimSpaceWeaverUnityPackage.zip in Ihrem aktuellen Ordner heruntergeladen und entpackt.

3. Lesen Sie SimSpaceWeaverUnityPackage\Release\Documentation\Unity_SDK_for_AWS_SimSpace_Weaver.pdf.

Important

Wenn Sie in Unity die Fehlermeldung über einen fehlenden Namespace für JsonProperty oder erhaltenJsonAttribute, führen Sie die folgenden Schritte aus, um das NewtonsoftJson Paket hinzuzufügen:

1. Wählen Sie im Unity-Editor in der Menüleiste Window >Package Manager aus.
2. Wählen Sie im Fenster Package Manager die Schaltfläche + (plus) oben im Fenster aus.
3. Wählen Sie unter git URL die Option Paket hinzufügen aus.
4. Geben Sie Folgendes ein:


```
com.unity.nuget.newtonsoft-json
```

5. Wählen Sie Hinzufügen aus.

Important

Die unterstützt keine Unity SDK benannten Profile für die AWS Command Line Interface (AWS CLI). Wenn Sie -AWS IAM Identity Center oder -AWS CLIProfile verwenden, müssen Sie Ihr benanntes Profil kopieren oder in das default Profil umbenennen, bevor Sie die verwendenUnity SDK. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface -Benutzerhandbuch.

Unreal Engine

Sie müssen einen Unreal Engine dedizierten Server aus dem Quellcode erstellen. Die SimSpaceWeaverAppSdkDistributable enthält eine Version von PathfindingSample für Unreal Engine. Weitere Informationen finden Sie in den separaten Anweisungen:

```
sdk-folder\AWS_SimSpace_Weaver_Unreal_Guide.pdf
```

Verwendung von lizenzierte Software mitAWS SimSpace Weaver

AWS SimSpace Weaverermöglicht es Ihnen, Simulationen mit der Simulationsengine und dem Inhalt Ihrer Wahl zu erstellen. Im Zusammenhang mit Ihrer Nutzung vonSimSpace Weaver sind Sie dafür verantwortlich, die Lizenzbedingungen aller Software oder Inhalte, die Sie in Ihren Simulationen verwenden, zu beschaffen, zu pflegen und einzuhalten. Stellen Sie sicher, dass Ihre Lizenzvereinbarung sieht vor, dass Ihre Lizenzvereinbarung sieht vor, dass Ihre Software und Inhalte in einer virtuell gehosteten Umgebung ausgeführt werden dürfen.

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 Ihnen hilft, 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 einzelne Einheit bereitzustellen, die als [Stack](#) bezeichnet wird. Wenn Sie Ihren Stapel AWS CloudFormation löschen, können Sie alles im Stapel gleichzeitig löschen lassen. Sie können Ihre Vorlage mithilfe von Standardprozessen zur Quellcode-Verwaltung 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

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

Wie kann ich es 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 spezifizieren. 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` Sie können sie dann verwenden, AWS CloudFormation um all diese Ressourcen als Einheit und gleichzeitig bereitzustellen.

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

Die folgende Beispielvorlage erstellt eine IAM-Rolle und SimSpace Weaver Berechtigungen, die zum Ausführen von Aktionen in Ihrem Konto erforderlich sind. Die SimSpace Weaver App SDK-Skripts erstellen die Rolle und die Berechtigungen in einem bestimmten, AWS-Region wenn Sie ein Projekt

erstellen. Sie können jedoch eine AWS CloudFormation Vorlage verwenden, um die Simulation für ein anderes 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 Name der Simulation `MySimulation`. Ein Bucket für das Schema ist bereits in dem Verzeichnis vorhanden, in AWS-Region dem der Stack erstellt AWS CloudFormation wird. 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 genau 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 When AWS CloudFormation Builds the Stack vorhanden sein, sonst wird Ihre Simulation nicht gestartet. Beachten Sie, dass der Bucket-Name in diesem Beispiel das beinhaltet 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 S3-Konsole, mit den Amazon S3 S3-APIs oder mit den Amazon S3 S3-Befehlen in der überprüfen AWS CLI).

In diesem Beispiel werden einige eingebaute Funktionen und Parameter verwendet AWS CloudFormation, um die Variablenersetzung durchzuführen. Weitere Informationen finden Sie im Benutzerhandbuch unter Referenz für [intrinsische Funktionen und Referenz zu Pseudoparametern](#).
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 aus 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 `MaximumDuration` der ursprünglichen Simulation unterscheiden.

Wir empfehlen, dass Sie eine Kopie der App-Rolle Ihrer ursprünglichen Simulation erstellen und 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 einen Snapshot erstellen, um Ihre Simulationsentitätsdaten zu jedem Zeitpunkt zu sichern. SimSpace Weaver erstellt eine ZIP-Datei in einem Amazon S3 S3-Bucket. Sie können eine neue Simulation mit dem Snapshot 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 stellt die Uhr auf den entsprechenden Tick. SimSpace Weaver ruft die Konfiguration Ihrer Simulation aus dem Snapshot statt aus einer Schemadatei ab. Ihre App-ZIP-Dateien müssen sich in Amazon S3 am selben Ort befinden wie in der ursprünglichen Simulation. Sie müssen alle benutzerdefinierten Apps separat starten.

Themen

- [Anwendungsfälle für Snapshots](#)
- [Verwenden Sie das SimSpace Weaver App SDK, um mit Snapshots zu arbeiten](#)
- [Verwenden Sie die SimSpace Weaver Konsole, um mit Schnappschüssen zu arbeiten](#)
- [Verwenden Sie das AWS CLI, um mit Schnappschüssen zu arbeiten](#)
- [Verwenden Sie die SimSpace Weaver APIs, um mit Snapshots zu arbeiten](#)
- [Verwenden von Schnappschüssen mit AWS CloudFormation](#)
- [Häufig gestellte Fragen zu Snapshots](#)

Anwendungsfälle für Snapshots

Kehren Sie zu einem früheren Status zurück und erkunden Sie Verzweigungsszenarien

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

Bewährte Methoden für Notfallwiederherstellung und Sicherheit

Wir empfehlen, dass Sie Ihre Simulation regelmäßig sichern, insbesondere bei Simulationen, die länger als 1 Stunde laufen oder mehrere Mitarbeiter benötigen. Backups können Ihnen bei der Wiederherstellung nach Katastrophen und Sicherheitsvorfällen helfen. Schnappschüsse bieten Ihnen die Möglichkeit, Ihre Simulation zu sichern. Für Snapshots müssen die .zip-Dateien Ihrer App am selben Ort in Amazon S3 vorhanden sein wie zuvor. Wenn Sie in der Lage sein müssen, Ihre App-ZIP-Dateien 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 Methoden für die Sicherheit 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 ihre `erreichtMaximumDuration` erreicht. Der Höchstwert von `MaximumDuration` ist 14D (14 Tage).

Wenn Sie möchten, dass Ihre Simulation länger als die `MaximumDuration` ihrer Simulationsressource besteht, 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 weitere benutzerdefinierte Initialisierung durchführen. Sie können den Wert `MaximumDuration` der neuen Simulationsressource auf einen anderen Wert setzen, wenn Sie sie starten.

Verwenden Sie das SimSpace Weaver App SDK, um mit Snapshots zu arbeiten

Sie können die im SimSpace Weaver App SDK (Mindestversion 1.13) bereitgestellten Skripts verwenden, um Snapshots zu erstellen und zu verwenden.

Das SimSpace Weaver App SDK organisiert Ihre Simulationen nach Projekten. Sie können mehrere Simulationen von einem einzigen Projekt aus starten. Jede dieser Simulationen verwendet dasselbe Schema und dieselben .zip-Dateien für Apps. Die SimSpace Weaver App SDK-Skripts platzieren die Assets für eine Simulation in einem bestimmten Amazon S3 S3-Bucket, basierend auf dem Projektnamen, der AWS-Konto Nummer und AWS-Region. Die Skripts arbeiten mit Snapshot-Dateien, die sich in einem snapshot Ordner im Stammverzeichnis dieses Buckets befinden. Der Amazon S3 S3-URI für den snapshot Ordner hat die folgende Form:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Beispiel

- Projektname: MyProject
- AWS-KontoNummer: 111122223333
- AWS-Region: us-west-2
- Amazon S3 S3-URI des Snapshot-Ordners: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot

Wenn Sie einen anderen Amazon S3 S3-Bucket verwenden möchten, finden Sie unter den folgenden alternativen Möglichkeiten zur Arbeit mit Snapshots.

Weitere Möglichkeiten zur Arbeit mit Snapshots

- [SimSpace Weaver-Konsole](#)
- [AWS CLI](#)
- [SimSpace Weaver-APIs](#)

Themen

- [Verwenden Sie das SimSpace Weaver App SDK, um einen Snapshot zu erstellen](#)

- [Verwenden Sie das SimSpace Weaver App SDK, um eine Simulation aus einem Snapshot zu starten](#)
- [Verwenden Sie das SimSpace Weaver App-SDK, um eine Simulation schnell von einem Snapshot aus zu starten](#)
- [Verwenden Sie das SimSpace Weaver App-SDK, um Schnappschüsse für ein Projekt aufzulisten](#)

Verwenden Sie das SimSpace Weaver App SDK, um einen Snapshot zu erstellen

Um einen Snapshot zu erstellen, muss sich Ihre Simulation im STARTED Status befinden. Die Snapshot-Erstellung beginnt, nachdem das aktuelle Häkchen abgeschlossen ist. SimSpace Weaversendet keine Zecken mehr an die Apps, aber der Uhrstatus zeigt immer noch anSTARTED. Der Simulationsstatus ändert sich aufSNAPSHOT_IN_PROGRESS. Nachdem der Snapshot abgeschlossen ist, wechselt der Simulationsstatus wieder zu STARTED und die Apps erhalten erneut Zecken.

So erstellen Sie einen Snapshot

1. Gehen Sie in einer Windows-Befehlszeile zum Ordner Tools für Ihr Projekt.

```
cd project-folder\tools\windows
```

2. Wenn Sie den Namen Ihrer Simulation nicht kennen, rufen Sie die `list-simulations` API auf, um eine Liste Ihrer Simulationsressourcen zu sehen. Stellen Sie sicher, dass der Simulationsstatus lautetSTARTED.

```
.\weaver-project-name-cli.bat list-simulations
```

3. Führen Sie das `create-snapshot` Skript für Ihr Projekt aus.

```
.\create-snapshot-project-name.bat --simulation simulation-name
```

Beispiel

```
.\create-snapshot-MyProject.bat --simulation MyProjectSimulation_23-04-29_12_00_00
```

SimSpace Weavererstellt die Snapshot-Datei im Artefakt-Bucket für Ihr Projekt.

Beispiel

- Projektname: MyProject
- AWS-KontoNummer: 111122223333
- AWS-Region: us-west-2
- Amazon S3 S3-URI des Snapshot-Ordners: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot
- Name der Simulation: MyProjectSimulation_23-04-29_12_00_00
- Momentaufnahme: 29. April 2023, 15:30:27 UTC
- Name der Snapshot-Datei:
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
- Amazon S3 S3-URI der Snapshot-Datei: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot/
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip

Verwenden Sie das SimSpace Weaver App SDK, um eine Simulation aus einem Snapshot zu starten

Wenn Sie ein App-SDK-Skript verwenden, um eine Simulation von einem Snapshot aus zu starten, erstellen die Skripts auf die gleiche Weise einen neuen Simulationsnamen wie beim Starten einer Simulation ohne Snapshot.

Ihre Snapshot-Datei muss am Snapshot-Speicherort in Amazon S3 mit der folgenden Amazon S3 S3-URI vorhanden sein:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Die .zip-Dateien der App müssen sich an demselben Speicherort befinden, an dem sie sich bei der Erstellung des Snapshots befanden.

SimSpace Weaver erstellt eine neue Simulationsressource, initialisiert die State Fabric mit den im Snapshot gespeicherten Entitätsdaten, startet neue Instanzen derselben Spatial- und Service-Apps, die bei der Erstellung des Snapshots ausgeführt wurden, und stellt die Uhr auf den entsprechenden Tick. Sie müssen benutzerdefinierte Apps separat wie gewohnt starten.

Das `start-from-snapshot` Skript ist die Snapshot-Version des `start-simulation` Skripts. Genau wie das `start-simulation` Drehbuch startet das `start-from-snapshot` Skript die Uhr nicht für Sie. Sie müssen die Uhr separat starten.

Um eine Simulation von einem Snapshot aus zu starten

1. Gehen Sie in einer Windows-Befehlszeile zum Ordner Tools für Ihr Projekt.

```
cd project-folder\tools\windows
```

2. Führen Sie das `start-from-snapshot`-Skript aus.

```
.\start-from-snapshot-project-name.bat --snapshot-s3-file snapshot-file-name
```

Beispiel

```
.\start-from-snapshot-MyProject.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Verwenden Sie das SimSpace Weaver App-SDK, um eine Simulation schnell von einem Snapshot aus zu starten

Sie können eine Simulation schnell von einem Snapshot aus starten. Das ist vergleichbar mit einem Schnellstart ohne Snapshot.

Ihre Snapshot-Datei muss am Snapshot-Speicherort in Amazon S3 mit der folgenden Amazon S3 S3-URI vorhanden sein:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Die `.zip`-Dateien der App müssen sich an demselben Speicherort befinden, an dem sie sich bei der Erstellung des Snapshots befanden.

SimSpace Weaver erstellt eine neue Simulationsressource, initialisiert die State Fabric mit den im Snapshot gespeicherten Entitätsdaten, startet neue Instanzen derselben Spatial- und Service-Apps, die bei der Erstellung des Snapshots ausgeführt wurden, und stellt die Uhr auf den entsprechenden Tick. Sie müssen benutzerdefinierte Apps separat wie gewohnt starten.

Das `quick-start-from-snapshot` Skript ist die Snapshot-Version des `quick-start` Skripts. Genau wie das `quick-start` Drehbuch startet das `quick-start-from-snapshot` Drehbuch die Uhr für Sie. Außerdem wird die View-App für das Pathfinding-Beispielprojekt gestartet.

So starten Sie schnell eine Simulation aus einem Snapshot

1. Gehen Sie in einer Windows-Befehlszeile zum Ordner Tools für Ihr Projekt.

