



Guida dell'utente per la versione 1.16.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Guida dell'utente per la versione 1.16.0

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è SimSpace Weaver?	1
Concetti chiave	1
Funzionamento di SimSpace Weaver	2
Come si usa SimSpace Weaver	5
Schema di simulazione	6
Lavoratori e unità di risorse	6
Orologio di simulazione	6
Partizioni	7
Tessuto di stato	7
Entità	8
App	8
Casi d'uso di esempio	11
Configurazione	12
Configura il tuo account	12
Registrarsi per creare un Account AWS	12
Creazione di un utente amministratore	13
Aggiungi le autorizzazioni da utilizzare SimSpace Weaver	14
Configurazione dell'ambiente locale	15
AL2 in Docker	16
AL2 in WSL	17
Utilizzo di software concesso in licenza	20
Nozioni di base	22
Guida	22
Fase 1: creazione di un progetto	23
Fase 2: Attivare la registrazione	26
Passaggio 3: Esegui lo script di avvio rapido	28
Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta	31
Fase 5: Visualizza la simulazione	38
Passaggio 6: interrompi e ripulisci la simulazione	45
Tutorial dettagliato	52
Fase 1: creazione di un progetto	52
Fase 2: Attivare la registrazione	56
Fase 3: Carica lo schema di simulazione	58
Fase 4: Costruire il progetto	60

Passaggio 5: caricare app	61
Fase 6: Inizia la simulazione	64
Fase 7: Ottieni i dettagli della simulazione	68
Passaggio 8: Avvia app personalizzate	74
Passaggio 9: avviare l'orologio	78
Fase 10: Controllare i registri	80
Fase 11: Visualizza la simulazione	83
Passaggio 12: interrompi e ripulisci la simulazione	91
Utilizzo di SimSpace Weaver	98
Configurazione della simulazione	98
Parametri di configurazione della simulazione	100
Versione SDK	101
Proprietà di simulazione	101
Worker	102
Orologio	103
Strategie di partizionamento	106
Domini	107
Durata massima	117
Valore massimo	117
Valore predefinito	117
Valore minimo	117
Avvio di una simulazione utilizzando gli script SDKSimSpace Weaver dell'app	117
Avvio di una simulazione utilizzando la console	118
Lo stato di una simulazione che raggiunge la sua durata massima	118
Sviluppo di app	118
App spaziali	119
App personalizzate	120
Sviluppo di applicazioni client	121
Sviluppo locale	122
Creazione	123
Esecuzione	124
Vista	126
Interrompi	127
Esegui il debug	127
Differenze nella versione 1.15.3	128
SimSpace Weaverapp SDK	134

I metodi API restituiscono un Result	135
Interazione con l'SDK dell'app al livello più alto	135
Gestione della simulazione	136
Sottoscrizioni	139
Entità	140
Eventi dell'entità	152
Resulte gestione degli errori	159
Generici e tipi di dominio	161
Operazioni varie dell'SDK dell'app	161
SimSpace Weaver struttura dimostrativa	164
Utilizzo delle quote	165
Scopri i limiti di un'app	165
Ottieni la quantità di risorse utilizzate da un'app	166
Reimpostazione delle metriche	167
Superamento di un limite	167
In esaurimento della memoria	168
Best practice	168
Simulazioni di debug	168
Usa SimSpace Weaver Local e guarda l'output della console	169
Guarda i tuoi log in Amazon CloudWatch Logs	169
Usa le chiamate describe API	169
Connect a client	170
Eseguire il debug di simulazioni locali	171
Contenitori personalizzati	171
Crea un contenitore personalizzato	172
Modifica un progetto per utilizzare un contenitore personalizzato	173
Domande frequenti	176
Risoluzione dei problemi	177
Uso di Python	177
Creare un progetto Python	178
Avvio di una simulazione Python	181
Il client Python di esempio	181
Scrivere i propri script di build	182
Domande frequenti	183
Risoluzione dei problemi	183
Support per altri motori	185

Unità	185
Unreal Engine	186
Utilizzo di software concesso in licenza	187
Gestire le risorse con AWS CloudFormation	187
Snapshot	189
Snapshot	190
Casi d'uso per uno snapshot	191
SimSpace Weaverapp SDK	192
Console SimSpace Weaver	196
AWS CLI	198
API SimSpace Weaver	201
Domande frequenti	201
Messaggistica	202
Casi d'uso per la messaggistica	203
Utilizzo delle API di messaggistica	203
Quando usare la messaggistica	211
Suggerimenti per l'utilizzo della messaggistica	214
Errori di messaggistica e risoluzione dei problemi	216
Best practice	218
Imposta gli allarmi di fatturazione	218
Utilizzare SimSpace Weaver Local	218
Interrompi le simulazioni che non ti servono	219
Elimina le risorse che non ti servono	219
Avere dei backup	219
Sicurezza	220
Protezione dei dati	221
Crittografia dei dati a riposo	222
Crittografia in transito	222
Riservatezza del traffico Internet	223
Identity and Access Management	223
Destinatari	223
Autenticazione con identità	224
Gestione dell'accesso con policy	228
Funzionamento di AWS SimSpace Weavercon IAM	230
Esempi di policy basate su identità	237
Autorizzazioni create per te SimSpace Weaver	242

Prevenzione del problema "confused deputy" tra servizi	244
Risoluzione dei problemi	246
Registrazione e monitoraggio degli eventi di sicurezza	249
Convalida della conformità	250
Resilienza	251
Sicurezza dell'infrastruttura	252
Modello di sicurezza della connettività di rete	252
Analisi della configurazione e delle vulnerabilità	253
Best practice di sicurezza	254
Crittografa le comunicazioni tra le tue app e i relativi client	254
Effettua periodicamente il backup dello stato della simulazione	254
Gestisci le tue app e gli SDK	254
Registrazione e monitoraggio	256
Effettua il login CloudWatch	256
Accedere ai tuoi registri SimSpace Weaver	256
Log SimSpace Weaver	257
Monitoraggio con CloudWatch	260
SimSpace Weaver metriche a livello di account	260
CloudTrail registri	261
SimSpace Weaver informazioni in CloudTrail	261
Comprensione delle voci dei file di log di SimSpace Weaver	262
Endpoint e quote di servizio	264
Endpoint del servizio	264
Quote del servizio	265
Quote di messaggistica	267
Frequenze di clock	268
Service Quotas per SimSpace Weaver Local	268
Risoluzione dei problemi	270
AssumeRoleAccessDenied	270
InvalidBucketName	272
ServiceQuotaExceededException	273
TooManyBuckets	273
Autorizzazione negata durante l'avvio della simulazione	274
Problemi legati al tempo di utilizzo Docker	274
Il client della console non riesce a connettersi	275
No simspaceweaver nel AWS CLI	277

riferimento allo schema	278
Esempi di uno schema completo	278
Formato dello schema	280
Versione SDK	281
Proprietà di simulazione	281
Worker	283
Orologio	284
Strategie di partizionamento	284
Domini	286
Vincoli di posizionamento	296
Riferimenti API	298
Versioni SimSpace Weaver	299
Versione più recente	299
Come trovare la tua versione attuale	299
Scarica la versione più recente	299
Risoluzione dei problemi relativi ai download dell'SDK delle app	300
Installa la versione più recente	301
Versioni del servizio	301
1.15.1	317
Aggiorna un progetto Python esistente a 1.15.1	317
Risoluzione dei problemi per la versione 1.15.1	318
Domande frequenti sulla versione 1.15.1	318
Cronologia dei documenti	319
Glossario	326
.....	cccxxi

Cos'è AWS SimSpace Weaver?

AWS SimSpace Weaver è un servizio che puoi utilizzare per creare ed eseguire simulazioni spaziali in larga scala in Cloud AWS. Ad esempio, puoi creare simulazioni di folla, grandi ambienti del mondo reale ed esperienze coinvolgenti e interattive.

Con SimSpace Weaver, puoi distribuire i carichi di lavoro di simulazione in istanze Amazon Elastic Compute Cloud (Amazon EC2). SimSpace Weaver implementa l'AWS infrastruttura sottostante per te e gestisce la gestione dei dati di simulazione e la comunicazione di rete tra le istanze Amazon EC2 che eseguono la simulazione.

Concetti chiave per SimSpace Weaver

Una simulazione o un gioco è limitato dal computer che lo esegue. Man mano che cresci le dimensioni e la complessità del tuo mondo virtuale, le prestazioni di elaborazione iniziano a peggiorare. I calcoli richiedono più tempo, i sistemi esauriscono la memoria e i frame rate dei client diminuiscono. Per le simulazioni che non richiedono prestazioni in tempo reale, questo potrebbe essere solo fastidioso. Oppure, potrebbe trattarsi di una situazione aziendale critica in cui l'aumento dei ritardi di elaborazione comporta un aumento dei costi. Se la tua simulazione o il tuo gioco richiedono prestazioni in tempo reale, il degrado delle prestazioni è sicuramente un problema.

Una soluzione comune per una simulazione che raggiunge un limite di prestazioni consiste nel semplificare la simulazione. I giochi online con molti utenti spesso risolvono i problemi di scalabilità creando copie del loro mondo virtuale su server diversi e distribuendo gli utenti su di essi.

SimSpace Weaver risolve il problema della scalabilità dividendo il mondo virtuale spazialmente e distribuendo i pezzi su un cluster di istanze di calcolo eseguite nel Cloud AWS. Le istanze di calcolo lavorano insieme per elaborare l'intero mondo della simulazione in parallel. Il tuo mondo di simulazione appare come un unico spazio integrato per tutto ciò che contiene e per tutti i client che vi si connettono. Non è più necessario semplificare una simulazione a causa di un limite di prestazioni hardware. Puoi invece aggiungere più capacità di calcolo nel cloud.