```
cd project-folder\tools\windows
```

2. Führen Sie das `quick-start-from-snapshot`-Skript aus.

```
.\quick-start-from-snapshot-project-name-cli.bat --snapshot-s3-file snapshot-file-name
```

Beispiel

```
.\quick-start-from-snapshot-MyProject-cli.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Verwenden Sie das SimSpace Weaver App-SDK, um Schnappschüsse für ein Projekt aufzulisten

Sie können das `list-snapshots` Skript verwenden, um die Schnappschüsse für ein Projekt aufzulisten. Dieses Skript listet die Dateien im `snapshot` Ordner für das Projekt auf. Projekte sind nur für das SimSpace Weaver App SDK verfügbar, sodass Sie dies nur mit den App SDK-Skripten und nur für Projekte tun können. Das Skript geht davon aus, dass es sich bei allen Dateien im `snapshot` Ordner auf Amazon S3 um Snapshot-Dateien handelt. Wenn Sie Dateien aus dem Ordner verschieben oder löschen, werden diese Dateien nicht in der Liste angezeigt.

Um die Snapshots für ein Projekt aufzulisten

1. Gehen Sie in einer Windows-Befehlszeile zum Ordner Tools für Ihr Projekt.

```
cd project-folder\tools\windows
```

2. Führen Sie das `list-snapshots`-Skript aus.

```
.\list-snapshots-project-name.bat
```

Beispiel

```
.\list-snapshots-MyProject.bat
```

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

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

Weitere Möglichkeiten zur Arbeit mit Snapshots

- [SimSpace WeaverApp SDK-Skripts](#)
- [AWS CLI](#)
- [SimSpace Weaver-APIs](#)

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 WeaverKonsole](#) her.
2. Wählen Sie im Navigationsbereich Simulationen.
3. Aktivieren Sie das Kontrollkästchen neben dem Namen der Simulation. Der Status Ihrer Simulation muss Gestartet sein.
4. Wählen Sie oben auf der Seite Create Snapshot.
5. Geben Sie unter Snapshot-Einstellungen für Snapshot-Ziel die Amazon S3 S3-URI eines Buckets oder eines Buckets und des Ordners ein, in dem Sie Ihren Snapshot erstellen SimSpace

Weaver möchten. Sie können „S3 durchsuchen“ wählen, wenn Sie Ihre verfügbaren Buckets durchsuchen und einen Standort auswählen möchten.

 **Important**

Der Amazon S3 S3-Bucket muss sich im gleichen AWS-Region befinden wie die Simulation.

 **Note**

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

6. Wählen Sie Create snapshot (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 Starten der Simulation auswählen. Alle .zip-Dateien der App aus der ursprünglichen Simulation müssen sich an den gleichen Speicherorten befinden, an denen der Snapshot erstellt wurde.

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 WeaverKonsole](#) her.
2. Wählen Sie im Navigationsbereich Simulationen.
3. Wählen Sie oben auf der Seite Simulation starten.
4. Geben Sie unter Simulationseinstellungen einen Namen und eine optionale Beschreibung für Ihre Simulation ein. Ihr Simulationsname muss in Ihrem eindeutig seinAWS-Konto.
5. Wählen Sie als Startmethode für die Simulation 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 „S3 durchsuchen“, um die Datei zu durchsuchen und auszuwählen.

⚠ Important

Der Amazon S3 S3-Bucket muss sich im gleichen 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 unter Maximale Dauer die maximale Zeit 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 Schlagworte — optional die Option Neues Tag hinzufügen aus, wenn Sie ein Tag hinzufügen möchten.
10. Wählen Sie Simulation starten.

Verwenden Sie dasAWS CLI, um mit Schnappschüssen zu arbeiten

Sie können die verwendenAWS CLI, um die 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 auf die neueste Version der AWS CLI](#) [AWS Command Line Interface](#) im -Version 2.

Weitere Möglichkeiten zur Arbeit mit Snapshots

- [SimSpace WeaverApp SDK-Skripts](#)
- [SimSpace Weaver-Konsole](#)
- [SimSpace Weaver-APIs](#)

Themen

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

Verwenden Sie denAWS 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 `aws simspaceweaver list-simulations` damit die Namen und Status Ihrer Simulationen 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 im gleichen 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 WeaverAPI-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 Name der Simulation muss in Ihrem eindeutig sein AWS-Konto. Sie können `aws simspaceweaver list-simulations` verwenden, 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 im gleichen 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.

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

Sie können die SimSpace Weaver APIs direkt aufrufen, um mit Snapshots zu arbeiten. Informationen zu den APIs finden Sie in der [AWS SimSpace WeaverAPI-Referenz](#).

Weitere Möglichkeiten zur Arbeit mit Snapshots

- [SimSpace WeaverApp SDK-Skripts](#)
- [SimSpace Weaver-Konsole](#)
- [AWS CLI](#)

Snapshot erstellen

Sie können die `CreateSnapshot` API aufrufen, um einen Snapshot für eine Simulation zu erstellen. Der Simulationsstatus muss sein `STARTED`. SimSpace Weaver erstellt die Snapshot-Datei in einem `snapshot` Ordner im Amazon S3 S3-Bucket und Objektpräfix, das Sie angeben. Weitere Informationen finden Sie unter [CreateSnapshot](#) in der AWS SimSpace Weaver-API-Referenz.

Starten Sie eine Simulation aus einem Snapshot

Sie können einen Snapshot bereitstellen, wenn Sie die `StartSimulation` API aufrufen, um eine neue Simulation zu starten. Sie geben eine JSON-Zeichenfolge als Argument für den `SnapshotS3Location` Parameter an. Die Zeichenfolge gibt den Amazon S3 S3-Bucket-Namen und Objektschlüssel der Snapshot-Datei an. Wenn Sie eine `angebenSnapshotS3Location` angeben, können Sie keine `angebenSchemaS3Location`. Weitere Informationen finden Sie unter [StartSimulation](#) in der AWS SimSpace Weaver-API-Referenz.

Häufig gestellte Fragen zu Snapshots

Läuft meine Simulation während eines Snapshots weiter?

Ihre Simulationsressourcen laufen während eines Snapshots weiter, und Sie erhalten für diesen Zeitraum weiterhin Abrechnungsgebühren. Die Zeit wird auf die maximale Dauer Ihrer Simulation angerechnet. Ihre Apps erhalten keine Zecken, während der Snapshot läuft. Wenn Ihre Uhr den Status hatte, `STARTED` als die Snapshot-Erstellung gestartet wurde, zeigt Ihre Uhr immer noch `STARTED` den Status an. Ihre Apps erhalten nach Abschluss des Snapshots erneut Zecken. Wenn Ihr Uhrstatus `warSTOPPED`, bleibt Ihr Uhrstatus erhalten `STOPPED`. Beachten Sie, dass eine Simulation mit einem `STARTED` Status ausgeführt wird, auch wenn ihr Taktstatus lautet `STOPPED`.

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

Ihre Simulation beendet den Snapshot und stoppt dann, sobald der Snapshot-Prozess beendet ist (entweder erfolgreich oder erfolglos). Wir empfehlen, dass Sie den Snapshot-Prozess vorher testen, um herauszufinden, wie lange er dauert, wie groß die zu erwartende Snapshot-Datei ist und ob er erfolgreich abgeschlossen werden sollte.

Was passiert, wenn ich eine Simulation beende, für die 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 stoppen, besteht darin, die Simulation zu beenden. Sie können eine Simulation nicht neu starten, nachdem Sie sie gestoppt haben.

Wie lange wird es dauern, bis mein Snapshot fertiggestellt ist?

Die Zeit, die benötigt wird, um einen Snapshot zu erstellen, 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, dass Sie den Snapshot-Prozess vorher testen, um herauszufinden, wie groß die Datei für Ihre Simulation sein könnte.

Messaging

Die Messaging-API vereinfacht die Anwendung bis zur Anwendung innerhalb der Simulation. APIs zum Senden und Empfangen von Nachrichten sind Teil des SimSpace Weaver App-SDK. Messaging verwendet derzeit einen Best-Effort-Ansatz, um Nachrichten zu senden und zu empfangen. SimSpace Weaver versucht, Nachrichten im nächsten Simulations-Tick zu senden/zu empfangen, es gibt jedoch keine Garantien für Zustellungs-, Bestell- oder Ankunftszeit.

Themen

- [Anwendungsfälle für Messaging](#)

- [Verwenden der Messaging-APIs](#)
- [Wann sollte Messaging verwendet werden?](#)
- [Tipps bei der Arbeit mit Messaging](#)
- [Messaging-Fehler und Fehlerbehebung](#)

Anwendungsfälle für Messaging

Kommunikation zwischen Simulationsanwendungen

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

Bestätigen des Empfangs einer Nachricht

Gesendete Nachrichten enthalten Informationen über den Sender im Nachrichten-Header. Verwenden Sie diese Informationen, um eine Bestätigungsantwort beim Empfang einer Nachricht zurückzusenden.

Weiterleiten von Daten, die von einer benutzerdefinierten App empfangen wurden, an andere Apps innerhalb der Simulation

Messaging ist kein Ersatz dafür, wie Clients eine Verbindung zu benutzerdefinierten Apps herstellen, die in ausgeführt werden SimSpace Weaver. Messaging ermöglicht es Benutzern jedoch, Daten von benutzerdefinierten Apps, die Clientdaten empfangen, an andere Apps weiterzuleiten, 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.

Verwenden der Messaging-APIs

Die Messaging-APIs sind im SimSpace Weaver App-SDK enthalten (Mindestversion 1.16.0). 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 von 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 von 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 von Nachrichten

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

Themen

- [Senden von Nachrichten](#)
- [Empfangen von Nachrichten](#)
- [Antwort an den Sender](#)

Senden von Nachrichten

Nachrichten bestehen aus einer Transaktion (ähnlich wie andere Weaver-API-Aufrufe), einer Nutzlast und einem Ziel.

Nachrichten-Payload

Die Nachrichtennutzlast ist eine flexible Datenstruktur von bis zu 256 Byte. Als bewährte Methode zum Erstellen Ihrer Nachrichtennutzlasten empfehlen wir Folgendes.

So erstellen Sie die Nachrichtennutzlast

1. Erstellen Sie eine Datenstruktur (z. B. in `struct 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 `-MessagePayloadObjekt`.

Nachrichtenziel

Das Ziel einer Nachricht wird durch das `-MessageEndpointObjekt` definiert. Dazu gehören sowohl ein Endpunkttyp als auch eine Endpunkt-ID. Der einzige derzeit unterstützte Endpunkttyp ist `Partition`, mit dem Sie Nachrichten an andere Partitionen in der Simulation adressieren können. Die Endpunkt-ID ist die `Partitions-ID` Ihres Ziels.

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

Anleitungen zum Auflösen eines Nachrichtenendpunkts von einer Position finden Sie unter [Tipps bei der Arbeit mit Messaging](#).

Senden der Nachricht

Sie können die API verwenden, nachdem Sie die `ZielSendMessage-` und `Nutzlastobjekte` 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 dem aktuellen Simulations-Tick.

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 an die eingehende Nachrichtenwarteschlange 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

```

Antwort an den Sender

Jede empfangene Nachricht enthält einen Nachrichten-Header mit Informationen über den ursprünglichen Sender 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?

Messaging in SimSpace Weaver bietet ein weiteres Muster für den Austausch von Informationen zwischen Simulationsanwendungen. Abonnements bieten einen Pull-Mechanismus zum Lesen von Daten aus bestimmten Anwendungen oder Bereichen der Simulation. Nachrichten bieten einen Push-Mechanismus zum Senden von Daten an bestimmte Anwendungen oder Bereiche der Simulation.

Im Folgenden finden Sie zwei Anwendungsfälle, in denen es hilfreicher ist, Daten mithilfe von Messaging zu übertragen, anstatt Daten über ein Abonnement abzurufen oder zu lesen.

Example 1: Senden eines Befehls an eine andere App zum Ändern einer Entitätsposition

```
// 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 empfangenden Seite 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 Create-Entitätsnachricht an eine Geo-App

```

struct WeaverMessage
{
    const Aws::WeaverRuntime::Api::TypeId messageId;
};

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 empfangenden Seite erstellt die App eine neue Entität in State Fabric und aktualisiert ihre 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 bei der Arbeit mit Messaging

Lösen eines Endpunkts von einer Position oder einem App-Namen

Sie können die `AllPartitions` Funktion verwenden, um die räumlichen Grenzen und eine Domänen-ID abzurufen, die Sie benötigen, um Nachrichtenpartitions-IDs und Nachrichtenziele zu

ermitteln. Wenn Sie jedoch die Position kennen, an die 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);
};
```