Argomenti

- [Funzionamento di SimSpace Weaver](#)
- [Come si usa SimSpace Weaver](#)
- [Schema di simulazione](#)

- [Lavoratori e unità di risorse](#)
- [Orologio di simulazione](#)
- [Partizioni](#)
- [Tessuto di stato](#)
- [Entità](#)
- [App](#)

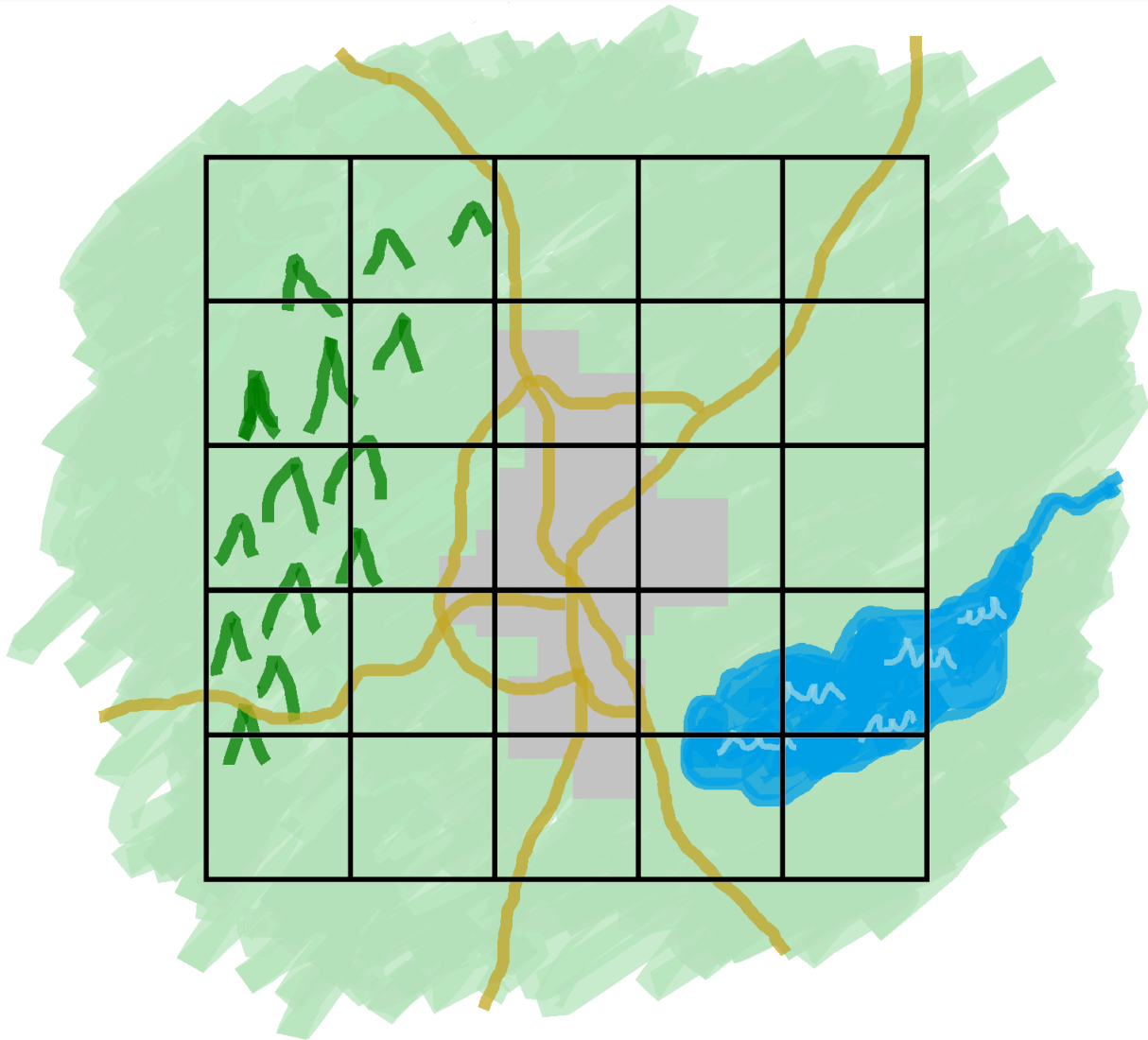
Funzionamento di SimSpace Weaver

La tua simulazione consiste in un mondo con oggetti al suo interno. Alcuni oggetti (come persone e veicoli) si muovono e fanno cose. Altri oggetti (come alberi ed edifici) sono statici. In SimSpace Weaver, un'entità è un oggetto nel tuo mondo di simulazione.

Definisci i confini del tuo mondo di simulazione e lo dividi in una griglia. Invece di creare una logica di simulazione che funzioni sull'intera griglia, si crea una logica di simulazione che opera su una cella della griglia. In SimSpace Weaver, un'app spaziale è un programma che scrivi che implementa la logica di simulazione per una cella della tua griglia. Ciò include la logica per tutte le entità in quella cella. L'area di proprietà di un'app spaziale è la cella della griglia controllata dall'app spaziale.

Note

Nel SimSpace Weaver, il termine «app» può riferirsi al codice di un'app o a un'istanza in esecuzione di quel codice.



Il tuo mondo di simulazione diviso in una griglia

Dividi il tuo mondo di simulazione in una griglia. Ogni app spaziale implementa la logica di simulazione per una singola cella in quella griglia.

SimSpace Weaver esegue un'istanza del codice spaziale dell'app per ogni cella della griglia. Tutte le istanze spaziali dell'app vengono eseguite in parallelo. In sostanza, SimSpace Weaver divide la simulazione complessiva in più simulazioni più piccole. Ciascuna delle simulazioni più piccole gestisce una parte del mondo complessivo della simulazione. SimSpace Weaver è in grado di distribuire ed eseguire queste simulazioni di dimensioni ridotte su più istanze Amazon Elastic

Compute Cloud (Amazon EC2) (denominati worker) nelCloud AWS. Un singolo lavoratore può eseguire più app spaziali.

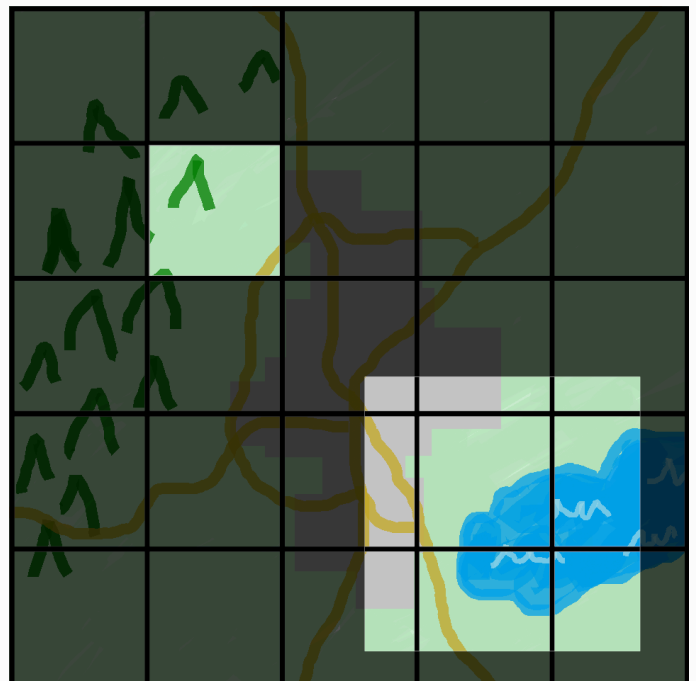
Le entità possono muoversi nel mondo della simulazione. Se un'entità entra nell'area di proprietà di un'altra app spaziale (un'altra cella nella griglia), il proprietario dell'app spaziale della nuova area assume il controllo dell'entità. Se la simulazione viene eseguita su più lavoratori, un'entità potrebbe passare dal controllo di un'app spaziale su un lavoratore a un'app spaziale su un lavoratore diverso. Quando un'entità passa a un altro lavoratore, SimSpace Weaver gestisce la comunicazione di rete sottostante.

Sottoscrizioni

La visione del mondo di un'app spaziale è un'area di proprietà a sé stante. Per scoprire cosa succede in un'altra parte del mondo della simulazione, l'app spaziale crea un abbonamento. L'area abbonamenti è un sottoinsieme dell'area globale della simulazione. Un'area di abbonamento può includere parti di più aree di proprietà, inclusa l'area di proprietà dell'app spaziale. SimSpace Weaver notifica all'app spaziale tutti gli eventi dell'entità (ad esempio, ingresso, uscita, creazione, aggiornamento ed eliminazione) che si verificano all'interno dell'area di sottoscrizione.



La visione del mondo di un'app spaziale



Una vista spaziale dell'app con un'area di sottoscrizione aggiunta

La visione del mondo di un'app spaziale è la sua area di proprietà, che è una cella nella griglia mondiale.

Un'app spaziale utilizza un abbonamento per scoprire cosa sta succedendo in un'altra parte del mondo della simulazione. L'area di sottoscrizione può contenere più celle della griglia e parti di celle.

Ad esempio, un'app che simula le entità che interagiscono fisicamente potrebbe aver bisogno di conoscere le entità che si trovano appena oltre i confini spaziali della sua area di proprietà. A tale scopo, l'app può iscriversi alle aree che confinano con l'area di proprietà. Dopo aver creato l'abbonamento, l'app riceve notifiche sugli eventi delle entità in quelle aree e può leggere le entità. Un altro esempio è un veicolo autonomo che deve vedere tutte le entità a 200 metri di distanza indipendentemente dall'app che possiede l'area. L'app per il veicolo può creare un abbonamento con un filtro che funge da riquadro di delimitazione allineato agli assi(AABB) che copre l'area visibile.

È possibile creare una logica di simulazione che non sia responsabile della gestione degli aspetti spaziali della simulazione. Un'app personalizzata è un programma eseguibile che viene eseguito su un singolo operatore. Sei tu a controllare il ciclo di vita (avvio e arresto) di un'app personalizzata. I client di simulazione possono connettersi a un'app personalizzata per visualizzare o interagire con la simulazione. Puoi anche creare un'app di servizio che venga eseguita su ogni lavoratore. SimSpace Weaver avvia un'istanza dell'app di servizio su ogni operatore che esegue la simulazione.

Le app personalizzate e le app di servizio creano abbonamenti per conoscere gli eventi delle entità e leggere le entità. Queste app non hanno aree di proprietà perché non sono spaziali. L'utilizzo di un abbonamento è l'unico modo per scoprire cosa sta succedendo nel mondo della simulazione.

Come si usa SimSpace Weaver

Durante l'uso SimSpace Weaver, questi sono i passaggi principali da seguire:

1. Scrivi e crea C++ app che integrano l'SDK SimSpace Weaver dell'app.
 - a. Le tue app effettuano chiamate API per interagire con lo stato della simulazione.
2. Scrivi ai clienti che visualizzano e interagiscono con la tua simulazione tramite alcune app.
3. Configura la tua simulazione in un file di testo.
4. Carica i pacchetti di app e la configurazione della simulazione sul servizio.
5. Inizia la tua simulazione.
6. Avvia e interrompi le app personalizzate in base alle necessità.

7. Collega i clienti alle tue app personalizzate o di servizio per visualizzare o interagire con la simulazione.
8. Controlla i log di simulazione in AmazonCloudWatch Logs.
9. Arrestare la simulazione.
10. Pulizia della simulazione.

Schema di simulazione

Lo schema (o schema) di simulazione è un file YAML di testo in formato che contiene informazioni di configurazione per la simulazione. SimSpace Weaver usa il tuo schema quando avvia una simulazione. Il pacchetto distribuibile SimSpace Weaver dell'app SDK include uno schema per un progetto di esempio. Questo può essere utilizzato come punto di partenza per il proprio schema. Per ulteriori informazioni sullo schema di simulazione, consulta [SimSpace Weaver riferimento allo schema di simulazione](#).

Lavoratori e unità di risorse

Un worker è un'istanza Amazon EC2 che esegue la tua simulazione. Si specifica un tipo di lavoratore nello schema di simulazione. SimSpace Weaver associa il tipo di lavoratore a un tipo di istanza Amazon EC2 specifico utilizzato dal servizio. SimSpace Weaver avvia e arresta i tuoi dipendenti per te e gestisce la comunicazione di rete tra i lavoratori. SimSpace Weaver avvia un set di lavoratori per ogni simulazione. Diverse simulazioni utilizzano lavoratori diversi.

La capacità di elaborazione disponibile (processore e memoria) di un lavoratore è suddivisa in unità logiche chiamate unità di risorse di calcolo (o unità di risorsa). Un'unità di risorsa rappresenta una quantità fissa di capacità del processore e della memoria.

Note

In precedenza ci riferivamo a un'unità di risorsa di calcolo come a uno slot. Potresti ancora vedere questo termine precedente nella nostra documentazione.

Orologio di simulazione

Ogni simulazione ha il suo orologio. Si avvia e si arresta l'orologio utilizzando le chiamate API o la SimSpace Weaver console. La simulazione si aggiorna solo quando l'orologio è in funzione. Tutte le

operazioni della simulazione avvengono all'interno di segmenti temporali chiamati zecche. L'orologio annuncia l'ora di inizio di ogni segno di spunta a tutti i lavoratori.

La frequenza di clock (o ticchettio) è il numero di ticchetti al secondo (hertz o Hz) che l'orologio annuncia. La frequenza di clock desiderata per una simulazione fa parte dello schema di simulazione. Tutte le operazioni relative a un segno di spunta devono essere completate prima dell'inizio del segno di spunta successivo. Per questo motivo, la frequenza di clock effettiva può essere inferiore alla frequenza di clock desiderata. La frequenza di clock effettiva non sarà superiore alla frequenza di clock desiderata.

Partizioni

Una partizione è un segmento della memoria condivisa di un lavoratore. Ogni partizione contiene parte dei dati sullo stato della simulazione.

Una partizione per un'app spaziale (chiamata anche partizione dell'app spaziale o partizione spaziale) contiene tutte le entità nell'area di proprietà di un'app spaziale. SimSpace Weaver inserisce le entità in partizioni spaziali dell'app in base alla posizione spaziale di ciascuna entità. Ciò significa che SimSpace Weaver tenta di posizionare sullo stesso lavoratore entità spazialmente vicine l'una all'altra. Ciò riduce al minimo la quantità di conoscenza che un'app richiede alle entità che non possiede per simulare le entità che possiede.

Tessuto di stato

Lo State Fabric è il sistema di memoria condivisa (la raccolta di tutte le partizioni) su tutti i lavoratori. Contiene tutti i dati di stato per la tua simulazione.

Lo State Fabric utilizza un formato binario personalizzato che descrive un'entità come un insieme di dati iniziali e un registro di aggiornamento, per ogni campo di dati di quell'entità. Con questo formato, puoi accedere allo stato di un'entità in un momento precedente del tempo di simulazione e mapparla a un punto nel tempo reale. Il buffer ha una dimensione finita e non è possibile tornare indietro nel tempo oltre a ciò che è nel buffer. SimSpace Weaver utilizza un puntatore all'offset corrente nel registro degli aggiornamenti per ogni campo e aggiorna un puntatore come parte di un aggiornamento del campo. SimSpace Weaver mappa questi log di aggiornamento nello spazio dei processi di un'app utilizzando la memoria condivisa.

Questo formato di oggetto comporta costi generali ridotti e nessun costo di serializzazione. SimSpace Weaver utilizza anche questo formato di oggetto per analizzare e identificare i campi dell'indice (come la posizione dell'entità).

Entità

Un'entità è il più piccolo elemento di dati nella simulazione. Esempi di entità includono attori (come persone e veicoli) e oggetti statici (come edifici e ostacoli). Le entità hanno proprietà (come posizione e orientamento) in cui è possibile memorizzare come dati persistenti SimSpace Weaver. Le entità esistono all'interno delle partizioni.

App

Un'App SimSpace Weaver è un software scritto da te che contiene una logica personalizzata che esegue ogni segno di simulazione. Lo scopo della maggior parte delle app è aggiornare le entità durante l'esecuzione della simulazione. Le tue app richiamano le API nell'SDK SimSpace Weaver dell'app per eseguire azioni (come lettura e aggiornamento) sulle entità della simulazione.

Puoi impacchettare le tue app e le relative risorse necessarie (come le librerie) come file.zip e caricarle su SimSpace Weaver. Un'app viene eseguita in un contenitore Docker su un lavoratore. SimSpace Weaver assegna a ciascuna app un numero fisso di unità di risorse sul lavoratore.

SimSpace Weaver assegna la proprietà di una (e solo una) partizione a ciascuna app. Un'app e la relativa partizione si trovano sullo stesso worker. Ogni partizione ha un solo proprietario dell'app. Un'app è in grado di creare, leggere, aggiornare ed eliminare le entità nella sua partizione. Un'app possiede tutte le entità nella sua partizione.

Esistono tre tipi di app: app spaziali, app personalizzate e app di servizio. Differiscono in base ai casi d'uso e ai cicli di vita.

Note

Nel SimSpace Weaver, il termine «app» può riferirsi al codice di un'app o di un'istanza in esecuzione di quel codice.

App spaziali

Le app spaziali aggiornano lo stato delle entità che esistono spazialmente nella simulazione. Ad esempio, potresti definire un'App `Physics` responsabile dello spostamento e della collisione delle entità per ogni segno di spunta in base alla loro velocità, forma e dimensione. In questo caso, SimSpace Weaver esegue più istanze dell'App `Physics` in parallel per gestire le dimensioni del carico di lavoro.

SimSpace Weaver gestisce il ciclo di vita delle app spaziali. Si specifica una disposizione delle partizioni spaziali delle app nello schema di simulazione. Quando avvia la simulazione, SimSpace Weaver avvia un'app spaziale per ogni partizione dell'app spaziale. Quando interrompi la simulazione, SimSpace Weaver disattiva le app spaziali.

Altri tipi di app possono creare entità, ma solo le app spaziali possono aggiornare le entità. Altri tipi di app devono trasferire le entità che creano in un dominio spaziale. SimSpace Weaver utilizza la posizione spaziale di un'entità per spostare l'entità nella partizione di un'app spaziale. Questo trasferisce la proprietà dell'entità all'app spaziale.

App personalizzate

Utilizzi app personalizzate per interagire con la tua simulazione. Un'app personalizzata legge i dati delle entità utilizzando gli abbonamenti. Un'app personalizzata può creare entità. Tuttavia, l'app deve trasferire un'entità in un'app spaziale per includerla nella simulazione e aggiornarla. È possibile che SimSpace Weaver assegni un endpoint di rete a un'app personalizzata. I client di simulazione possono connettersi all'endpoint di rete per interagire con la simulazione. Definisci le tue app personalizzate nel tuo schema di simulazione, ma sei responsabile di avviarle e interromperle (utilizzando le chiamate SimSpace Weaver API). Dopo aver avviato un'istanza di app personalizzata su un lavoratore, SimSpace Weaver non trasferisce l'istanza a un altro lavoratore.

App di servizio

Puoi utilizzare un'app di servizio quando hai bisogno di un processo di sola lettura in esecuzione su ogni lavoratore. Ad esempio, puoi utilizzare un'app di servizio se hai una simulazione di grandi dimensioni e hai bisogno di un client di visualizzazione che si muova attraverso la simulazione e mostri all'utente solo le entità visibili. In questo caso, una singola istanza dell'app personalizzata non può elaborare tutte le entità della simulazione. Puoi configurare un'app di servizio da avviare su ogni lavoratore. Ciascuna di queste app di servizio può quindi filtrare le entità sul lavoratore assegnato e inviare solo le entità pertinenti ai client connessi. Il client di visualizzazione può quindi connettersi a diverse app di servizio mentre si muove nello spazio di simulazione. Le app di servizio vengono configurate nello schema di simulazione. SimSpace Weaver avvia e arresta le app di servizio per te.

Riepilogo dell'app

La tabella seguente offre un riepilogo delle caratteristiche dei diversi tipi di SimSpace Weaver app.

	App spaziali	App personalizzate	App di servizio
Entità di lettura	Sì	Sì	Sì
Aggiorna le entità	Sì	No	No
Creazione di entità	Sì	Sì*	Sì*
Ciclo di vita	Gestito (IoSimSpace Weaver controlla).	Non gestito (sei tu a controllarlo).	Gestito (IoSimSpace Weaver controlla).
Metodo di avvio	SimSpace Weaver avvia un'istanza dell'app per ogni partizione spaziale, come specificato nello schema.	Si avvia ogni istanza dell'app.	SimSpace Weaver avvia una o più istanze di app su ciascun lavoratore, come specificato nello schema.
I clienti possono connettersi	No	Sì	Sì

* Quando un'app o un'app di servizio personalizzata crea un'entità, l'app deve trasferire la proprietà dell'entità a un'app spaziale in modo che l'app spaziale possa aggiornare lo stato dell'entità.