Serialisieren und Deserialisieren der Nachrichtennutzlast

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());
}
```

Messaging-Fehler und Fehlerbehebung

Es können die folgenden Fehler auftreten, wenn Sie die Messaging-APIs verwenden.

Fehler bei der Endpunktauflösung

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

Domännennamenprüfung

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 noch nicht an der Simulation beteiligt ist. Verwenden Sie die `DescribeSimulation`-API, um sicherzustellen, dass Ihre benutzerdefinierte App gestartet wurde, bevor Sie eine Nachricht an sie senden. Dieses Verhalten ist in SimSpace Weaver Local und gleichAWS Cloud.

Positionsprüfung

Der Versuch, einen Endpunkt mit einem gültigen Domännennamen, 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, die `MessageEndpointResolver` in der `MessageUtils` Bibliothek zu verwenden, die im SimSpace Weaver App-SDK enthalten ist.

Fehler beim Senden von Nachrichten

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

Sendelimit für Nachrichten pro App, pro Häkchen, überschritten

Das aktuelle Limit für die Anzahl der Nachrichten, die pro App und Simulations-Tick gesendet werden können, beträgt 128. Nachfolgende Aufrufe für dasselbe Häkchen schlagen mit dem folgenden Fehler fehl:

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

SimSpace Weaver versucht, beim nächsten Häkchen nicht gesendete Nachrichten zu senden. Senken Sie die Sendehäufigkeit, um dieses Problem zu beheben. Kombinieren Sie Nachrichtennutzlasten, die kleiner als das Limit von 256 Byte sind, um die Anzahl der ausgehenden Nachrichten zu verringern.

Dieses Verhalten ist in SimSpace Weaver Local und in der gleichAWS Cloud.

Nachrichtennutzlast-Größenbeschränkung überschritten

Das aktuelle Limit für die Nachrichtennutzlastgröße beträgt 256 Byte sowohl in als auch SimSpace Weaver Local 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 1 die Prüfung nicht besteht, wird nur diese 1 Nachricht abgelehnt. SimSpace Weaver sendet die anderen 9 Nachrichten.

Dieses Verhalten ist in SimSpace Weaver Local und gleichAWS 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 in SimSpace Weaver Local und gleichAWS Cloud.

Best-Effort-Messaging

SimSpace Weaver garantiert keine Nachrichtenzustellung. Der Service versucht, die Zustellung von Nachrichten im nachfolgenden Simulations-Tick abzuschließen, aber Nachrichten können verloren gehen oder verzögert werden.

Bewährte Methoden bei der Arbeit mit SimSpace Weaver

Wir empfehlen die folgenden bewährten Methoden bei der Arbeit mit SimSpace Weaver.

Themen

- [Einrichten von Abrechnungsalarmen](#)
- [Verwenden von SimSpace Weaver Local](#)
- [Simulationen stoppen, die Sie nicht benötigen](#)
- [Löschen von Ressourcen, die Sie nicht benötigen](#)
- [Backups haben](#)

Einrichten von Abrechnungsalarmen

Es ist einfach, Ressourcen in bereitzustellen AWS und sie laufen zu lassen, auch wenn sie nicht mehr benötigt werden. Dies kann zu Kosten für unübersichtliche Abläufe führen, die bei Erhalt Ihrer Rechnung überraschend sein können. Sie können in Amazon einen Alarm konfigurieren CloudWatch , der ausgelöst und Sie benachrichtigt, wenn Ihre Kosten einen von Ihnen festgelegten Schwellenwert überschreiten. Sie können Ihre Kosten mithilfe von Kostenmanagement-Tools untersuchen. Weitere Informationen finden Sie hier:

- [Erstellen eines Abrechnungsalarms zur Überwachung Ihrer geschätzten AWS Gebühren](#)
- [Was ist AWS Cost Management?](#)

Verwenden von SimSpace Weaver Local

Wir empfehlen Ihnen, zu verwenden, SimSpace Weaver Local um Ihre Simulationen zu entwickeln und zu testen, bevor Sie sie in den SimSpace Weaver Service in der hochladenAWS Cloud. Zu den Vorteilen der Entwicklung von mit SimSpace Weaver Local gehören:

- Auf große Uploads muss nicht gewartet werden
- Keine Begrenzung der Anzahl der lokalen Simulationen, die Sie erstellen können
- Ihnen wird die Rechenzeit auf Ihrem lokalen Computer nicht in Rechnung gestellt
- Direkter Zugriff auf die Konsolenausgabe von Ihren Apps

- Ändern, neu erstellen und neu starten Sie Ihre lokale Simulation, ohne sie in der neu erstellen zu müssen AWS Cloud

Simulationen stoppen, die Sie nicht benötigen

Sie erhalten Abrechnungsgebühren für eine Simulation, während sie ausgeführt wird. Sie müssen eine Simulation beenden, damit keine Gebühren dafür mehr anfallen. Laufende Simulationen werden auch auf Ihr Kontingent für die maximale Anzahl von Simulationen angerechnet. Eine laufende Simulation, für die die Protokollierung konfiguriert ist, kann auch große Mengen an Protokollen generieren, für die Sie auch Abrechnungsgebühren erhalten. Sie sollten alle Simulationen beenden, die Sie nicht benötigen, um keine zusätzlichen Gebühren mehr zu erhalten.

Important

Wenn Sie die Simulationsuhr anhalten, wird die Simulation nicht angehalten, sondern die Veröffentlichung von Häkchen in Ihren Apps beendet. Sie können eine Simulation nicht neu starten, nachdem Sie sie angehalten haben.

Löschen von Ressourcen, die Sie nicht benötigen

Jede Simulation, die Sie in erstellen, erstellt SimSpace Weaver auch Ressourcen in anderen - AWS Services. Sie können Abrechnungsgebühren für Ressourcen und Daten in diesen anderen Services erhalten. 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 Services vorhanden sind, möglicherweise nicht gelöscht. Simulationsprotokolldaten in Amazon CloudWatch Logs bleiben beispielsweise dort, bis Sie sie löschen. Für diese Protokolldaten fallen Gebühren an. Sie sollten alle zugehörigen Ressourcen für Ihre Simulationen bereinigen, wenn Sie sie nicht mehr benötigen. Weitere Informationen finden Sie unter [Schritt 6: Anhalten und Bereinigen Ihrer Simulation](#) im Schnellstart-Tutorial.

Backups haben

Es ist eine gute Idee, Backups und Backup-Pläne für alles zu haben. Sie sollten das nicht annehmen, nur weil sich Ihre Daten darin befinden AWS, dass Sie sie nicht sichern müssen. Sie müssen Ihr

eigenes System erstellen, wenn Sie Ihren Simulationsstatus sichern müssen. Erwägen Sie, mehrere zu verwenden AWS-Regionen und einen Plan einzurichten, um Ihren Produktions-Workload AWS-Region bei Bedarf schnell auf einen anderen umstellen zu können. Weitere Informationen zu AWS-Regionen, die unterstützen SimSpace Weaver, finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Sicherheit im AWS SimSpace Weaver

Die Sicherheit in der Cloud hat für AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

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

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für AWS SimSpace Weaver gelten, finden Sie unter [Im Rahmen des Compliance-Programms zugelassene AWS-Services](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. 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 geteilten Verantwortung bei der Verwendung von SimSpace Weaver einsetzen können. Die folgenden Themen veranschaulichen, wie Sie SimSpace Weaver zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre SimSpace Weaver-Ressourcen zu überwachen und zu schützen.

Themen

- [Datenschutz in AWS SimSpace Weaver](#)
- [Identitäts- und Zugriffsverwaltung für AWS SimSpace Weaver](#)
- [Protokollierung und Überwachung von Sicherheitsereignissen in AWS SimSpace Weaver](#)
- [Compliance-Validierung für AWS SimSpace Weaver](#)
- [Ausfallsicherheit in AWS SimSpace Weaver](#)
- [Sicherheit der Infrastruktur in AWS SimSpace Weaver](#)
- [Konfigurations- und Schwachstellenanalyse in AWS SimSpace Weaver](#)
- [Bewährte Methoden für die Sicherheit für SimSpace Weaver](#)

Datenschutz in AWS SimSpace Weaver

Das [Modell der geteilten Verantwortung](#) von AWS gilt für den Datenschutz in AWS SimSpace Weaver. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. 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, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in 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 auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

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 SimSpace Weaver oder anderen AWS-Services über die Konsole, API, AWS CLI oder AWS-SDKs arbeiten. 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 werden im Ruhezustand betrachtet, wenn sie sich im nicht flüchtigen (persistenten) Datenspeicher befinden, z. B. auf einer Festplatte. Daten, die sich im flüchtigen Datenspeicher befinden, wie Arbeitsspeicher und Register, gelten nicht als im Ruhezustand.

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

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

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

Informationen zum Verschlüsseln Ihrer Daten im Ruhezustand finden Sie unter:

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

Verschlüsselung während der Übertragung

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

Intern SimSpace Weaver verwendet TLS, um eine Verbindung zu anderen -AWS Services herzustellen, die es verwendet.

Important

Die Kommunikation zwischen Ihren Apps und ihren Clients umfasst nicht SimSpace Weaver. Es liegt in Ihrer Verantwortung, die Kommunikation mit Simulationsclients zu verschlüsseln, falls erforderlich. Wir empfehlen Ihnen, eine Lösung zu erstellen, um alle Daten während der Übertragung über Client-Verbindungen hinweg zu verschlüsseln.

Weitere Informationen zu AWS Services, die Ihre Verschlüsselungslösungen unterstützen können, finden Sie [im -AWSSicherheitsblog](#).

Datenschutz für den Datenverkehr zwischen Netzwerken

SimSpace Weaver -Rechenressourcen befinden sich in 1 Amazon VPC, die von allen SimSpace Weaver Kunden gemeinsam genutzt wird. Der gesamte interne SimSpace Weaver Service-Datenverkehr bleibt im AWS Netzwerk und wird nicht über das Internet übertragen. Die Kommunikation zwischen Simulationsclients und Ihren Apps wird über das Internet übertragen.

Identitäts- und Zugriffsverwaltung für AWS SimSpace Weaver

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, um SimSpace Weaver Ressourcen zu nutzen. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Featuresweise von AWS SimSpace Weaver mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#)
- [Berechtigungen, die für Sie SimSpace Weaver erstellt](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Fehlerbehebung für AWS SimSpace Weaver-Identität und -Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in SimSpace Weaver.

Service-Benutzer: Wenn Sie den SimSpace Weaver-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere SimSpace Weaver-Funktionen ausführen, 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 für AWS SimSpace Weaver-Identität und -Zugriff](#) finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Funktion in SimSpace Weaver haben.

Service-Administrator: Wenn Sie in Ihrem Unternehmen für SimSpace Weaver-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf SimSpace Weaver. Es ist Ihre Aufgabe, zu bestimmen, auf welche SimSpace Weaver-Funktionen und Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit SimSpace Weaver verwenden kann, finden Sie unter [Featuresweise von AWS SimSpace Weaver 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 identitätsbasierte SimSpace Weaver-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, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuertem Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen

Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter [Signieren von AWS-API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center-Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, 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 empfiehlt es sich, menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, aufzufordern, den Verbund mit einem Identitätsanbieter zu verwenden, um auf AWS-Services mit temporären Anmeldeinformationen zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web Identity Provider, AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt werden, auf AWS-Services zugreift. Wenn Verbundidentitäten auf AWS-Konten zugreifen, 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 im IAM Identity Center erstellen oder Sie können eine Verbindung mit einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie in allen AWS-Konten und 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 in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. 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 einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten 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 [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie [Rollen wechseln](#). Sie können eine Rolle annehmen, 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 [Verwenden von IAM-Rollen](#) 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](#) 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. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff: Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. 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 zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur dann gestellt, wenn ein Service eine Anfrage erhält, die eine Interaktion mit anderen AWS-Services oder -Ressourcen erfordert. 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.

- **Serviceverknüpfte Rolle:** Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen in Amazon EC2** – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und – AWS CLI oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in 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 festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

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 Benutzerinformationen über die AWS Management Console, die AWS CLI oder die 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 [Erstellen von IAM-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 AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl 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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete 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.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) 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 dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter [Logik für die Richtlinienauswertung](#) im IAM-Benutzerhandbuch.

Featuresweise von AWS SimSpace Weaver mit IAM

Bevor Sie IAM zum Verwalten des Zugriffs auf SimSpace Weaver verwenden, erfahren Sie, welche IAM-Funktionen Sie mit SimSpace Weaver verwenden können.

IAM-Funktionen, die Sie mit AWS SimSpace Weaver verwenden können

IAM-Feature	SimSpace Weaver-Support
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Ja
Servicerollen	Ja

IAM-Feature	SimSpace Weaver-Support
Service-verknüpfte Rollen	Nein

Einen Überblick über das Zusammenwirken von SimSpace Weaver und anderen AWS-Services mit den meisten IAM-Funktionen finden Sie unter [AWS-Services, die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

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 [Erstellen von IAM-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 identitätsbasierte SimSpace Weaver-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#).

Ressourcenbasierte Richtlinien 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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

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 in unterschiedlichen AWS-Konten befinden, muss ein IAM-Administrator im vertrauenswürdigen Konto auch der Prinzipalentität (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 [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für SimSpace Weaver

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

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 die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keine passende API-Operation 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 [Von AWS SimSpace Weaver definierte Aktionen](#) in der Service-Autorisierungs-Referenz.

Richtlinienaktionen in SimSpace Weaver verwenden das folgende Präfix vor der Aktion:

```
simspaceweaver
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "simspaceweaver:action1",  
  "simspaceweaver:action2"  
]
```

Beispiele für identitätsbasierte SimSpace Weaver-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#).

Richtlinienressourcen für SimSpace Weaver

Unterstützt Richtlinienressourcen

Ja

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen 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 finden Sie unter [Von AWS SimSpace Weaver definierte Ressourcen](#) in der Service-Autorisierungs-Referenz. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von AWS SimSpace Weaver definierte Aktionen](#).

Beispiele für identitätsbasierte SimSpace Weaver-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#).

Richtlinien-Bedingungsschlüssel für SimSpace Weaver

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja
---	----

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. 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, wertet AWS die Bedingung mittels 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 servicespezifische Bedingungsschlüssel. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

Eine Liste von SimSpace Weaver-Bedingungsschlüsseln finden Sie unter [Bedingungsschlüssel für AWS SimSpace Weaver](#) in der Service-Autorisierungs-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von AWS SimSpace Weaver definierte Aktionen](#).

Beispiele für identitätsbasierte SimSpace Weaver-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für AWS SimSpace Weaver](#).

Zugriffssteuerungslisten (ACLs) in SimSpace Weaver

Unterstützt ACLs	Nein
------------------	------

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (Attribute-Based Access Control, 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 AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und mehrere AWS-Ressourcen anfügen. 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 [Was ist ABAC?](#) 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 AWS-Services Featureieren nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, unter anderem darüber, welche AWS-Services mit temporären Anmeldeinformationen arbeiten, finden Sie unter [AWS-Services, die mit IAM arbeiten](#) im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der AWS Management Console anmelden. Wenn Sie beispielsweise über den Single Sign-On (SSO)-Link Ihres Unternehmens auf AWS zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. 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 zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Sie können mithilfe der AWS CLI- oder AWS-API manuell temporäre Anmeldeinformationen erstellen. Sie können dann diese temporären Anmeldeinformationen verwenden, um auf AWS zuzugreifen. AWS empfiehlt, dass Sie temporäre Anmeldeinformationen dynamisch generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Serviceübergreifende Prinzipal-Berechtigungen für SimSpace Weaver

Unterstützt Forward Access Sessions (FAS)	Ja
---	----

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion

in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur dann gestellt, wenn ein Service eine Anfrage erhält, die eine Interaktion mit anderen AWS-Services oder -Ressourcen erfordert. 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

Die Änderung der Berechtigungen für eine Servicerolle kann die Funktionalität von SimSpace Weaver beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn SimSpace Weaver eine Anleitung dazu gibt.

Die SimSpace Weaver App-SDK-Skripte verwenden eine -AWS CloudFormationVorlage, um Ressourcen in anderen -AWSServices zur Unterstützung Ihrer Simulation zu erstellen. Eine dieser Ressourcen ist die App-Rolle für Ihre Simulation. SimSpace Weaver übernimmt die App-Rolle, um in AWS-Konto Ihrem Namen Aktionen in Ihrem durchzuführen, z. B. um Protokolldaten in - CloudWatch Protokolle zu schreiben. Weitere Informationen zur App-Rolle finden Sie unter [Berechtigungen, die für Sie SimSpace Weaver erstellt](#).

Serviceverknüpfte Rollen für SimSpace Weaver

Unterstützt serviceverknüpfte Rollen

Nein

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen.

Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. 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 über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die 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](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von SimSpace Weaver definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für AWS SimSpace Weaver](#) in der Service-Autorisierungs-Referenz.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der SimSpace Weaver-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Benutzern das Erstellen und Ausführen von Simulationen erlauben](#)

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 verursauchen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- **Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen:** Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete AWS-Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [AWS-verwaltete Richtlinien für Auftragsfunktionen](#) 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 Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. 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 zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- **Bedarf einer Multi-Faktor-Authentifizierung (MFA):** Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, 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 [Konfigurieren eines MFA-geschützten API-Zugriffs](#) 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 zuzugreifen, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu den SimSpace Weaver-Ressourcen in Ihrem AWS-Konto gestatten. 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.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. 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 weiterhin die SimSpace Weaver Konsole verwenden können, fügen Sie den Entitäten auch die von SimSpace Weaver *ConsoleAccess* oder *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 enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-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": "*"
  }
]
}

```

Benutzern das Erstellen und Ausführen von Simulationen erlauben

Diese Beispiel-IAM-Richtlinie enthält die grundlegenden Berechtigungen, die zum Erstellen und Ausführen von Simulationen in erforderlich sind 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"
        }
    }
}
]
}
}

```

Berechtigungen, die für Sie SimSpace Weaver erstellt

Wenn Sie ein SimSpace Weaver Projekt erstellen, erstellt der Service 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, damit sie Vorgänge für Sie ausführen kann.

Richtlinie für App-Rollenberechtigungen

Die Rolle der Simulations-App verfügt über 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 Simulations-App-Rolle eine Vertrauensstellung 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 über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere Entität zum Ausführen der Aktion verlocken kann. In AWS kann der serviceübergreifende Identitätswechsel zu Confused-Deputy-Problem 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 Verwendung der globalen Bedingungskontext-Schlüssel [aws:SourceArn](#) und [aws:SourceAccount](#) in ressourcenbasierten Richtlinien, um die Berechtigungen, die AWS SimSpace Weaver einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. 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 unbekanntenen Teile des ARN. Beispiel:

```
arn:aws:simspaceweaver:*:111122223333:*
```

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in SimSpace Weaver verwenden können, um das Confused-Deputy-Problem zu vermeiden. Diese Richtlinie erlaubt es nur, die Rolle SimSpace Weaver zu übernehmen, wenn die Anforderung vom angegebenen Quellkonto stammt und mit dem angegebenen ARN bereitgestellt wird. In diesem Fall SimSpace Weaver kann die Rolle nur für Anfragen von Simulationen im eigenen Konto des Anforderers (*111122223333*) und nur in der angegebenen Region (*us-west-2*) übernehmen.

```
{
  "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 Möglichkeit, 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 Ihr `aws:SourceArn` explizit eine -Kontonummer enthält, können Sie den `Condition` Elementtest für die auslassen `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 für AWS SimSpace Weaver-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit SimSpace Weaver und IAM auftreten könnten.

Themen

- [Ich bin nicht autorisiert, eine Aktion in SimSpace Weaver auszuführen.](#)
- [Ich bin nicht autorisiert, iam durchzuführen:PassRole](#)
- [Ich möchte meine Zugriffsschlüssel anzeigen](#)
- [Ich bin Administrator und möchte anderen Zugriff auf SimSpace Weaver gewähren.](#)
- [Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine SimSpace Weaver-Ressourcen gewähren](#)

Ich bin nicht autorisiert, eine Aktion in SimSpace Weaver auszuführen.

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, 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 autorisiert, iam durchzuführen: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 erlauben die Übergabe einer vorhandenen Rolle an diesen Dienst, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. 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.


Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. 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. wJalrXUtnFEMI/

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](#). Wenn Sie dies tun, gewähren Sie anderen Personen möglicherweise den permanenten Zugriff auf Ihr 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 Zugriff auf SimSpace Weaver gewähren.

Um anderen Personen oder einer Anwendung Zugriff auf SimSpace Weaver zu gewähren, müssen Sie eine IAM-Entität (Benutzer oder Rolle) für die Person oder Anwendung erstellen, die Zugriff benötigt. Sie werden die Anmeldeinformationen für diese Einrichtung verwenden, um auf AWS zuzugreifen. Anschließend müssen Sie der Entität eine Richtlinie anfügen, die dieser die korrekten Berechtigungen in SimSpace Weaver gewährt.

Informationen zum Einstieg finden Sie unter [Erstellen Ihrer ersten delegierten IAM-Benutzer und -Gruppen](#) im IAM-Benutzerhandbuch.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine SimSpace Weaver-Ressourcen gewähren

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. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob SimSpace Weaver diese Funktionen unterstützt, finden Sie unter [Featuresweise von AWS SimSpace Weaver mit IAM](#).
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter [Gewähren des Zugriffs auf AWS-Konten von externen Benutzern](#) im IAM-Benutzerhandbuch.
- 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 [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Protokollierung und Überwachung von Sicherheitsereignissen in AWS SimSpace Weaver

Die Überwachung ist ein wichtiger Teil der Wahrung von Zuverlässigkeit, Verfügbarkeit und Performance von SimSpace Weaver und Ihren AWS-Lösungen. Sie sollten von allen Teilen Ihrer AWS-Lösung Überwachungsdaten sammeln, damit Sie Ausfälle, die sich über mehrere Punkte erstrecken, leichter debuggen können.

AWS und SimSpace Weaver bieten mehrere Tools zur Überwachung Ihrer Simulationsressourcen und zur Reaktion auf potenzielle Vorfälle.

Protokolle in Amazon CloudWatch

SimSpace Weaver speichert seine Protokolle in CloudWatch. Sie können diese Protokolle verwenden, um Ereignisse in Ihrer Simulation (z. B. das Starten und Stoppen von Apps) sowie zum Debuggen zu überwachen. Weitere Informationen finden Sie unter [SimSpace Weaver -Protokolle in Amazon CloudWatch Logs](#).

Amazon- CloudWatch Alarme

Mithilfe von Amazon- CloudWatch Alarmen überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS-Thema oder eine AWS Auto Scaling-Richtlinie gesendet. CloudWatch Alarme werden ausgelöst, wenn sich ihr Status ändert und für eine bestimmte Anzahl von Zeiträumen beibehalten wird, nicht durch einen bestimmten Status. Weitere Informationen finden Sie unter [Überwachung SimSpace Weaver mit Amazon CloudWatch](#).

AWS CloudTrail Protokolle

CloudTrail bietet eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem -AWSService in durchgeführten Aktionen SimSpace Weaver. Anhand der von CloudTrail gesammelten Informationen können Sie die angeforderte Anfrage SimSpace Weaver, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen. Weitere Informationen finden Sie unter [Protokollieren von AWS SimSpace Weaver-API-Aufrufen mithilfe von AWS CloudTrail](#).

Compliance-Validierung für AWS SimSpace Weaver

SimSpace Weaver ist nicht im Umfang der AWS-Compliance-Programme enthalten.

Externe Prüfer bewerten im Rahmen verschiedener AWS -Compliance-Programme die Sicherheit und Compliance anderer -AWSServices. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.


Informationen darüber, ob ein AWS-Service in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter [AWS-Services in Geltungsbereich nach Compliance-Programm](#). Wählen Sie das Compliance-Programm, das Sie interessiert. Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Berichte herunterladen in AWS Artifact](#).

Ihre Compliance-Verantwortung bei der Verwendung von AWS-Services ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#): In diesen Bereitstellungsleitfäden werden Überlegungen zur Architektur erörtert und Schritte zum Bereitstellen von Basisumgebungen auf AWS zur Verfügung gestellt, die auf Sicherheit und Compliance ausgerichtet sind.

- [Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance in Amazon Web Services](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-berechtigte Anwendungen erstellen können.

 Note

Nicht alle AWS-Services sind HIPAA-berechtigt. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS-Compliance-Ressourcen](#) – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- [AWS-Compliance-Leitfäden für Kunden](#): Verstehen Sie das Modell der geteilten Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Methoden zum Schutz von AWS-Services zusammengefasst 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.
- [Auswertung von Ressourcen mit Regeln](#) im AWS ConfigEntwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- [AWS Security Hub](#): Dieser AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb von AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS-Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [AWS Audit Manager](#) – Dieser AWS-Service hilft Ihnen, Ihre AWS-Nutzung kontinuierlich zu überprüfen, um den Umgang mit Risiken und die Compliance von Branchenstandards zu vereinfachen.

Ausfallsicherheit in AWS SimSpace Weaver

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz 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 über AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Sicherheit der Infrastruktur in AWS SimSpace Weaver

Als verwalteter Service ist AWS SimSpace Weaver durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt. Informationen zu AWS-Sicherheitsservices und wie AWS die Infrastruktur schützt, finden Sie unter [AWS-Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf SimSpace Weaver zuzugreifen. 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 Datenverarbeitungs-Instances innerhalb einer Amazon VPC ausgeführt, die sich in einer von Ihnen ausgewählten -AWSRegion befindet. Eine Amazon VPC ist ein virtuelles Netzwerk in der AWS Cloud, das die Infrastruktur nach Workload oder Organisationseinheit isoliert. Die Kommunikation zwischen Datenverarbeitungs-Instances innerhalb der Amazon VPC bleibt innerhalb des -AWSNetzwerks und wird nicht über das Internet übertragen. Einige interne Servicekommunikationen werden über das Internet übertragen und sind verschlüsselt. Simulationen für alle Kunden, die in derselben AWS Region ausgeführt werden, haben dieselbe Amazon VPC. Simulationen für verschiedene Kunden verwenden separate Rechen-Instances innerhalb derselben Amazon VPC.

Die Kommunikation zwischen Ihren Simulationsclients und Ihren Simulationen, die SimSpace Weaver über das Internet ausgeführt werden, verarbeitet diese Verbindungen SimSpace Weaver nicht. Es liegt in Ihrer Verantwortung, Ihre Client-Verbindungen zu sichern.

Ihre Verbindungen zum SimSpace Weaver Service über das Internet und sind verschlüsselt. Dazu gehören Verbindungen mit AWS Management Console, AWS Command Line Interface (AWS CLI), AWS Software Development Kits (SDK) und dem SimSpace Weaver App-SDK.

Konfigurations- und Schwachstellenanalyse in AWS SimSpace Weaver

Konfigurations- und IT-Kontrollen sind eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Weitere Informationen finden Sie im Modell der AWS übergreifenden VerantwortlichkeitModell der . AWS übernimmt grundlegende Sicherheitsaufgaben für die zugrunde liegende Infrastruktur, wie z. B. das Patchen des Betriebssystems auf Datenverarbeitungs-Instances, die Firewall-Konfiguration und AWS die Infrastruktur-Notfallwiederherstellung. <https://aws.amazon.com/compliance/shared-responsibility-model/> 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 Aktualisierungen von 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, wenn Sie die Simulation neu erstellen müssen.

Bewährte Methoden für die Sicherheit für SimSpace Weaver

In diesem Abschnitt werden bewährte Sicherheitsmethoden beschrieben, die spezifisch für SimSpace Weaver sind. Weitere Informationen zu bewährten Sicherheitsmethoden in AWS finden Sie unter [Bewährte Methoden für Sicherheit, Identität und Compliance](#).

Themen

- [Verschlüsseln der Kommunikation zwischen Ihren Apps und ihren Clients](#)
- [Sichern Sie Ihren Simulationsstatus regelmäßig](#)
- [Verwalten Ihrer Apps und SDKs](#)

Verschlüsseln der Kommunikation zwischen Ihren Apps und ihren Clients

SimSpace Weaver verwaltet keine Kommunikation zwischen Ihren Apps und ihren Clients. Sie sollten eine Form der Authentifizierung und Verschlüsselung für Clientsitzungen implementieren.