Domini

UnSimSpace Weaver dominio è una raccolta di istanze di app che eseguono lo stesso codice dell'app eseguibile e hanno le stesse opzioni e comandi di avvio. Ci riferiamo ai domini in base ai tipi di app che contengono: domini spaziali, domini personalizzati e domini di servizio. Puoi configurare le tue app all'interno dei domini.

Abbonamenti e replica

Un'app crea una sottoscrizione a un'area spaziale per apprendere gli eventi delle entità (ad esempio, ingresso, uscita, creazione, aggiornamento ed eliminazione) in quella regione. Un'app elabora gli eventi di entità di un abbonamento prima di leggere i dati relativi alle entità in partizioni di cui non è proprietaria.

Una partizione può esistere sullo stesso worker dell'app (chiamata partizione locale), ma un'altra app può possedere la partizione. Una partizione può esistere anche su un altro lavoratore (si chiama partizione remota). Se la sottoscrizione riguarda una partizione remota, il lavoratore crea una copia locale della partizione remota tramite un processo chiamato replica. Il lavoratore legge quindi la copia locale (partizione remota replicata). Se un'altra app sul worker deve leggere da quella partizione con lo stesso segno di spunta, il lavoratore legge la stessa copia locale.

Casi d'uso di esempio per SimSpace Weaver

È possibile utilizzare SimSpace Weaver per modelli basati su agenti e simulazioni di fasi temporali discrete con un componente spaziale.

Crea simulazioni di grandi folle

Puoi utilizzare SimSpace Weaver per simulare la folla in ambienti reali. SimSpace Weaver consente di adattare le simulazioni a milioni di oggetti dinamici con comportamenti propri.

Crea ambienti su scala urbana

Utilizza SimSpace Weaver per creare un gemello digitale di un'intera città. Crea simulazioni per la pianificazione urbana, per progettare percorsi di traffico e pianificare la risposta ai rischi ambientali. Puoi utilizzare le tue fonti di dati geospaziali come elementi costitutivi per i tuoi ambienti.

Crea esperienze coinvolgenti e interattive

Crea esperienze di simulazione a cui più utenti possono partecipare e interagire. Usa strumenti di sviluppo popolari come Unreal Engine e Unity per creare mondi virtuali tridimensionali (3D). Personalizza la tua esperienza 3D con contenuti e comportamenti personalizzati.

Configurazione di SimSpace Weaver

Per iniziare a utilizzarlo SimSpace Weaver per la prima volta, è necessario configurare il proprio ambiente Account AWS e quello locale. Al termine di queste operazioni, sarà possibile passare al [tutorial Nozioni](#) di base su queste operazioni.

Configurazione di configurazione delle

1. [Configura il tuo Account AWS da usare SimSpace Weaver.](#)
2. [Configurazione dell'ambiente locale per SimSpace Weaver.](#)

Configura il tuo Account AWS da usare SimSpace Weaver

Completa le seguenti attività per Account AWS configurarlo da utilizzare SimSpace Weaver.

Registrarsi per creare un Account AWS

Se non disponi di un Account AWS, completa la procedura seguente per crearne uno.

Per registrarsi a un Account AWS

1. Apri la pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Durante la registrazione di un Account AWS, viene creato un Utente root dell'account AWS. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, [assegna l'accesso amministrativo a un utente amministrativo](#) e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

Al termine del processo di registrazione, riceverai un'e-mail di conferma da AWS. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo <https://aws.amazon.com/> e selezionando Il mio account.

Creazione di un utente amministratore

Dopo aver effettuato la registrazione di un Account AWS, proteggi Utente root dell'account AWS, abilita AWS IAM Identity Center e crea un utente amministratore in modo da non utilizzare l'utente root per le attività quotidiane.

Protezione dell'Utente root dell'account AWS

1. Accedi alla [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e immettendo l'indirizzo email del Account AWS. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Accesso come utente root](#) della Guida per l'utente di Accedi ad AWS.

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per ricevere istruzioni, consulta [Abilitazione di un dispositivo MFA virtuale per l'utente root dell'Account AWS \(console\)](#) nella Guida per l'utente IAM.

Creazione di un utente amministratore

1. Abilita IAM Identity Center.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center.

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente amministratore.

Per un tutorial sull'utilizzo di IAM Identity Center directory come origine di identità, consulta [Configurazione dell'accesso utente con IAM Identity Center directory predefinito](#) nella Guida per l'utente di AWS IAM Identity Center.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [Accedere al portale di accesso AWS](#) nella Guida per l'utente Accedi ad AWS.

Aggiungi le autorizzazioni da utilizzare SimSpace Weaver

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center:

Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina [Create a permission set](#) (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center.

- Utenti gestiti in IAM tramite un provider di identità:

Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina [Creating a role for a third-party identity provider \(federation\)](#) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.

- Utenti IAM:

- Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina [Creating a role for an IAM user](#) (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina [Aggiunta di autorizzazioni a un utente \(console\)](#) nella Guida per l'utente di IAM.

Example Politica IAM per concedere le autorizzazioni di utilizzo 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"
      }
    }
  }
]
}

```

Configurazione dell'ambiente locale per SimSpace Weaver

SimSpace Weaver simulazioni eseguite in ambienti containerizzati Amazon Linux 2 (AL2). È necessario disporre di un ambiente AL2 per compilare e collegare le app all'SDK SimSpace Weaver dell'app. L'ambiente di sviluppo locale standard è un contenitore AL2 in Docker. Se scegli di non utilizzarlo Docker, forniamo istruzioni alternative per eseguire un ambiente AL2 in Windows Subsystem for Linux (WSL). Puoi anche usare il tuo metodo per creare un ambiente AL2 locale. Per altri modi di eseguire AL2 localmente, consulta la [documentazione di Amazon EC2](#).

Important

Docker on Microsoft Windows è l'ambiente di sviluppo standard. Per comodità, suggeriamo altri modi per configurare l'ambiente di sviluppo locale, ma non sono standard e non sono supportati.

Argomenti

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

Configurazione Amazon Linux 2 (AL2) in Docker

Questa sezione fornisce istruzioni per configurare l'ambiente AL2 locale in Docker. Per istruzioni su come configurare AL2 in Windows Subsystem for Linux (WSL), vedere [Configurazione Amazon Linux 2 \(AL2\) in Windows Subsystem for Linux \(WSL\)](#)

Requisiti

- Microsoft Windows 10 o superiore
- [Microsoft Visual Studio 2019](#) o più tardi, con il [Desktop development with C++](#) carico di lavoro installato
- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)

Per configurare AL2 in Docker

1. Se non hai ancora configurato AWS le tue credenziali per AWS CLI, segui queste istruzioni: [Configurazione dell'AWS CLI](#). Se lo stai solo utilizzando, SimSpace Weaver puoi configurarlo AWS CLI per utilizzare le tue SimSpace Weaver credenziali come impostazione predefinita.
2. [Scarica il pacchetto distribuibile SimSpace Weaver dell'app SDK](#). Contiene quanto segue:
 - Binari e librerie per lo sviluppo di SimSpace Weaver app

- Script di supporto che automatizzano parti del flusso di lavoro di sviluppo
 - Esempi di applicazioni che illustrano i concetti SimSpace Weaver
3. Decomprimi il file in un *sdk-folder* a tua scelta.
 4. Vai a *sdk-folder*.
 5. Immettete il seguente comando per creare un'Dockerimmagine.

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

Note

Se ricevi un errore durante questo passaggio, assicurati che Docker sia in esecuzione.

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

Questa sezione fornisce istruzioni per configurare l'ambiente AL2 locale in Windows Subsystem for Linux (WSL). Per istruzioni su come configurare AL2 in Docker, vedere [Configurazione Amazon Linux 2 \(AL2\) in Docker](#)

Important

Questa sezione descrive una soluzione che utilizza una versione di AL2 non posseduta, sviluppata o supportata da Amazon. Questa soluzione viene fornita solo per comodità dell'utente, se si sceglie di non Docker utilizzarla. Amazon e non AWS si assumono alcuna responsabilità se scegli di utilizzare questa soluzione.

Requisiti

- [Hyper-V - Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- [Distribuzione AL2 open source di terze parti per WSL \(scarica la versione 2.0.20200722.0-update.2\) \(vedi le istruzioni\)](#)

⚠ Important

[Le nostre istruzioni utilizzano la versione 2.0.20200722.0-update.2 della distribuzione AL2 per. WSL](#) WSL È possibile che si verifichino errori se si utilizza un'altra versione.

Per configurare AL2 in WSL

1. Al prompt dei comandi di Windows, avvia l'ambiente AL2 in. WSL

```
ws1 -d Amazon2
```

⚠ Important

Durante l'esecuzione WSL, utilizzate le istruzioni di Linux anziché le istruzioni di Windows riportate in questa guida (se avete una scelta). Se non ci sono istruzioni Linux, usa le seguenti sostituzioni:

- Usa invece di `tools/linux tools\windows`
- Usa `.sh` gli script anziché gli script `.bat`

2. Al prompt della shell Linux, aggiorna il tuo gestore di pacchetti yum.

```
yum update -y
```

⚠ Important

Se questo passaggio scade, potrebbe essere necessario passare a WSL1 e riprovare queste procedure. Esci dalla sessione WSL AL2 e inserisci quanto segue al prompt dei comandi di Windows:

```
ws1 --set-version Amazon2 1
```

3. Installa lo strumento di decompressione.

```
yum install -y unzip
```

4. Rimuovi tutto AWS CLI ciò che è yum stato installato. Prova entrambi i seguenti comandi se non sei sicuro di aver yum installato unAWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Crea una cartella temporanea e accedi ad essa.

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

6. Scarica e installaAWS CLI:

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

7. È possibile rimuovere la cartella temporanea.

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

8. Riavvia la sessione di shell per aggiornare il percorso nell'ambiente.

```
exec
```

9. Configura AWS le tue credenziali per il AWS CLI nel tuo ambiente AL2. Per ulteriori informazioni, consulta [Configurazione di](#) . Se lo utilizziAWS IAM Identity Center, consulta [Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente](#). AWS Command Line Interface

```
aws configure
```

10. Installa Git.

```
yum install -y git
```

11. Installare wget.

```
yum install -y wget
```

12. Crea una cartella per l'SDK dell'SimSpace Weaverapp.