Sichern Sie Ihren Simulationsstatus regelmäßig


SimSpace Weaver speichert Ihren Simulationsstatus nicht. Simulationen, die angehalten werden (aufgrund eines API-Aufrufs, einer Konsolenoption oder eines Systemabsturzes), speichern ihren Status nicht und haben keine inhärente Möglichkeit, sie wiederherzustellen. Angehaltene 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 [Datenbankservices](#), mit denen Sie Ihren Simulationsstatus speichern können.

Verwalten Ihrer Apps und SDKs

Pflegen Sie Ihre Apps, lokale Installationen der AWS Software Development Kits (SDKs) und das SimSpace Weaver App-SDK. Sie können neue Versionen der AWS SDKs herunterladen und installieren. Testen Sie neue Versionen des SimSpace Weaver App-SDK mit App-Builds, die nicht zur Produktion gehören, 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 das Ändern Ihres Simulationsstatus und speichern Sie ihn (falls erforderlich).
3. Halten Sie Ihre Simulation an (einmal gestoppt, kann sie nicht neu gestartet werden).
4. Löschen Sie Ihre Simulation (gestoppte Simulationen, die nicht gelöscht wurden, werden auf Ihre Service-Limits angerechnet).
5. Erstellen Sie Ihre Simulation mit derselben Konfiguration und demselben aktualisierten App-Code neu.
6. Initialisieren Sie Ihre Simulation mit gespeicherten Statusdaten (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.

Protokollieren und Überwachen in SimSpace Weaver

Die Überwachung ist wesentlich zur Wahrung der Zuverlässigkeit, Verfügbarkeit und Leistung von SimSpace Weaver und Ihren anderen AWS-Lösungen. AWS bietet folgende Überwachungswerkzeuge, mit denen Sie SimSpace Weaver beobachten, Missstände melden und ggf. automatisch Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie ausgeführt werden, 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 [Amazon- CloudWatch Benutzerhandbuch](#).
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldaten von Ihren SimSpace Weaver Workern, und anderen Quellen überwachen, speichern CloudTrail und darauf zugreifen. - CloudWatch Protokolle können 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](#).

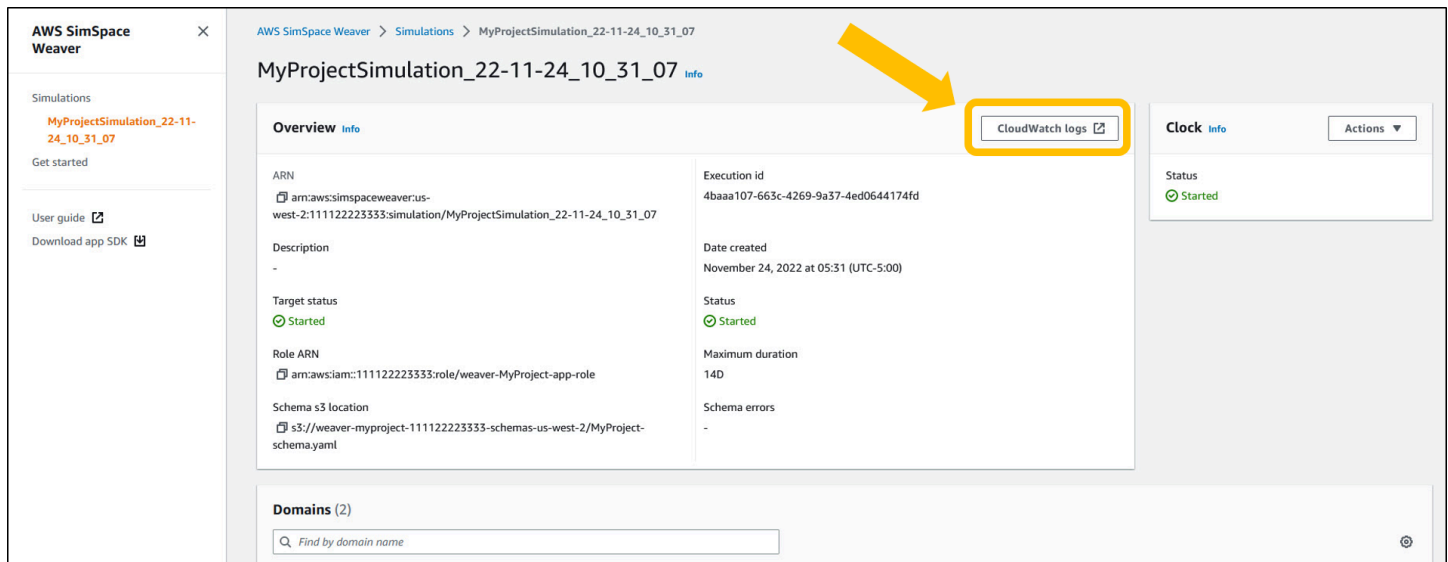
Themen

- [SimSpace Weaver -Protokolle in Amazon CloudWatch Logs](#)
- [Überwachung SimSpace Weaver mit Amazon CloudWatch](#)
- [Protokollieren von AWS SimSpace Weaver-API-Aufrufen mithilfe von AWS CloudTrail](#)

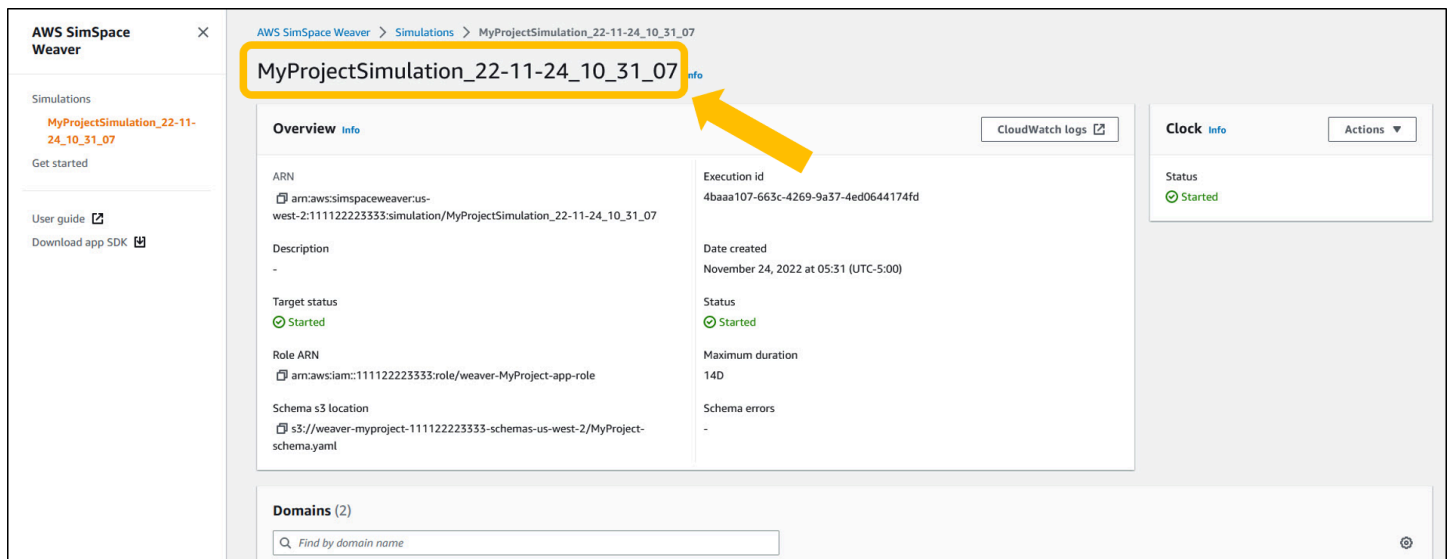
SimSpace Weaver -Protokolle in Amazon CloudWatch Logs

Zugreifen auf Ihre SimSpace Weaver Protokolle

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 WeaverKonsole verwenden, die Sie direkt zu den Protokollen für diese spezifische Simulation führt.



Sie können auf die Protokolle auch über die CloudWatch Konsole zugreifen. Sie benötigen den Namen Ihrer Simulation, um nach ihren Protokollen zu suchen.



SimSpace Weaver-Protokolle

SimSpace Weaver schreibt Simulationsverwaltungsnachrichten und die Konsolenausgabe aus Ihren Apps in Amazon CloudWatch Logs. Weitere Informationen zum Arbeiten mit Protokollen finden Sie unter [Arbeiten mit Protokollgruppen und Protokollstreams](#) im Amazon- CloudWatch Logs-Benutzerhandbuch.

Jede Simulation, die Sie erstellen, hat eine eigene Protokollgruppe in - CloudWatch Protokollen. Der Name der Protokollgruppe wird im Simulationsschema angegeben. Im folgenden Schemaausschnitt `log_destination_service` ist der Wert von `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 finden, nachdem Sie sie gestartet haben.

Important

Wenn Sie AWS IAM Identity Center oder benannte Profile für die AWS Command Line Interface (AWS CLI) verwenden, müssen Sie die SimSpace Weaver App-SDK-Version 1.12.1 oder höher verwenden. Die neueste Version ist 1.16.0. Informationen zu den SimSpace Weaver-Versionen erhalten Sie unter [SimSpace Weaver-Versionen](#). Die SimSpace Weaver App-SDK-Skripte verwenden die AWS CLI. Wenn Sie IAM Identity Center verwenden, können Sie entweder Ihr IAM-Identity-Center-Profil für AWS CLI in Ihr default Profil kopieren oder den Namen Ihres IAM-Identity-Center-Profiles mit dem `---profile cli-profile-name` Parameter an SimSpace Weaver App-SDK-Skripte weitergeben. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI zur Verwendung AWS IAM Identity Center](#) von im AWS Command Line Interface -Benutzerhandbuch und [unter Einstellungen für Konfigurations- und Anmeldeinformationsdateien](#) im AWS Command Line Interface - Benutzerhandbuch.

Docker

```
project-folder\tools\windows\weaver-project-name-cli.bat describe-simulation --  
simulation simulation-name
```

WSL

⚠ Important

Wir stellen Ihnen diese Anweisungen zur Verfügung. Sie sind für die Verwendung mit dem bestimmten Windows Subsystem for Linux (WSL) und werden nicht unterstützt. Weitere Informationen finden Sie unter [Einrichten der lokalen Umgebung für SimSpace Weaver](#).

```
project-folder/tools/linux/weaver-project-name-cli.sh describe-simulation --  
simulation simulation-name
```

Das folgende Beispiel zeigt den Teil der Ausgabe von DescribeSimulation, der die Protokollierungskonfiguration beschreibt. Der Name der Protokollgruppe wird am Ende der angezeigten LogGroupArn.

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

Jede Simulationsprotokollgruppe enthält mehrere Protokollstreams:

- Verwaltungsprotokollstream – Simulationsverwaltungsnachrichten, die vom SimSpace Weaver Service erstellt wurden.

```
/sim/management
```

- Protokollstream für Fehler – vom SimSpace Weaver Service 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
```

- Protokollstreams für räumliche Anwendungen (1 für jede räumliche App auf jedem Worker) – Konsolenausgabe, die von räumlichen Apps erstellt wird. Jede räumliche App schreibt in ihren eigenen Protokollstream. Der *spatial-app-id* steht für alle Zeichen nach dem abschließenden Schrägstrich am Ende des *worker-id*.

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

- Benutzerdefinierte App-Protokollstreams (1 für jede benutzerdefinierte App-Instance) – Konsolenausgabe, die von benutzerdefinierten Apps erstellt wird. Jede benutzerdefinierte App-Instance schreibt in ihren eigenen Protokollstream.

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

- Protokollstreams der Service-App (1 für jede Service-App-Instance) – Konsolenausgabe, die von Service-Apps erzeugt wird. Jede Service-App schreibt in einen eigenen Protokollstream. Die *service-app-id* besteht aus allen Zeichen nach dem abschließenden Schrägstrich am Ende der *service-app-name*.

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

Überwachung SimSpace Weaver mit Amazon CloudWatch

Sie können SimSpace Weaver mit Amazon überwachen CloudWatch, das Rohdaten sammelt und sie in lesbare Metriken verarbeitet, die nahezu in Echtzeit vorliegen. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen 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 [Amazon- CloudWatch Benutzerhandbuch](#).

Der SimSpace Weaver Service meldet die folgenden Metriken im -AWS/simspaceweaverNamespace.

SimSpace Weaver -Metriken auf Kontoebene

Der SimSpace Weaver Namespace enthält die folgenden Metriken im Zusammenhang mit Aktivitäten auf AWS Kontoebene.

Metrik	Beschreibung
SimulationCount	Die Anzahl der Simulationen für das aktuelle Konto. Einheiten: Anzahl Dimensionen: keine Statistiken: Durchschnitt, Minimum, Maximum

Protokollieren von AWS SimSpace Weaver-API-Aufrufen mithilfe von AWS CloudTrail

AWS SimSpace Weaver ist integriert, einem ServiceAWS CloudTrail, der die Aktionen eines Benutzers, einer Rolle oder eines -AWSServices in aufzeichnetSimSpace Weaver. CloudTrail erfasst alle API-Aufrufe für SimSpace Weaver als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der SimSpace Weaver-Konsole und Code-Aufrufe der SimSpace Weaver-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3-Bucket aktivieren, einschließlich Ereignissen für SimSpace Weaver. Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse in der CloudTrail Konsole in anzeigenEvent history. Anhand der von CloudTrailgesammelten Informationen können Sie die ange stellte AnfrageSimSpace Weaver, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrailfinden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

SimSpace Weaver -Informationen in CloudTrail

CloudTrail wird beim Erstellen des Kontos AWS-Konto auf Ihrem aktiviert. Wenn eine Aktivität in auftritt SimSpace Weaver, wird diese Aktivität in einem - CloudTrail Ereignis zusammen mit anderen - AWSServiceereignissen in aufgezeichnet Event history. Sie können in Ihrem AWS-Konto die neusten Ereignisse anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail Ereignisverlauf](#) .

Zur kontinuierlichen Aufzeichnung von Ereignissen in Ihrem AWS-Konto, einschließlich Ereignissen für SimSpace Weaver, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen Amazon S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere -AWS-Services konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Von unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien aus mehreren Konten](#)

Alle SimSpace Weaver Aktionen werden von protokolliert CloudTrail und sind in der [AWS SimSpace Weaver API-Referenz](#) zu dokumentiert. Aufrufe der DeleteSimulation Aktionen ListSimulations, DescribeSimulation und erzeugen beispielsweise Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Benutzeranmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer ausgeführt wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

Grundlagen zu SimSpace Weaver-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Bereitstellung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon 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, z. B. Datum und Uhrzeit der Aktion, Anforderungsparameter und andere details. CloudTrail log-Dateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

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 Endpunkte und Kontingente von Weaver

In den folgenden Tabellen werden die Service-Endpunkte und Service Quotas für SimSpace Weaver beschrieben. 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 (USA-West)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Service Quotas

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
Höchstdauer (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.
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 STARTED 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 Workern, die Sie einer Simulation zuweisen können. Sie können eine Erhöhung des Kontingents auf bis zu 10 beantragen.
vCPUs für jede Rechenressourceneinheit	Jede unterstützte Region: 2	Nein	Die Anzahl der virtuellen zentralen Verarbeitungseinheiten (vCPUs), die eine App für jede Rechenressourceneinheit erhält.

Messaging-Kontingente

Die folgenden Kontingente gelten für App-to-App-Nachrichten 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 Häkchen 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.

Service-Kontingente für SimSpace Weaver Local

Die folgenden Service-Kontingente gelten nur für SimSpace Weaver Local. Alle anderen Kontingente gelten auch für SimSpace Weaver Local.

Name	Standard	Anpassbar	Beschreibung
Maximale Partitionen	SimSpace Weaver Local: 24	Nein	Die maximale Anzahl von Partitionen für eine Simulation.
Maximale Anzahl von Apps	SimSpace Weaver Local: 24	Nein	Die maximale Gesamtzahl von Apps (jeder Art) für eine Simulation.
Maximale Domains	SimSpace Weaver Local: 24	Nein	Die maximale Gesamtzahl von Domains (jeder Art) für eine Simulation.
Entitäten pro Partition	SimSpace Weaver Local: 4.096	Nein	Die maximale Anzahl der Entitäten in jeder Partition.

Name	Standard	Anpassbar	Beschreibung
Felder pro Entität	SimSpace Weaver Local: 8	Nein	Die maximale Anzahl von Feldern für jede Entität.
Feldgröße	SimSpace Weaver Local: 1024 Byte	Nein	Die maximale Größe eines Entitätsfelds.

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 Amazon-Ressourcenname (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-KontoNummer: `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. Weitere Informationen finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#). Informationen dazu, wie Sie unnötige Simulationen bereinigen können, finden Sie [Schritt 6: Anhalten und Bereinigen Ihrer Simulation](#) im Schnellstart-Tutorial.

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. Anleitungen zur Bereinigung Ihrer Simulation finden Sie [Schritt 6: Anhalten und Bereinigen Ihrer Simulation](#) im Schnellstart-Tutorial.

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](#)
- 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

befindetAWS-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 Sie das App-SDK verwenden Docker und beim Ausführen von Skripts aus dem SimSpace Weaver App-SDK zeitbedingte Fehler auftreten, kann dies daran liegen, dass die Uhr Ihrer Docker virtuellen Maschine falsch ist. Dies kann passieren, wenn Ihr Computer lief Docker und dann aus dem Standbymodus oder Ruhezustand wieder aufgenommen wurde.

Lösungen zum Ausprobieren

- Starten Sie Docker neu.
- 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 Konsolenclient kann keine Verbindung herstellen

Möglicherweise wird vom Konsolenclient die folgende Fehlermeldung angezeigt, wenn Sie eine Verbindung zu der im [Schnellstart-Tutorial](#) und im [ausführlichen Tutorial](#) beschriebenen PathfindingSample Simulation herstellen. 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 der 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 [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#). Sie können auch die [SimSpace WeaverKonsole](#) 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 [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#). Sie können die [SimSpace WeaverKonsole](#) 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 [Schritt 4: Abrufen Ihrer IP-Adresse und Portnummer](#). Sie können die [SimSpace WeaverKonsole](#) 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 WeaverKonsole](#) 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 pingen. 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, haben Sie möglicherweise eine andere Software (z. B. eine lokale Firewall, Sicherheitseinstellungen oder Anti-Malware-Programme), die Ihre Verbindung blockiert.

- 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](#).
- 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](#).
- 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 Simulationsschemareferenz

SimSpace Weaver verwendet eine YAML-Datei, um die Eigenschaften einer Simulation zu konfigurieren. Diese Datei wird als Simulationsschema (oder nur als Schema) bezeichnet. Die im SimSpace Weaver App SDK enthaltene Beispielsimulation enthält ein Schema, das Sie für Ihre eigene Simulation kopieren und bearbeiten können.

Themen

- [Beispiele für ein vollständiges Schema](#)
- [Schemaformat](#)

Beispiele für ein vollständiges Schema

Das folgende Beispiel zeigt die YAML-formatierte Textdatei, die eine SimSpace Weaver Simulation beschreibt. Diese Beispiele enthalten Dummy-Werte für die Eigenschaften. Das Format der Datei variiert je nach dem Wert von , der darin `sdk_version` angegeben ist. Eine vollständige Beschreibung der Eigenschaften und ihrer gültigen Werte [Schemaformat](#) finden Sie unter .

```
sdk_version: "1.16"
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 ist unerheblich, solange die übergeordneten und untergeordneten Beziehungen identisch sind. Die Reihenfolge ist für Elemente in einem Array wichtig.

```
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](#)
- [Simulationseigenschaften](#)
- [Worker](#)
- [Uhr](#)
- [Partitionierungsstrategien](#)
- [Domains](#)
- [Platzierungseinschränkungen](#)

SDK-Version

Der `sdk_version` Abschnitt (erforderlich) identifiziert die Version des SimSpace Weaver App-SDK, das dieses Schema unterstützt. Gültige Werte: 1.16, 1.15, 1.14, 1.13, 1.12

Important

Der Wert von `sdk_version` enthält nur die Hauptversionsnummer und die erste Nebenversionsnummer. Der Wert 1.12 gibt beispielsweise alle Versionen an 1.12.x, z. B. 1.12.01, 1.12.1, und 1.12.2.

```
sdk_version: "1.16"
```

Simulationseigenschaften

Der `simulation_properties` Abschnitt (erforderlich) gibt verschiedene Eigenschaften Ihrer Simulation an. 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 SimSpace Weaver schreibt.

Erforderlich: Nein. Wenn diese Eigenschaft nicht enthalten ist, SimSpace Weaver schreibt keine Protokolle für die Simulation.

Typ: Zeichenfolge

Zulässige Werte:

- Der Name einer - CloudWatch Protokollgruppe (z. B. 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 die Simulation.

`log_destination_service`

Gibt den Typ der Protokollierungszielressource an, wenn Sie einen `logging_destination_resource_name` angeben, der kein ARN ist.

Erforderlich: Sie müssen diese Eigenschaft angeben, wenn angegeben `log_destination_resource_name` ist und kein ARN ist. Sie können diese Eigenschaft nicht angeben, wenn nicht angegeben `log_destination_resource_name` ist oder ein ARN ist.

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: `Vector3<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, verwenden Sie Domains, die keinen `default_image` angeben.

Required: No

Typ: Zeichenfolge

Gültige Werte:

- Der 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) legt Konfigurationen für Worker-Gruppen (Arbeitsgruppen) fest. SimSpace Weaver verwendet diese Informationen zusammen mit `placement_constraints` um die zugrunde liegende Infrastruktur Ihrer Simulation zu konfigurieren. Derzeit wird nur eine Worker-Gruppe unterstützt.

Um die Eigenschaften für eine Worker-Gruppe anzugeben, ersetzen Sie durch *worker-group-name* einen Namen Ihrer Wahl. Der Name muss 3–64 Zeichen lang sein und kann die Zeichen A-Z, -z, 0-9 und _ (Bindestrich) enthalten. Geben Sie die Eigenschaften der Worker-Gruppe nach dem Namen an.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Eigenschaften

type

Gibt den Worker-Typ an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: `sim.c5.24xlarge`

desired

Gibt die gewünschte Anzahl von Workern für diese Worker-Gruppe an.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1-3. Ihr Servicekontingent (Limit) für die Anzahl der Worker für Ihre Simulationen bestimmt den Höchstwert dieser Eigenschaft. Wenn Ihr Servicekontingent beispielsweise ist, ist 2 der Höchstwert für diese Eigenschaft 2. Sie können eine Erhöhung Ihres Servicekontingents beantragen. Weitere Informationen finden Sie unter [SimSpace Endpunkte und Kontingente von Weaver](#).

Uhr

Der `clock` Abschnitt (erforderlich) gibt die Eigenschaften der Simulationsuhr an.

```
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: Ganzzahl

Zulässige Werte:

- Version 1.14 und 1.15: "10" | "15" | "30" | "unlimited"
 - "unlimited": Die Uhr sendet das nächste Häkchen, sobald alle Apps ihre Commit-Operationen für das aktuelle Häkchen abgeschlossen haben.
- Version 1.13 und 1.12: 10 | 15 | 30

Partitionierungsstrategien

Der `partitioning_strategies` Abschnitt (erforderlich) gibt die Organisation von Partitionen für eine räumliche Domain an.

Note

SimSpace Weaver unterstützt nur eine Partitionierungsstrategie.

Um die Eigenschaften für eine Partitionierungsstrategie anzugeben, ersetzen Sie durch *partitioning-strategy-name* einen Namen Ihrer Wahl. Der Name muss 3–64 Zeichen lang sein und kann die Zeichen A-Z, -z, 0-9 und _ (Bindestrich) enthalten. Geben Sie die Eigenschaften der Partitionierungsstrategie nach dem Namen 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 Achsen ausgerichteten Hauptbegrenzungsrahmens (AABB) für Ihre Simulation an. Sie geben die Grenzen als 2-elementgeordnete Arrays an, die die Mindest- und Höchstwerte (in dieser Reihenfolge) für jede Achse (x und y) beschreiben.

Erforderlich: Bedingt. Diese Eigenschaft ist erforderlich (und kann nur angegeben werden), wenn die Topologie auf festgelegt 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 Domain), die räumlich nebeneinander liegen.

Erforderlich: Bedingt. Diese Eigenschaft ist erforderlich (und kann nur angegeben werden), wenn die Topologie auf festgelegt ist "Grid". Wenn Sie keine Konfiguration für Platzierungsgruppen angeben, SimSpace Weaver berechnet eine für Sie. Jede Domain, die eine Partitionierungsstrategie ohne Konfiguration von Platzierungsgruppen verwendet, muss eine angeben `grid_partition` (siehe [Strategie zur Partitionierung räumlicher Domains](#)).

Typ: Ganzzahl (für jede Achse)

Gültige Werte: 1-20. Wir empfehlen, dass $x * y$ der gewünschten Anzahl von Workern entspricht. Andernfalls SimSpace Weaver versucht, Ihre Platzierungsgruppen auf die verfügbaren Worker auszugleichen.

Domains

Der `domains` Abschnitt (erforderlich) gibt die Eigenschaften für jede Ihrer Domänen an. Alle Simulationen müssen mindestens einen Abschnitt für eine räumliche Domain haben. Sie können mehrere Abschnitte für zusätzliche Domains erstellen. Jeder Domärentyp hat sein eigenes Konfigurationsformat.

Important

Versionen 1.13 und unterstüzt nicht mehrere 1.12 räumliche Domains.

Important

SimSpace Weaver unterstüzt bis zu 5 Domains für jede Simulation. Dazu gehören alle räumlichen, benutzerdefinierten und Service-Domains.

```
domains:
```

```

domain-name:
  domain-configuration
domain-name:
  domain-configuration
...

```

Domänenkonfiguration

- [Räumliche Domänenkonfiguration](#)
- [Benutzerdefinierte Domänenkonfiguration](#)
- [Konfiguration der Service-Domain](#)

Räumliche Domänenkonfiguration

Um die Eigenschaften für eine räumliche Domain anzugeben, ersetzen Sie durch *spatial-domain-name* einen Namen Ihrer Wahl. Der Name muss 3–64 Zeichen lang sein und kann die Zeichen A-Z, -z, 0-9 und _ - (Bindestrich) enthalten. Geben Sie die Eigenschaften der räumlichen Domäne nach dem Namen 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 Domains

Der `launch_apps_by_partitioning_strategy` Abschnitt (erforderlich) gibt die Partitionierungsstrategie und Dimensionen (in Anzahl der Partitionen) des Simulationsbereichs an.

```

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 Domain an.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Der Wert dieser Eigenschaft muss mit dem Namen einer Partitionierungsstrategie übereinstimmen, die im `partitioning_strategies` Abschnitt definiert ist. Weitere Informationen finden Sie unter [Partitionierungsstrategien](#).

grid_partition

Gibt die Anzahl der Partitionen entlang jeder Achse (x und y) in einer Rastertopologie an. Diese Dimensionen beschreiben den gesamten Simulationsbereich für diese Domäne.

Erforderlich: Bedingt. Diese Eigenschaft kann nur angegeben werden, wenn die Topologie auf festgelegt ist "Grid". Diese Eigenschaft hängt von der `grid_placement_groups` Eigenschaft der angegebenen Partitionierungsstrategie für diese Domain ab:

- Diese Eigenschaft ist erforderlich, wenn die Partitionierungsstrategie dieser Domain keine `grid_placement_groups` Konfiguration angibt.
- Wenn es eine `grid_placement_groups` Konfiguration gibt, Sie aber nicht `angabengrid_partition`, SimSpace Weaver verwendet die gleichen Dimensionen wie die `grid_placment_groups` Konfiguration.
- Wenn Sie sowohl als auch `angabengrid_placement_groupsgrid_partition`, `grid_partition` müssen die Dimensionen von ein Vielfaches der Dimensionen von sein `grid_placement_groups` (wenn Ihre `grid_placement_groups` Dimensionen beispielsweise 2x2 sind, dann `grid_partition` sind einige gültige Dimensionen für 2x2, 4x4, 6x6, 8x8, 10x10).

Typ: Ganzzahl (für jede Achse)

Gültige Werte: 1-20

Räumliche App-Konfiguration

Der `app_config` Abschnitt (erforderlich) gibt das Paket, die Startkonfiguration und die Ressourcenanforderungen für Apps in dieser Domain an.