```
mkdir sdk-folder
```

13. Vai alla tua cartella SDK.

```
cd sdk-folder
```

14. Scarica il file distribuibile SimSpace Weaver dell'SDK dell'app. Contiene quanto segue:

- Binari e librerie per lo sviluppo di SimSpace Weaver app
- Script di supporto che automatizzano parti del flusso di lavoro di sviluppo
- Esempi di applicazioni che illustrano i concetti SimSpace Weaver

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

15. Decomprimere il file.

```
unzip *.zip
```

16. Esegui lo script di configurazione aggiuntivo.

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

Note

È necessario eseguire questa operazione una sola volta per l'ambiente AL2 inWSL.

Utilizzo di software concesso in licenza conAWS SimSpace Weaver

AWS SimSpace Weaver ti consente di creare simulazioni con il motore e i contenuti di simulazione che preferisci. In relazione all'utilizzo di SimSpace Weaver, l'utente è responsabile dell'ottenimento,

della manutenzione e del rispetto dei termini di licenza di qualsiasi software o contenuto utilizzato nelle simulazioni. Verifica che il contratto di licenza ti consenta di distribuire il software e i contenuti in un ambiente ospitato virtuale.

Nozioni di base su SimSpace Weaver

Questa sezione fornisce tutorial che costituiranno un aiuto per iniziare SimSpace Weaver. Questi tutorial ti introducono al flusso di lavoro generale con cui creare simulazioni SimSpace Weaver. In entrambi i tutorial, imparerai come creare, distribuire ed eseguire simulazioni in SimSpace Weaver. Ti consigliamo di iniziare con il tutorial di avvio rapido per avviare una simulazione in pochi minuti. Quindi, consulta il tutorial dettagliato per ulteriori dettagli su ogni fase del processo.

Questi tutorial utilizzano un'applicazione di esempio (`PathfindingSample`) inclusa nel file `.zip` SDK SimSpace Weaver dell'app scaricato durante le [procedure di configurazione](#). L'applicazione di esempio dimostra i concetti condivisi da tutte SimSpace Weaver le simulazioni, tra cui partizionamento spaziale, trasferimento di entità tra partizioni, app e abbonamenti.

Nei tutorial, creerai una simulazione con quattro partizioni spaziali. Un'istanza separata dell'app `PathfindingSample` spaziale gestisce ogni singola partizione. Le app spaziali creano entità nelle proprie partizioni. Le entità si spostano in una posizione particolare nel mondo della simulazione, evitando gli ostacoli mentre si muovono. È possibile utilizzare un'applicazione client separata (inclusa nell'SDK SimSpace Weaver dell'app) per visualizzare la simulazione.

Argomenti

- [Tutorial di avvio rapido: crea ed esegui una simulazione in pochi minuti](#)
- [Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio](#)

Tutorial di avvio rapido: crea ed esegui una simulazione in pochi minuti

Questo tutorial ti guida attraverso il processo di creazione ed esecuzione di una simulazione SimSpace Weaver in pochi minuti. Ti consigliamo di iniziare con questo tutorial e poi di seguire il tutorial dettagliato in seguito.

Requisiti

Prima di iniziare, devi accertarti di aver completato le fasi in [Configurazione di SimSpace Weaver](#).

Fasi

- [Fase 1: creazione di un progetto](#)
- [Passaggio 2: attivare la registrazione \(opzionale\)](#)
- [Passaggio 3: Esegui lo script di avvio rapido](#)
- [Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#)
- [Fase 5: Visualizza la simulazione](#)
- [Passaggio 6: interrompi e ripulisci la simulazione](#)

Fase 1: creazione di un progetto

L'SimSpace Weaverapp SDK distributable contiene uno script che crea un progetto dal progetto in PathfindingSample bundle. È necessario eseguire lo script dalla sua posizione nel file system. Lo script crea un *percorso project-name* interno utilizzando i valori forniti nella riga di comando.

Docker

Per creare un progetto

1. Al Windowsprompt dei comandi, accedete alla cartella del progetto.

```
cd sdk-folder
```

2. Eseguire lo script create-project.bat.

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

Segue l'elenco completo dei comandi e dei parametri.

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

Important

Non utilizzare un nome di progetto più lungo di 20 caratteri. Il superamento di questo limite potrebbe causare errori.

Note

Se ricevi una richiesta di condivisione di un'unità Docker, scegli Yes di condividerla.

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

WSL**Important**

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per creare un progetto

1. Al prompt della Linux shell, accedete alla cartella del progetto.

```
cd sdk-folder
```

2. Eseguire lo script `create-project.sh`.



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

Segue l'elenco completo dei comandi e dei parametri.

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

 Important

Non utilizzare un nome di progetto più lungo di 20 caratteri. Il superamento di questo limite potrebbe causare errori.

 Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Parametri

name

Il nome del progetto.

path

La posizione del progetto nel file system. La struttura del progetto a volte può avere percorsi di file lunghi che possono superare il limite di lunghezza dei percorsi nel sistema operativo in uso. Si consiglia di utilizzare un nome di percorso il più breve possibile.

profile

Il nome del AWS CLI profilo che lo script deve utilizzare per l'autenticazione. Per ulteriori informazioni, consulta [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'AWS Command Line Interface. Questo parametro è disponibile solo per la versione SDK dell'App SDK per SimSpace Weaver 1.12.1 o successiva. Per ulteriori informazioni sulle versioni di SimSpace Weaver, consulta [Versioni SimSpace Weaver](#).

app-sdk-version

(opzionale) La versione dell'App SDK per SimSpace Weaver utilizzata dal progetto. Creerai e collegherai le tue app utilizzando questa versione. Se lo script non trova la versione nella posizione del distribuito o se non fornisci il numero di versione, lo script scaricherà automaticamente la versione più recente.

template

(opzionale) Un modello di progetto che lo script utilizzerà per creare il progetto. Se non fornisci un modello, lo script utilizzerà il `PathfindingSample`. Valori validi:

- **PathfindingSample**— un'applicazione di esempio che utilizza un solo [lavoratore](#).
- **MultiWorkerPathfindingSample**— una versione dell'applicazione di esempio che utilizza più [worker](#).

overwriteproject

(opzionale) Utilizzate questa opzione per sovrascrivere una cartella di progetto esistente con lo stesso nome e percorso di progetto.

Passaggio 2: attivare la registrazione (opzionale)

La registrazione è disattivata per impostazione predefinita per il `PathfindingSample` progetto. Questo tutorial presuppone che la registrazione sia attiva. Puoi scegliere di attivare la registrazione, ma è facoltativa.

⚠ Important

PathfindingSampleGenera grandi quantità di dati di registro. Se scegli di attivare la registrazione, riceverai le spese di fatturazione per i dati di registro. Continuerai a ricevere gli addebiti di fatturazione per i dati di registro finché esisteranno. Ti consigliamo vivamente di interrompere questa simulazione ed eseguire i passaggi di pulizia alla fine del tutorial il prima possibile se attivi la registrazione.

Per attivare la registrazione

1. Apri il seguente file in un editor di testo:

Docker

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

i Note

La *cartella del progetto path\project-name* utilizza i valori che hai fornito al momento della creazione del progetto.

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

i Note

La *cartella del progetto path/project-name* utilizza i valori che hai fornito al momento della creazione del progetto.

2. Trova la `simulation_properties`: sezione all'inizio del file:

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

3. Inserisci le seguenti 2 righe dopo la `simulation_properties::`

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

4. Conferma che la `simulation_properties`: sezione sia uguale alla seguente:

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

5. Salva il file ed esci dall'editor di testo.

Passaggio 3: Esegui lo script di avvio rapido

L'applicazione di esempio include uno script di avvio rapido. Lo script creerà, creerà, caricherà e avvierà la simulazione e le relative app.

Docker

Per eseguire lo script di avvio rapido

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai `path\project-name` utilizzando i valori che hai fornito quando hai creato il progetto. `project-folder`

Al prompt dei Windows comandi, immettete:

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

2. Esegui lo script di avvio rapido per questo progetto.

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

⚠ Important

A partire dalla versione 1.12.3, `quick-start` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

⚠ Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per eseguire lo script di avvio rapido

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Eseguite lo script di avvio rapido per questo progetto.

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

Important

A partire dalla versione 1.12.3, `quick-start` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile` *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Lo script avvierà un ciclo e si interromperà automaticamente dopo che tutti i componenti lo sono stati. STARTED Cerca un output simile al seguente:

```
[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,
```

Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta

Devi ottenere l'indirizzo IP e il numero di porta dell'app View (personalizzata) per poterti connettere alla simulazione. La procedura seguente presuppone che tu non sappia nulla della simulazione (ad esempio il nome della simulazione). È possibile utilizzare questa procedura in qualsiasi momento per trovare l'indirizzo IP e il numero di porta per un'app o un'app di servizio personalizzata. L'output di esempio seguente si riferisce a un progetto denominato MyProject.

Docker

Per ottenere l'indirizzo IP e il numero di porta

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Usa l' ListSimulationsAPI per ottenere il nome della tua simulazione.

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

⚠ Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Output di esempio:

```
{
  "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. Usa l' `DescribeSimulationAPI` per ottenere un elenco di domini nella tua simulazione.

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

Cerca la `Domains` sezione nella `LiveSimulationState` sezione dell'output.

Output di esempio:

```

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

```

4. Usa l' `ListAppsAPI` per ottenere un elenco di app personalizzate in un dominio. Il nome di dominio per l'app di visualizzazione (personalizzata) nel progetto di esempio è `MyViewDomain`. Cerca il nome dell'app nell'output.

```

.\weaver-project-name-cli.bat list-apps --simulation simulation-name --
domain domain-name

```

Output di esempio:

```

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

```

5. Usa l' `DescribeAppAPI` per ottenere l'indirizzo IP e il numero di porta. Per il progetto di esempio, il nome di dominio è `MyViewDomain` e il nome dell'app è `ViewApp`.

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

L'indirizzo IP e il numero di porta si trovano nel `EndpointInfo` blocco dell'output. L'indirizzo IP è il valore di `Address` e il numero di porta è il valore di `Actual`.

Output di esempio:

```
{  
  "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

Il valore di `Declared` è il numero di porta a cui deve essere associato il codice dell'app. Il valore di `Actual` è il numero di porta che SimSpace Weaver i client

possono connettersi alla tua app. SimSpace Weaver mappa la Declared porta alla Actual porta.

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per ottenere l'indirizzo IP e il numero di porta

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Usa l' `ListSimulationsAPI` per ottenere il nome della tua simulazione.

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI](#)

per l'[AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Output di esempio:

```
{
  "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. Usa l' `DescribeSimulationAPI` per ottenere un elenco di domini nella tua simulazione.

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

Cerca la `Domains` sezione nella `LiveSimulationState` sezione dell'output.

Output di esempio:

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

```

        "Lifecycle": "ByRequest"
    }
],

```

4. Usa l' `ListAppsAPI` per ottenere un elenco di app personalizzate in un dominio. Il nome di dominio per l'app di visualizzazione (personalizzata) nel progetto di esempio è `MyViewDomain`. Cerca il nome dell'app nell'output.

```

./weaver-project-name-cli.sh list-apps --simulation simulation-name --
domain domain-name

```

Output di esempio:

```

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

```

5. Usa l' `DescribeAppAPI` per ottenere l'indirizzo IP e il numero di porta. Per il progetto di esempio, il nome di dominio è `MyViewDomain` e il nome dell'app è `ViewApp`.

```

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

```

L'indirizzo IP e il numero di porta si trovano nel `EndpointInfo` blocco dell'output. L'indirizzo IP è il valore di `Address` e il numero di porta è il valore di `Actual`.

Output di esempio:

```
{
  "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

Il valore di `Declared` è il numero di porta a cui deve essere associato il codice dell'app. Il valore di `Actual` è il numero di porta che SimSpace Weaver i client possono connettersi alla tua app. SimSpace Weaver mappa la `Declared` porta alla `Actual` porta.

Fase 5: Visualizza la simulazione

L'SDK SimSpace Weaver dell'app offre diverse opzioni per visualizzare l'applicazione di esempio. Puoi utilizzare il client della console di esempio se non disponi di alcun supporto locale per Unreal Engine lo sviluppo. Le istruzioni per il Unreal Engine client presuppongono che tu stia utilizzando Windows.

Il client della console visualizza un elenco di eventi di entità man mano che si verificano. Il client ottiene le informazioni sugli eventi dell'entità da `ViewApp`. Se il client della console visualizza l'elenco degli eventi, conferma la connettività di rete con l'attività `ViewApp` e della simulazione.

La PathfindingSample simulazione crea entità fisse e mobili su un piano bidimensionale. Le entità mobili si muovono attorno alle entità fisse. Il client Unreal Engine fornisce una visualizzazione degli eventi dell'entità.

Windows console client

Requisiti

- Microsoft Windows 10o superiore
- [Microsoft Visual Studio 2019](#)o più tardi, con il [Desktop development with C++](#)carico di lavoro installato
- [CMake3](#)
- [Git](#)

Per connettersi all'applicazione di esempio con il client della console di esempio

1. In una finestra del prompt dei comandi, vai alla cartella del client della console (nella cartella SDK dell'app).

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

2. Usa CMake3 per creare una Visual Studio soluzione in questa cartella.

```
cmake .
```

Note

Assicurati di includere lo spazio e il periodo alla fine.

Important

Tieni aperta la finestra del prompt dei comandi per ulteriori passaggi.

3. InVisual Studio, apri PathfindingSampleConsoleClient.sln quello che hai creato nel passaggio precedente.
4. Seleziona la configurazione di RelWithDebInfo.

5. Scegliete Build> Build Solution.
6. Nella finestra precedente del prompt dei comandi, vai alla cartella di output della build nella cartella del client della console.

```
cd RelWithDebInfo
```

7. Esegui il client con l'indirizzo IP e il numero di porta del tuo ViewApp.

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

La finestra del prompt dei comandi dovrebbe mostrare i numeri per l'aggiornamento, l'eliminazione e la creazione di eventi dell'entità, simili all'output di esempio seguente.

Note

Gli indirizzi IP e i numeri di porta nell'output di esempio seguente sono segnaposto. Fornisci l'indirizzo IP e il numero di porta del tuo ViewApp al client della console. Fornisci il numero di Actual porta se desideri connetterti a un ViewApp sistema eseguito inCloud AWS. Fornisci l'indirizzo IP e il numero di porta 127.0.0. 1:7000 quando ti connetti a un sistema ViewApp che funziona sul tuo sistema locale. Per ulteriori informazioni, consulta [Sviluppo locale](#).

```
##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

Per una guida alla risoluzione dei problemi, vedere. [PathfindingSample il client della console non riesce a connettersi](#)

8. Premere CTRL+C per chiudere il client della console.

Linux console client

Important

Forniamo queste istruzioni per facilitare l'utente. Potrebbero non funzionare in alcuni Linux ambienti. Queste procedure non sono supportate.

Questa procedura presuppone che l'utente stia lavorando interamente all'interno di un Linux ambiente. È inoltre possibile visualizzare la simulazione utilizzando client integrati. Windows

Requisiti

- CMake3
- Ccompilatore (già incluso) Amazon Linux 2
- Git

Per connettersi all'applicazione di esempio con il client della console di esempio

1. Al prompt della Linux shell, accedi alla cartella del client della console (nella cartella SDK dell'app).

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

2. Crea una cartella di compilazione.

```
mkdir build
```

3. Vai alla cartella di compilazione.

```
cd build
```

4. Usa CMake3 per creare il client.

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

Note

Assicurati di includere lo spazio e il periodo alla fine.

5. Esegui il client con l'indirizzo IP e il numero di porta del tuo ViewApp.

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

La finestra del prompt dei comandi dovrebbe mostrare i numeri per l'aggiornamento, l'eliminazione e la creazione di eventi dell'entità, simili all'output di esempio seguente.