```
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 App-Datei/Binärdatei enthält. Das Paket muss in einem Amazon S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Der Amazon S3-URI des Pakets in einem Amazon S3-Bucket. Beispiel: `s3://example-bucket/MySpatialApp.zip`

`launch_command`

Gibt den ausführbaren/binären Dateinamen und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolgen-Token ist ein -Element im Array.

Erforderlich: Ja

Typ: Zeichenfolge-Array

`required_resource_units`

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instance dieser App zuweisen SimSpace Weaver soll. Eine Ressourceneinheit ist eine feste Menge an virtuellen zentralen Verarbeitungseinheiten (vCPUs) und Random-Access-Speicher (RAM) auf einem Worker. Weitere Informationen zu Ressourceneinheiten finden Sie unter [Endpunkte und Service Quotas](#). Die `-compute`Eigenschaft gibt eine Ressourceneinheitenzuweisung für die Worker-`compute`Familie an und ist derzeit der einzige gültige Zuweisungstyp.

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 (in Versionen 1.13 und nicht unterstützt 1.12). Stellen Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) bereit, das das Image enthält. Wenn diese Eigenschaft nicht angegeben ist, aber im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain die `default_image`. Weitere Informationen finden Sie unter [Benutzerdefinierte Container](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

image

Gibt den Speicherort eines Container-Images zum Ausführen von Apps in dieser Domain an.

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 Domänenkonfiguration

Um die Eigenschaften für eine benutzerdefinierte Domäne anzugeben, ersetzen Sie durch `custom-domain-name` einen Namen Ihrer Wahl. Der Name muss 3–64 Zeichen lang sein und kann die Zeichen A-Z, -z, 0-9 und _ (Bindestrich) enthalten. Geben Sie die Eigenschaften der benutzerdefinierten Domäne nach dem Namen 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 der StartApp-API zu starten.

Erforderlich: Ja

Typ: N/A

Gültige Werte: {}

Benutzerdefinierte App-Konfiguration

Der `app_config` section (erforderlich) gibt das Paket, die Startkonfiguration, die Ressourcenanforderungen und die Netzwerkports für Apps in dieser benutzerdefinierten Domain an.

```
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 App-Datei/Binärdatei enthält. Das Paket muss in einem Amazon S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Der Amazon S3-URI des Pakets in einem Amazon S3-Bucket. Beispiel: `s3://example-bucket/MyCustomApp.zip`

launch_command

Gibt den ausführbaren/binären Dateinamen und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolgen-Token ist ein -Element im Array.

Erforderlich: Ja

Typ: String-Array

required_resource_units

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instance dieser App zuweisen SimSpace Weaver soll. Eine Ressourceneinheit ist eine feste Menge an virtuellen zentralen Verarbeitungseinheiten (vCPUs) und Random-Access-Speicher (RAM) auf einem Worker. Weitere Informationen zu Ressourceneinheiten finden Sie unter [Endpunkte und Service Quotas](#). Die `-compute`Eigenschaft gibt eine Ressourceneinheitenzuweisung für die Worker-`compute`Familie an und ist derzeit der einzige gültige Zuweisungstyp.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1-4

endpoint_config

Gibt die Netzwerkendpunkte für Apps in dieser Domain an. Der Wert von `ingress_ports` gibt die Ports an, an die Ihre benutzerdefinierten Apps für eingehende Client-Verbindungen 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 finden.

Erforderlich: Nein. Wenn Sie keine Endpunktconfiguration angeben, haben Ihre benutzerdefinierten Apps in dieser Domain keine Netzwerkendpunkte.

Typ: Ganzzahl-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 (in Versionen 1.13 und nicht unterstützt 1.12). Stellen Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) bereit, das das Image enthält. Wenn diese Eigenschaft nicht angegeben ist, aber im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain die `default_image`. Weitere Informationen finden Sie unter [Benutzerdefinierte Container](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

image

Gibt den Speicherort eines Container-Images zum Ausführen von Apps in dieser Domain an.

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`)

Konfiguration der Service-Domain

Um die Eigenschaften für eine Service-Domain anzugeben, ersetzen Sie durch *service-domain-name* einen Namen Ihrer Wahl. Der Name muss 3–64 Zeichen lang sein und kann die Zeichen A-Z, -z, 0-9 und _ - (Bindestrich) enthalten. Geben Sie die Eigenschaften der Service-Domain nach dem Namen an. Wiederholen Sie diesen Vorgang für jede Service-Domain.

```
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 von Apps pro Worker

Der `launch_apps_per_worker` Abschnitt (erforderlich) gibt an, dass es sich um eine Service-Domain-Konfiguration 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

Zulässige Werte: {} | 1 | 2. Ein Wert von {} gibt den Standardwert von an1.

Service-App-Konfiguration

Der `app_config` section (erforderlich) gibt das Paket, die Startkonfiguration, die Ressourcenanforderungen und die Netzwerkports für Apps in dieser Service-Domain an.

```
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 App-Datei/Binärdatei enthält. Das Paket muss in einem Amazon S3-Bucket gespeichert werden. Es wird nur das ZIP-Dateiformat unterstützt.

Erforderlich: Ja

Typ: Zeichenfolge

Gültige Werte: Der Amazon S3-URI des Pakets in einem Amazon S3-Bucket. Beispiel: `s3://example-bucket/MyServiceApp.zip`

launch_command

Gibt den ausführbaren/binären Dateinamen und die Befehlszeilenparameter zum Starten der App an. Jedes Befehlszeilen-Zeichenfolgen-Token ist ein -Element im Array.

Erforderlich: Ja

Typ: String-Array

required_resource_units

Gibt die Anzahl der Ressourceneinheiten an, die jeder Instance dieser App zuweisen SimSpace Weaver soll. Eine Ressourceneinheit ist eine feste Menge an virtuellen zentralen Verarbeitungseinheiten (vCPUs) und Random-Access-Speicher (RAM) auf einem Worker. Weitere Informationen zu Ressourceneinheiten finden Sie unter [Endpunkte und Service Quotas](#). Die `-compute`Eigenschaft gibt eine Ressourceneinheitenzuweisung für die Worker-`compute`Familie an und ist derzeit der einzige gültige Zuweisungstyp.

Erforderlich: Ja

Typ: Ganzzahl

Gültige Werte: 1-4

endpoint_config

Gibt die Netzwerkendpunkte für Apps in dieser Domain an. Der Wert von `ingress_ports` gibt die Ports an, an die Ihre Service-Apps für eingehende Client-Verbindungen 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 finden.

Erforderlich: Nein. Wenn Sie keine Endpunktconfiguration angeben, haben Ihre Service-Apps in dieser Domain keine Netzwerkendpunkte.

Typ: Ganzzahl-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 (in Versionen 1.13 und nicht unterstützt 1.12). Stellen Sie den URI für ein Repository in Amazon Elastic Container Registry (Amazon ECR) bereit, das das Image enthält. Wenn diese Eigenschaft nicht angegeben ist, aber im `simulation_properties` Abschnitt der obersten Ebene angegeben `default_image` ist, verwenden Apps in dieser Domain die `default_image`. Weitere Informationen finden Sie unter [Benutzerdefinierte Container](#).