Note

Gli indirizzi IP e i numeri di porta nell'output di esempio seguente sono segnaposto. Fornisci l'indirizzo IP e il numero di porta del tuo ViewApp al client della console. Fornisci il numero di Actual porta se desideri connetterti a un ViewApp sistema eseguito inCloud AWS. Fornisci l'indirizzo IP e il numero di porta 127.0.0. 1:7000 quando ti connetti a un sistema ViewApp che funziona sul tuo sistema locale. Per ulteriori informazioni, consulta [Sviluppo locale](#).

```
##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

Per una guida alla risoluzione dei problemi, vedere. [PathfindingSample il client della console non riesce a connettersi](#)

6. Premere CTRL+C per chiudere il client della console.

Unreal Engine on Windows

Requisiti

- Unreal Engine 5 ambiente di sviluppo
- Microsoft .NET Framework 4.8 Developer Pack
- Windowsclient di console (vedi la scheda client della Windows console in questa pagina)

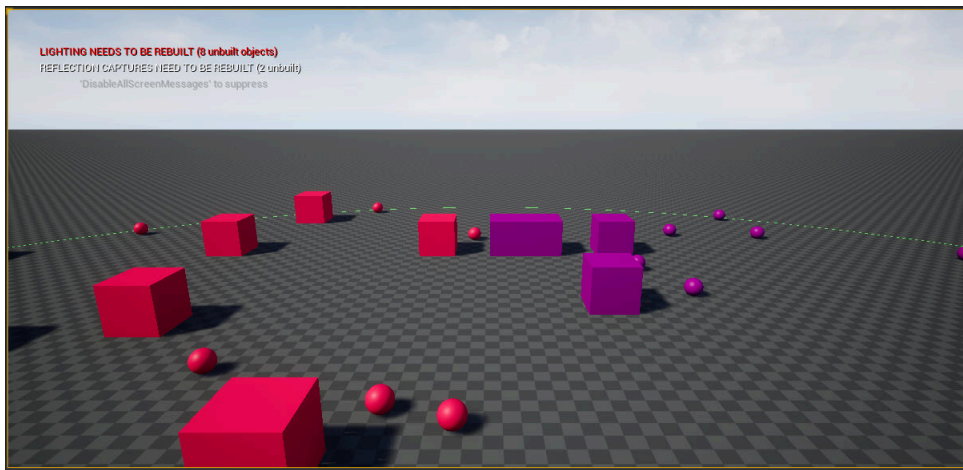
Important

Altre versioni di Unreal Engine e non .NET sono supportate e potrebbero causare problemi.

Per connettersi all'applicazione di esempio con il Unreal client di esempio

1. Il Unreal Engine client utilizza la NNG libreria del client della console. È necessario creare il client della console Windows se non l'hai già creato. Per ulteriori informazioni, consulta la scheda client della Windows console in questa pagina.
2. In una finestra di gestione dei file, vai a `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Aprire `UnrealClient.uproject`.
4. Se l'editor ti chiede se vuoi ricostruire i `UnrealClient` moduli, scegli yes.
5. In un editor di testo, apri `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`.
6. Aggiorna l'URL con l'indirizzo IP e il numero di porta dell'app View: `tcp://ip-address:port-number` (dovrebbe apparire così `tcp://198.51.100.135:1234`).
7. Nell'Unrealeditor, scegli play.

L'Unrealeditor dovrebbe mostrare una visualizzazione della simulazione, simile alla schermata seguente.



Note

A seconda della potenza del sistema di sviluppo locale, l'Unrealeditor potrebbe impiegare alcuni minuti per visualizzare la simulazione. Durante questo periodo, il sistema potrebbe sembrare bloccato.

Usa i D tasti W A S,, per muoverti nel Unreal client. Tieni premuto il pulsante del mouse e trascina il mouse per ruotare.

Puoi premere il tasto [(parentesi quadra sinistra) per ridurre le dimensioni dell'area di iscrizione. Puoi premere il tasto] (parentesi quadra destra) per aumentare le dimensioni dell'area di sottoscrizione. La dimensione dell'area di sottoscrizione determina il numero di entità che appaiono nel client.

È possibile premere il C tasto per creare un'entità nella simulazione. Il client invia un CreateEntity comando all'app di visualizzazione. L'app di visualizzazione creerà quindi l'entità e la trasferirà nel dominio spaziale.


Puoi esaminare il codice per ViewAppDriver::HandleEntityCreationRequests `project-folder\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp` vedere come l'app implementa questo processo.

Passaggio 6: interrompi e ripulisci la simulazione


È importante ripulire le simulazioni quando non ne hai più bisogno. SimSpace Weaver le risorse di simulazione vengono conteggiate ai fini delle quote (limiti) di servizio, anche se la simulazione

viene interrotta. Continuerai a ricevere addebiti di fatturazione per le simulazioni in esecuzione. Potresti anche addebitare costi di fatturazione per l'archiviazione dei dati nei servizi di supporto, come Amazon CloudWatch Logs e Amazon Simple Storage Service. Per ulteriori informazioni sulle quote di SimSpace Weaver servizio, consulta. [SimSpace Endpoint e quote Weaver](#)

Segui le procedure in questa sezione quando sei pronto per ripulire la simulazione.

 Important

Non è possibile riavviare una simulazione interrotta.

 Important

Non è possibile recuperare una simulazione eliminata.

Pulisci le risorse di simulazione in SimSpace Weaver

È necessario interrompere la simulazione prima di poterla eliminare. L'eliminazione di una simulazione rimuove solo le risorse. SimSpace Weaver È necessario eseguire passaggi separati per eliminare le risorse create dalla simulazione o utilizzate in altri servizi (vedere la sezione seguente).

Docker

Per ripulire la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Trova i nomi delle tue simulazioni.

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

⚠ Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

3. Interrompi una simulazione.

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

4. Eliminare una simulazione interrotta.

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

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per ripulire la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Trova i nomi delle tue simulazioni.

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

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

3. Interrompi una simulazione.

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

4. Eliminare una simulazione interrotta.

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

AWS Management Console


Per ripulire la simulazione

1. Apri la SimSpace Weaver console dalla [SimSpace Weaverconsole](#).
2. Dal pannello di navigazione, scegli Simulations.

3. Dall'Simulationselecco, seleziona l'opzione accanto al nome della simulazione che desideri eliminare.
4. Se il Statusnome della simulazione selezionata è: STARTED
 - a. Scegli il Actionsmenu a discesa.
 - b. Scegli Stop.
 - c. Per confermare, inserisci il nome della simulazione.
 - d. Scegli Stop.
 - e. StatusAttendi che sia finita la simulazione. STOPPED
5. Scegli il Actionsmenu a discesa.
6. Scegli Delete.
7. Per confermare, scegli. Delete

Pulisci le risorse di simulazione nei servizi di supporto

Per supportare la simulazione, SimSpace Weaver crea risorse in altri servizi. SimSpace Weaver non elimina queste risorse quando elimini la simulazione. Puoi eliminare queste risorse aggiuntive se non ne hai bisogno.

 Important

Potresti ricevere addebiti di fatturazione per ognuna di queste risorse che non elimini.

Per eliminare le risorse di supporto per il tuo progetto

1. Se hai finito con il progetto, elimina il relativo AWS CloudFormation stack. Per ulteriori informazioni sull'utilizzo diAWS CloudFormation, consulta [Eliminazione di uno stack sulla AWS CloudFormation console nella Guida per l'utente](#). AWS CloudFormation

- *weaver- nome-progetto -stack*

⚠ Important

Le simulazioni avviate dallo stesso progetto condividono risorse come il ruolo dell'app. Quando elimini lo AWS CloudFormation stack, elimini il ruolo dell'app. Non eliminare lo AWS CloudFormation stack se hai altre simulazioni che condividono le stesse risorse.

ℹ Note

Il tuo AWS CloudFormation stack verrà probabilmente segnalato DELETE_FAILED perché non può eliminare i bucket Amazon S3 che non sono vuoti. Eliminerai i bucket Amazon S3 nel passaggio successivo.

2. Se hai finito con il tuo progetto, elimina i relativi bucket Amazon S3. Per ulteriori informazioni sull'utilizzo dei bucket Amazon S3, consulta [Eliminazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

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

Ad esempio, il progetto indicato MyProject nell'account 111122223333 nella us-west-2 regione ha il seguente bucket:

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

ℹ Note

È necessario eliminare il contenuto di un bucket Amazon S3 prima di poter eliminare il bucket.

ℹ Note

SimSpace WeaverI progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- *weaver- - numero di conto -app-zips- regione lowercase-project-name*
- *weaver- - numero-conto -schemi- regione lowercase-project-name*

3. Se hai attivato la registrazione per la simulazione, elimina il gruppo Logs log. CloudWatch Per ulteriori informazioni sull'utilizzo dei CloudWatch log, consulta [Working with log groups and log stream](#) nella Amazon CloudWatch Logs User Guide.

Il nome del gruppo di log per la simulazione è specificato nel relativo schema (file di configurazione): *project-folder\tools\project-name.yaml*

Il nome del gruppo di log è il valore di `log_destination_resource_name`. Il seguente frammento di schema mostra che il gruppo di log per l'applicazione di esempio è.

MySimulationLogs

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

Warning

Se si avviano più simulazioni che specificano lo stesso gruppo di log, i dati di log per tutte le simulazioni verranno inseriti nello stesso gruppo di log. Se si elimina il gruppo di log, si eliminano i dati di registro per tutte le simulazioni che utilizzano quel gruppo di log. Se si elimina un gruppo di log per una simulazione in esecuzione, la simulazione avrà esito negativo.

Important

Se lo schema della simulazione è specifico `log_destination_service: "logs"` e un `log_destination_resource_name` ma non riesci a trovare il gruppo di log in

CloudWatch Logs, assicurati di controllare lo stesso Regione AWS in cui è stata eseguita la simulazione.

Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio

Questo tutorial ti guida attraverso la stessa procedura generale presentata nel tutorial di avvio rapido, ma in modo più dettagliato. Il tutorial di avvio rapido ha semplificato molti passaggi e ha nascosto i dettagli all'interno dell'automazione. Questo tutorial espone e spiega questi dettagli.

Ti consigliamo di completare il [tutorial di avvio rapido](#) prima di procedere con questo tutorial.

Requisiti

Prima di iniziare, devi accertarti di aver completato le fasi in [Configurazione di SimSpace Weaver](#).

Fasi

- [Fase 1: creazione di un progetto](#)
- [Passaggio 2: attivare la registrazione \(opzionale\)](#)
- [Fase 3: Carica lo schema di simulazione](#)
- [Fase 4: Costruire il progetto](#)
- [Passaggio 5: caricare app](#)
- [Fase 6: Inizia la simulazione](#)
- [Fase 7: Ottieni i dettagli della simulazione](#)
- [Passaggio 8: Avvia app personalizzate](#)
- [Passaggio 9: avviare l'orologio](#)
- [Fase 10: Controllare i registri](#)
- [Fase 11: Visualizza la simulazione](#)
- [Passaggio 12: interrompi e ripulisci la simulazione](#)

Fase 1: creazione di un progetto

L'SimSpace Weaver SDK distributable contiene uno script che crea un progetto dal progetto in PathfindingSample bundle. È necessario eseguire lo script dalla sua posizione nel file system. Lo script crea un *percorso project-name* interno utilizzando i valori forniti nella riga di comando.

Docker

Per creare un progetto

1. Al Windows prompt dei comandi, accedete alla cartella del progetto.

```
cd sdk-folder
```

2. Eseguire lo script `create-project.bat`.

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

Segue l'elenco completo dei comandi e dei parametri.

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

Important

Non utilizzare un nome di progetto più lungo di 20 caratteri. Il superamento di questo limite potrebbe causare errori.

Note

Se ricevi una richiesta di condivisione di un'unità Docker, scegli Yes di condividerla.

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK

SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per creare un progetto

1. Al prompt della Linux shell, accedete alla cartella del progetto.