```
image: "ecr-repository-uri"
```

Eigenschaften

image

Gibt den Speicherort eines Container-Images zum Ausführen von Apps in dieser Domain an.

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`)

Platzierungseinschränkungen

Der `placement_constraints` Abschnitt (optional) gibt an, welche räumlichen Domänen auf denselben Workern zusammen platziert werden SimSpace Weaver sollen. Weitere Informationen finden Sie unter [Konfiguration räumlicher Domänen](#).

Important

Versionen 1.13 und unterstützen 1.12 nicht `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 werden SimSpace Weaver sollen.

Erforderlich: Ja

Typ: String-Array

Gültige Werte: Namen von Geodomänen, die im Schema angegeben sind

`on_workers`

Gibt die Worker-Gruppe an, in der die Domains platziert werden SimSpace Weaver sollen.

Erforderlich: Ja

Typ: Zeichenfolgen-Array mit 1 Element

Gültige Werte: Name einer Worker-Gruppe, die im Schema angegeben ist

SimSpace Weaver API-Referenzen

SimSpace Weaver verfügt über 2 verschiedene Sätze von Anwendungsprogrammierschnittstellen (APIs):

- Service-APIs – Diese APIs steuern die Service- und Servicere Ressourcen, wie Simulationen, Uhren und Apps. Sie sind Teil des AWS Hauptsoftware Development Kit (SDK) und Sie können sie mit der -AWSBefehlszeilenschnittstelle (CLI) aufrufen. Sie können auch das Convenience-Skript im Tools-Ordner für Ihr Projekt und Ihre Plattform verwenden (z. B. *project-folder* \tools \windows\weaver-*project-name* -cli.bat), um diese APIs aufzurufen. Weitere Informationen zu den Service-APIs finden Sie in der [SimSpace Weaver API-Referenz zu](#) .
- 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 SDK-Ordner für entpackte Apps: *sdk-folder* \SimSpaceWeaverAppSdk-1.16.0 \documentation

Note

sdk-folder ist der Ordner, in dem Sie das SimSpaceWeaverAppSdkDistributable Paket entpackt haben. Wenn *sdk-folder* \SimSpaceWeaverAppSdk-1.16.0 nicht vorhanden ist, stellen Sie sicher, dass Sie über Version 1.16.0 des SimSpace Weaver App-SDK verfügen. Das enthält SimSpaceWeaverAppSdkDistributable nicht das gesamte SimSpace Weaver App-SDK. Wenn Sie zum ersten Mal die SimSpace Weaver App-SDK-Skripts verwenden, um ein Projekt zu erstellen, erstellen die Skripts den Ordner SimSpaceWeaverAppSdk-1.16.0 in Ihrem *sdk-Ordner* und laden den Rest des SimSpace Weaver App-SDK in ihn herunter.

AWS SimSpace Weaver-Versionen

Wir verbessern kontinuierlich AWS SimSpace Weaver. Sie müssen das neueste SimSpace Weaver Anwendungs-SDK herunterladen, wenn wir eine neue Version veröffentlichen, wenn Sie die Vorteile neuer Funktionen und Funktionsaktualisierungen nutzen möchten. Um eine vorhandene Simulation mit einer neueren Version auszuführen, müssen Sie möglicherweise ihr Schema und ihren Code aktualisieren und dann eine neue Instance der Simulation starten. Sie müssen kein Upgrade durchführen und können weiterhin vorhandene Simulationen mit früheren Versionen ausführen. Auf dieser Seite finden Sie Informationen zu den Unterschieden zwischen den Versionen. Derzeit werden alle Versionen unterstützt.

Important

Die neueste Version des [AWS SimSpace Weaver Benutzerhandbuchs](#) deckt nur die neueste Version des Services ab. Die Dokumentation für frühere Versionen finden Sie im [AWS SimSpace Weaver Guide Catalog](#), verfügbar auf der [Startseite für die Hauptdokumentation](#).

Aktuelle Version

Die neueste Version ist: 1.16.0

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 Unterprogramm in Ihrem heruntergeladenen *sdk-folder*. Das Unterverzeichnis, das die SDK-Bibliotheken enthält, hat einen Namen, der die SDK-Versionsnummer enthält: `SimSpaceWeaverAppSdk-sdk-version`. Beispielsweise befinden sich die Bibliotheken für Version 1.15.3 in `SimSpaceWeaverAppSdk-1.15.3`.

Sie finden die Version des verteilbaren SimSpace Weaver App-SDK-Pakets auch in der Textdatei `app_sdk_distributable_version.txt` in Ihrem *sdk-folder*.

Herunterladen der neuesten Version

Verwenden Sie einen der folgenden Links, um die neueste Version herunterzuladen.

- [Vervollständigen des verteilbaren App-SDK-Pakets](#)
- [Nur die App-SDK-Bibliotheken](#)

Sie können auch das vollständige verteilbare SimSpace Weaver App-SDK-Paket von der [SimSpace WeaverKonsole](#) in der heruntergeladenen AWS Management Console. Wählen Sie im Navigationsbereich App SDK heruntergeladen aus.

Warning

Verwenden Sie nicht die , AWS CLI um etwas herunterzuladen, das das verteilbare SimSpace Weaver App-SDK-Paket zu sein scheint. Verwenden Sie nur die Download-Links auf dieser Seite oder den Download-Link in der -Konsole. Jede andere Download-Methode oder jeder andere Speicherort wird nicht unterstützt und kann veraltet, falschen oder böartigen Code enthalten.

Fehlerbehebung bei Downloads des App-SDK

Wir verwenden Amazon CloudFront (CloudFront), um die App-SDK-ZIP-Dateien zu verteilen. Es kann einige der folgenden Situationen geben.

- 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. Nach 24 Stunden erneut herunterladen.
- Sie erhalten einen HTTP-Fehler 4xx oder 5xx über einen Download-Link
 - Versuchen Sie es nach 24 Stunden erneut. Wenn Sie denselben Fehler erhalten, verwenden Sie den Feedback-Link unten in der [SimSpace WeaverKonsole](#), um das Problem zu melden. Wählen Sie Problem melden als Feedback-Typ 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. Löschen Sie Ihren Browser-Cache und versuchen Sie es erneut. Stellen Sie sicher, dass Sie keine Firewall-Regeln haben, die die Download-URL blockieren könnten.
- Sie erhalten eine Fehlermeldung, wenn Sie versuchen, die Datei zu speichern

- Überprüfen Sie Ihre lokalen Dateisystemberechtigungen, um sicherzustellen, dass Sie über die richtigen Berechtigungen zum Speichern der Datei verfügen.
- Ihr Browser wird angezeigt 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 etwas die URL in Ihrem Browser nicht beeinträchtigt hat. Verwenden Sie den Link erneut.

Installieren der neuesten Version

Installieren der neuesten Version

1. [Laden Sie die neueste Version](#) herunter.
2. Entpacken Sie die SimSpaceWeaverAppSdkDistributableZIP-Datei in einen Ordner.
3. Führen Sie `docker-create-image.bat` (oder `docker-create-image.sh` für WSL) im entpackten SDK-Ordner der neuesten Versions-SimSpace WeaverApp aus.
4. Verwenden Sie den entpackten SDK-Ordner für die neueste Version der SimSpace Weaver App anstelle der vorherigen Version.

Serviceversionen

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.16.0	<p>Neues Feature:</p> <ul style="list-style-type: none"> • Sie können jetzt Messaging-APIs im SimSpace Weaver App-SDK verwenden, um Nachrichten zwischen Ihren Apps zu senden und zu empfangen. 	12. Februar 2024	Dieses Handbuch	<ul style="list-style-type: none"> • Vollständiges Paket • Nur Bibliotheken <p>Weitere Informationen finden Sie unter Fehlerbehebung.</p>

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<p>Diese Funktion ist für C++-, Python- und Unity- und Unreal-Engine-5-Integrationen verfügbar. Weitere Informationen finden Sie unter Messaging.</p>			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.15.3	<p>SimSpace Weaver Local Aktualisierung:</p> <ul style="list-style-type: none">Wir haben SimSpace Weaver Local stärker an die Entwicklung für die angepasste AWS Cloud. Diese Änderungen wirken sich auf Projekte und Workflows von C++, Python, Unity und Unreal Engine für aus SimSpace Weaver Local. Weitere Informationen finden Sie unter Lokale Entwicklungsunterschiede in Version 1.15.3.	4. Dezember 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.15.2	<p>Aktualisierung des verteilbaren App-SDK-Pakets:</p> <ul style="list-style-type: none"> Wir haben die aktualisierte <code>Dockerfile</code>, um die spezifische erforderliche Version von <code>cmake</code> zu verwenden. Docker-Container-Builds können ohne diese Änderung fehlschlagen. 	2. November 2022	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar
1.15.1	<p>Feature-Update:</p> <ul style="list-style-type: none"> Python SDK: Diese Version behebt ein Problem, das dazu führte, dass Python-basierte Simulationen in der <code>aws cloud</code> fehlschlagen. Weitere Informationen finden Sie in den Versionshinweisen zu. 	22. September 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.15.0	<p>Neues Feature:</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 seinen Python-Ansichtsclient. Weitere Informationen finden Sie unter Verwenden von Python.	31. August 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
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 zum Ausführen Ihrer SimSpace Weaver Apps in der AWS Cloud. Weitere Informationen finden Sie unter Benutzerdefinierte Container.• Mehrere räumliche Domänen: Erstellen Sie mehr als eine räumliche Domäne in einer	26. Juli 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<p>Simulation.</p> <p>Trennen Sie die Simulationslogik, anstatt sie in einer einzigen räumlichen App zu kombinieren. Weisen Sie Geo-Domains basierend auf ihren Anforderungen verschiedene Ressourcen zu. Weitere Informationen finden Sie unter Konfiguration räumlicher Domänen.</p> <ul style="list-style-type: none">• Unbegrenzte Tickrate: Aktivieren Sie Ihre Simulation so schnell, wie Ihr Code ausgeführt werden kann. Legen Sie die Uhr Ihrer Simulation so fest, dass das nächste Häkchen gesendet wird,			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<p>sobald alle Apps ihre Commit-Operationen für das aktuelle Häkchen abgeschlossen haben. Weitere Informationen finden Sie unter Uhr.</p> <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. Weitere Informationen finden Sie unter Uhr.			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.13.1	<p>SimSpace Weaver -App-SDK:</p> <ul style="list-style-type: none">• Feature-Update: Die Projekterstellung funktioniert jetzt korrekt mit der Pathfindi ngSampleU nreal Vorlage.	7. Juni 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.13.0	<p>SimSpace Weaver -Service-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 SnapshotS3Location Parameter hinzugefügt, um mit einem Snapshot zu beginnen. • Der SchemaS3Location Parameter ist jetzt optional. • Änderungen an der DescribeSimulation Ausgabe: 	29. April 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<ul style="list-style-type: none"> • SchemaError wird nicht mehr unterstützt. • StartError Feld hinzugefügt. • SnapshotS3Location Feld hinzugefügt. • SNAPSHOT_IN_PROGRESS Simulationsstatus hinzugefügt. • Neuer S3Destination Datentyp <p>SimSpace Weaver-Konsole</p> <ul style="list-style-type: none"> • Neue Funktionalität zum Erstellen von Snapshots. • Neue Funktionalität zum Starten einer Simulatio 			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<p>aus einem Snapshot.</p> <p>SimSpace Weaver -App-SDK:</p> <ul style="list-style-type: none"> • Neue Skripts zur Unterstützung von Snapshots • <code>create-snapshot- <i>project-name</i> .bat</code> • <code>start-from-snapshot- <i>project-name</i> .bat</code> • <code>quick-start-from-snapshot- <i>project-name</i> -cli.bat</code> • <code>list-snapshots- <i>project-name</i> .bat</code> • Projekte verwenden jetzt einen einzelnen Amazon S3-Bucket pro Projekt: <code>weaver-<i>lowercase-project-name</i> -account-</code> 			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	<p><i>number</i> <i>-region</i></p> <p>Weitere Informationen zu -Snapshots finden Sie unter Snapshots.</p>			

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.12.3	<p>SimSpace Weaver -App-SDK:</p> <ul style="list-style-type: none"> Die folgenden Skripts unterstützen jetzt den <code>---maximum-duration</code> Parameter: <code>quick-start-<i>project-name</i> -cli.bat</code> <code>quick-start-<i>project-name</i> -cli.sh</code> <code>start-simulation-<i>project-name</i> .bat</code> <code>start-simulation-<i>project-name</i> .sh</code> <code>run-<i>project-name</i> .bat</code> <code>run-<i>project-name</i> .sh</code> <p>Weitere Informationen finden Sie unter Maximale</p>	27. März 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
	Dauer einer Simulation.			
1.12.2	SimSpace Weaver -App-SDK: <ul style="list-style-type: none">• Fehlerbehebung: wird <code>docker-create-image.bat</code> jetzt korrekt ausgeführt.	1. März 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog.	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.12.1	<p>SimSpace Weaver -App-SDK:</p> <ul style="list-style-type: none"> Die Skripts akzeptieren jetzt ein -AWS CLIProfil, das für die AWS Authentifizierung verwendet werden soll. Die Skripts unterstützen jetzt AWS IAM Identity Center für die AWS Authentifizierung. <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none"> Fehlerbehebung: gibt <code>Api::BeginUpdateWillBlock</code> jetzt korrekt zurück <code>true</code>, wenn nicht alle Geo-Apps an der Simulation angeschlossen sind. 	28. Februar 2023	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

Version	Hinweise	Datum der Veröffentlichung	Dokumentation	App-SDK-Download
1.12.0	Veröffentlichung für allgemeine Verfügbarkeit (GA)	29. November 2022	Weitere Informationen finden Sie im AWS SimSpace Weaver -Handbuch katalog .	Nicht zum Herunterladen verfügbar

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 wirdAWS 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.

Dokumentverlauf für AWS SimSpace Weaver

In der folgenden Tabelle werden wichtige Änderungen an der SimSpace Weaver-Dokumentation beschrieben.

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
12. Februar 2024	Aktualisierter Inhalt	Das AWS SimSpace Weaver-Versionen Kapitel für die Version 1.16.0 wurde aktualisiert.	N/A
12. Februar 2024	Neuer Inhalt	Der Messaging Abschnitt wurde als Teil der Version 1.16.0 hinzugefügt. In diesem Abschnitt werden die Messaging-APIs beschrieben, die dem SimSpace Weaver App-SDK hinzugefügt wurden. Sie können diese APIs verwenden, um Nachrichten zwischen Ihren Apps zu senden und zu empfangen.	N/A
12. Februar 2024	Aktualisierter Inhalt	Das SimSpace Weaver Simulationsschemareferenz Kapitel für Version 1.16.0 wurde aktualisiert.	N/A
12. Februar 2024	Aktualisierter Inhalt	Dem SimSpace Endpunkte und Kontingente von Weaver Kapitel wurden Service Quotas	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		für Messaging hinzugefügt.	
12. Februar 2024	Neue Leitfäden	Teilen Sie den Inhalt für Versionen vor 1.16.0 in einen separaten Leitfaden auf. Der AWS SimSpace Weaver Guide Catalog (verfügbar auf der Startseite für die Hauptdokumentation) wurde zu Zugriffshandbüchern für frühere Versionen hinzugefügt.	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 wurde aktualisiert Service-Kontingente für SimSpace Weaver Local .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
4. Dezember 2023	Neuer und aktualisierter Inhalt	Der Lokale Entwicklung Abschnitt wurde umstrukturiert und eine neue Seite hinzugefügt, auf der die Unterschiede für beschrieben werden, die in Version 1.15.3 SimSpace Weaver Local eingeführt wurden.	N/A
7. November	Aktualisierter Inhalt	Die Anweisungen zum Einrichten von Docker und WSL zur Verwendung des direkten Download-Links/URL für das App-SDK wurden aktualisiert. Weitere Informationen finden Sie unter Einrichten der lokalen Umgebung für SimSpace Weaver .	N/A
2. November	Aktualisierter Inhalt	Die Seite mit den Serviceversionen für die Version 1.15.2 wurde aktualisiert. Weitere Informationen finden Sie unter Serviceversionen .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
23. Oktober 2	Aktualisierter Inhalt	Die Seite mit den Serviceversionen wurde mit neuen Anweisungen aktualisiert, um das verteilbare App-SDK-Paket herunterzuladen. Kunden sollten jetzt nur einen unserer genehmigten Links zum direkten Download und nicht die verwenden AWS CLI, um das verteilbare App-SDK-Paket herunterzuladen. Weitere Informationen finden Sie unter Herunterladen der neuesten Version .	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 aktualisiert
10. September 2023	Neuer Inhalt	Es wurde ein Abschnitt zur Fehlerbehebung für Situationen hinzugefügt, in denen die AWS CLI nicht erkennt SimSpace Weaver. Weitere Informationen finden Sie unter Das AWS CLI erkennt nicht simspaceweaver .	N/A
10. September 2023	Aktualisierter Inhalt	Die Installationsanweisungen für die AWS CLI in WSL wurden aktualisiert. Weitere Informationen finden Sie unter Einrichten Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL) .	N/A
07. September	API-Update	BucketName und ObjectKey sind jetzt für den Datentyp S3Location erforderlich. BucketName ist jetzt für den Datentyp S3Destination erforderlich.	AWS SDK: 2023-09-07
31. August 2023	Neuer Inhalt	Es wurde ein neuer Abschnitt für Version 1.15.0 hinzugefügt: Verwenden von Python .	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
15. August 2023	Aktualisierter Inhalt	Die Download-Anweisungen in AWS SimSpace Weaver-Versionen wurden aktualisiert, um nur offizielle SimSpace Weaver Amazon S3-Buckets aufzulisten. Andere Download-Speicherorte werden nicht von kontrolliert AWS und können schädlichen Code enthalten.	N/A
26. Juli 2023	Aktualisierter Inhalt	Aktualisiert Uhr .	N/A
26. Juli 2023	Aktualisierter Inhalt	Aktualisiert Konfiguration räumlicher Domänen .	N/A
26. Juli 2023	Neuer Inhalt	Neuer Abschnitt: Benutzerdefinierte Container .	N/A
26. Juli 2023	Aktualisierter Inhalt	Aktualisiert AWS SimSpace Weaver-Versionen für Version 1.14.0.	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 aktualisiert
7. Juni 2023	Aktualisierter Inhalt	Aktualisiert AWS SimSpace Weaver-Versionen für Version 1.13.1.	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 Notizen hinzugefügt, die den <code>--maximum-duration</code> Parameter zu SimSpace Weaver App-SDK-Skripten hinzugefügt haben.	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
9. März 2023	Geänderte Inhalte	Es wurde klargestellt, dass wir nur Anweisungen für Docker auf Windows und für bereitgestellten Windows Subsystem for Linux (WSL) und dass WSL (und jede andere Linux-Umgebung) nicht unterstützt wird.	N/A
28. Februar 2023	Neuer Inhalt	Es wurde ein Kapitel hinzugefügt, das SimSpace Weaver Versionen beschreibt.	N/A
28. Februar 2023	Geänderte Inhalte	Der Inhalt zur Authentifizierung wurde geändert, um die Verwendung von AWS IAM Identity Center und benannten Profilen für die AWS Command Line Interface (AWS CLI) einzuschließen.	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	Anweisungen zum Debuggen lokaler Simulationen hinzugefügt.	N/A

Datum	Änderung	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
29. November	Servicestart	Das -Benutzerhandbuch und die API-Referenz für wurden veröffentlicht SimSpace Weaver.	AWS SDK: 2022-11-29

Glossar

Dieses Glossar definiert Begriffe, die spezifisch für sind. AWS SimSpace Weaver

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der allgemeinen AWS-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 integrierenSimSpace Weaver. Das SDK bietet APIs zum Lesen und Schreiben von Entitätsdaten und zum Verfolgen 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 an 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 Endpunkte und Kontingente von Weaver .
Takttickrate	Siehe Taktrate .
Rechenressourceneinheit	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 den SimSpace Weaver APIs finden 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.

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.