```
cd sdk-folder
```

2. Eseguire lo script `create-project.sh`.

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

Segue l'elenco completo dei comandi e dei parametri.

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

Important

Non utilizzare un nome di progetto più lungo di 20 caratteri. Il superamento di questo limite potrebbe causare errori.

⚠ Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#)utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Parametri**name**

Il nome del progetto.

path

La posizione del progetto nel file system. La struttura del progetto a volte può avere percorsi di file lunghi che possono superare il limite di lunghezza dei percorsi nel sistema operativo in uso. Si consiglia di utilizzare un nome di percorso il più breve possibile.

profile

Il nome del AWS CLI profilo che lo script deve utilizzare per l'autenticazione. Per ulteriori informazioni, consulta [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'AWS Command Line Interfaceutente. Questo parametro è disponibile solo per la versione SDK dell'SimSpace Weaverapp 1.12.1 o successiva. Per ulteriori informazioni sulle versioni di SimSpace Weaver, consulta [Versioni SimSpace Weaver](#).

app-sdk-version

(opzionale) La versione dell'SDK dell'SimSpace Weaverapp utilizzata dal progetto. Creerai e collegherai le tue app utilizzando questa versione. Se lo script non trova la versione

nella posizione del distribuibile o se non fornisci il numero di versione, lo script scaricherà automaticamente la versione più recente.

template

(opzionale) Un modello di progetto che lo script utilizzerà per creare il progetto. Se non fornisci un modello, lo script utilizzerà il `PathfindingSample`. Valori validi:

- **PathfindingSample**— un'applicazione di esempio che utilizza un solo [lavoratore](#).
- **MultiWorkerPathfindingSample**— una versione dell'applicazione di esempio che utilizza più [worker](#).

overwriteproject

(opzionale) Utilizzate questa opzione per sovrascrivere una cartella di progetto esistente con lo stesso nome e percorso di progetto.

Passaggio 2: attivare la registrazione (opzionale)

La registrazione è disattivata per impostazione predefinita per il `PathfindingSample` progetto. Questo tutorial presuppone che la registrazione sia attiva. Puoi scegliere di attivare la registrazione, ma è facoltativa.

Important

`PathfindingSample` Genera grandi quantità di dati di registro. Se scegli di attivare la registrazione, riceverai le spese di fatturazione per i dati di registro. Continuerai a ricevere gli addebiti di fatturazione per i dati di registro finché esisteranno. Ti consigliamo vivamente di interrompere questa simulazione ed eseguire i passaggi di pulizia alla fine del tutorial il prima possibile se attivi la registrazione.

Per attivare la registrazione

1. Apri il seguente file in un editor di testo:

Docker

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


Note

La *cartella del progetto* `path\project-name` utilizza i valori che hai fornito al momento della creazione del progetto.

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

Note

La *cartella del progetto* `path/project-name` utilizza i valori che hai fornito al momento della creazione del progetto.

2. Trova la `simulation_properties`: sezione all'inizio del file:

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

3. Inserisci le seguenti 2 righe dopo la riga `simulation_properties::`

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

4. Conferma che la `simulation_properties`: sezione sia uguale alla seguente:

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

5. Salva il file ed esci dall'editor di testo.

Fase 3: Carica lo schema di simulazione

SimSpace Weaver utilizza uno schema per configurare la simulazione. Lo schema è un file YAML di testo semplice in formato. Per ulteriori informazioni, consulta [Configurazione della simulazione](#).

Docker

L'applicazione di esempio viene fornita con uno schema preconfigurato. Puoi trovare il file di schema dell'applicazione di esempio nella cartella degli strumenti del tuo progetto:

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

Per caricare il tuo schema

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path**project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Utilizzate lo script di supporto per caricare lo schema.

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profilo per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta

[Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

L'applicazione di esempio viene fornita con uno schema preconfigurato. Puoi trovare il file di schema dell'applicazione di esempio nella cartella degli strumenti del tuo progetto:

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

Per caricare il tuo schema

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Utilizzate lo script di supporto per caricare lo schema.

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni

SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Fase 4: Costruire il progetto

Ora sei pronto per creare le app spaziali e personalizzate per il progetto di esempio. Il progetto di esempio include uno script di supporto che crea queste app per te.

Docker

Lo script avvierà un contenitore Docker utilizzando l'immagine Docker creata durante [la configurazione dell'ambiente locale](#). Lo script esegue la tua build in un ambiente Amazon Linux nel contenitore Docker. Scrive gli artefatti della build e le relative dipendenze *project-folder*\build nella cartella di Windows.

Per costruire il tuo progetto

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Sta *path**project-name* usando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al Windows prompt dei comandi, inserisci:

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

2. Usa lo script di supporto per creare il progetto.

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

WSL

Important

Forniamo le istruzioni per comodità dell'utente. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Lo script scrive gli artefatti della build e le relative dipendenze *project-folder/build* nella cartella.

Per costruire il tuo progetto

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Your *project-folder* *stapath/project-name* usando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Usa lo script di supporto per creare il progetto.

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

Passaggio 5: caricare app

Lo script di compilazione ha impacchettato le tue app come file zip. Devi caricare questi file zip in bucket specifici in Amazon Simple Storage Service per eseguire la SimSpace Weaver simulazione nel cloud. L'SDK SimSpace Weaver dell'app fornisce uno script di supporto per gestire il caricamento.

Docker

Per caricare le tue app

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Usa lo script di supporto per caricare le tue app.

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

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per caricare le tue app

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Usa lo script di supporto per caricare le tue app.

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Controlla le tue risorse Amazon S3

Puoi controllare i tuoi bucket Amazon S3 per assicurarti che tutti i caricamenti siano stati eseguiti correttamente. Per ulteriori informazioni sull'uso di Amazon S3, consulta [Creazione, configurazione e utilizzo dei bucket Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service](#).

Per l'applicazione di esempio, lo schema (che hai caricato in un passaggio precedente) e le risorse dell'app utilizzano i seguenti formati di nome:

- Secchio di schemi: *simspaceweaver-project-name-lowercase-account-number-schemas-region*
- File di schema: *project-name*-schema.yaml
- Bucket per app: *simspaceweaver-project-name-lowercase-account-number-app-zips-region*

- App spaziale: *project-name*Spatial.zip
- Visualizza l'app (personalizzata): *project-name*View.zip

Ad esempio, date le seguenti proprietà del progetto:

- Nome progetto: MyProject
- AWSnumero di conto: 111122223333
- Regione AWS: us-west-2

Lo schema e le risorse dell'app avrebbero i seguenti nomi:

- Bucket dello schema: simspaceweaver-myproject-111122223333-schemas-us-west-2
 - File di schema: MyProject-schema.yaml
- Bucket per app: simspaceweaver-myproject-111122223333-apps-zips-us-west-2
 - App spaziale: MyProjectSpatial.zip
 - Visualizza l'app (personalizzata): MyProjectView.zip

Fase 6: Inizia la simulazione

L'SDK SimSpace Weaver dell'app fornisce uno script di supporto per avviare la simulazione. Lo script è un wrapper per una chiamata API. StartSimulation Fornisce i seguenti parametri alla chiamata API:

- Il nome dello schema di simulazione (che hai caricato in un passaggio precedente)
- Il nome della simulazione
- L'endpoint del SimSpace Weaver servizio

Docker

Per iniziare la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:


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

2. Usa lo script di supporto per caricare e avviare la simulazione.

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

Important

A partire dalla versione 1.12.3, `start-simulation` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Note

`run-project-name.bat` è uno script di supporto alternativo che avvia anche l'orologio di simulazione. In questo tutorial, avvierai l'orologio separatamente in un passaggio successivo.

A partire dalla versione 1.12.3, `run` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per iniziare la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Usa lo script di supporto per caricare e avviare la simulazione.

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

Important

A partire dalla versione 1.12.3, `start-simulation` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni

hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Note

`run-project-name.sh` è uno script di supporto alternativo che avvia anche l'orologio di simulazione. In questo tutorial, avvierai l'orologio separatamente in un passaggio successivo.

A partire dalla versione 1.12.3, `run` lo script avvia la simulazione con una durata massima di 1 ora. È possibile utilizzare il `--maximum-duration` parametro per specificare una durata massima diversa. Nella versione 1.12.2 o precedente, non è possibile fornire una durata massima allo script e le simulazioni hanno una durata massima di 14 giorni. Per ulteriori informazioni sulla durata massima di una simulazione, vedere. [Durata massima di un flusso di un flusso](#)

Lo script verrà ripetuto fino a quando lo stato della simulazione non sarà pari a oSTARTED. FAILED L'avvio di una simulazione può richiedere alcuni minuti. Se la simulazione inizia con successo, verrà visualizzato un risultato simile al seguente:

```
[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 i progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- *weaver- - numero di conto -app-zips- regione lowercase-project-name*
- *weaver- - numero-conto -schemi- regione lowercase-project-name*

Fase 7: Ottieni i dettagli della simulazione

L'SDK SimSpace Weaver dell'app fornisce uno script di supporto che avvolge il. AWS CLI Lo script semplifica le chiamate a fornendo l'AWS CLI endpoint del servizio. SimSpace Weaver Si utilizza questo script di supporto per chiamare le API. SimSpace Weaver L'DescribeSimulationAPI fornisce dettagli sulla simulazione, incluso lo stato. Una simulazione può trovarsi in uno dei seguenti stati:

Stati del ciclo di vita della simulazione

1. **STARTING**— Stato iniziale dopo la chiamata StartSimulation
2. **STARTED**— tutte le app spaziali sono avviate e funzionanti
3. **STOPPING**— Stato iniziale dopo la chiamata StopSimulation
4. **STOPPED**— Tutte le risorse di elaborazione vengono interrotte
5. **DELETING**— Stato iniziale dopo la chiamata DeleteSimulation
6. **DELETED**— Tutte le risorse assegnate alla simulazione vengono eliminate
7. **FAILED**— La simulazione ha avuto un errore/fallimento critico e si è interrotta

8. SNAPSHOT_IN_PROGRESS— È in corso un'[istantanea](#)

Docker

Per ottenere i dettagli della simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai `path\project-name` utilizzando i valori che hai fornito quando hai creato il progetto. `project-folder`

Al prompt dei Windows comandi, immettete:

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

2. Utilizza lo script di supporto CLI per chiamare l'API. ListSimulations

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'app SDK. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Lo script dovrebbe mostrare dettagli su ciascuna delle simulazioni, simili ai seguenti:

```
{  
  "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. Chiama DescribeSimulation per avere i dettagli della simulazione. Sostituiscilo *simulation-name* con la simulazione dell'output del passaggio precedente.

```

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

```

Lo script dovrebbe mostrare maggiori dettagli sulla simulazione specificata, simili ai seguenti:

```

{
  "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

Forniamo queste istruzioni per agevolare l'utente. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale perSimSpace Weaver](#).

Per ottenere i dettagli della simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Utilizza lo script di supporto CLI per chiamare l'API. ListSimulations

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'app SDK. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Lo script dovrebbe mostrare dettagli su ciascuna delle simulazioni, simili ai seguenti:

```
{
  "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. Chiama DescribeSimulation per avere i dettagli della simulazione. Sostituiscilo *simulation-name* Name con la simulazione dell'output del passaggio precedente.

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


Lo script dovrebbe mostrare maggiori dettagli sulla simulazione specificata, simili ai seguenti:

```
{
  "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": "[]"
}
```

Passaggio 8: Avvia app personalizzate

SimSpace Weaver non gestisce il ciclo di vita delle app personalizzate. È necessario avviare le app personalizzate. È consigliabile avviare le app personalizzate prima di avviare l'orologio di simulazione, ma è possibile avviare le app personalizzate dopo aver avviato l'orologio.

Puoi utilizzare lo script di supporto CLI per chiamare l'StartAppAPI per avviare le tue app personalizzate.

Important

Se utilizzate AWS IAM Identity Center o denominate profili per AWS Command Line Interface (AWS CLI), dovete utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'SDK dell'app. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Docker

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

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

La chiamata StartApp API creerà e avvierà una nuova istanza dell'app personalizzata utilizzando il nome fornito dall'utente. Se fornisci il nome di un'app già esistente, riceverai un errore. Se desideri riavviare una particolare app (istanza), devi prima arrestare quell'app ed eliminarla.

Note

Lo stato della simulazione deve essere quello STARTED precedente all'avvio di app personalizzate. Per verificare lo stato della simulazione, consulta. [Fase 7: Ottieni i dettagli della simulazione](#)

L'applicazione di esempio fornisce l'app ViewApp personalizzata per visualizzare la simulazione. Questa app fornisce un indirizzo IP statico e un numero di porta per connettere i client di simulazione (lo farai in un passaggio successivo di questo tutorial). Puoi pensare a una domain classe di app che hanno lo stesso codice eseguibile e le stesse opzioni di avvio. app namelidentifica l'istanza dell'app. Per ulteriori informazioni sui SimSpace Weaver concetti, vedere [Concetti chiave per SimSpace Weaver](#).

Puoi utilizzare l'DescribeAppAPI per verificare lo stato di un'app personalizzata dopo averla avviata.

Docker

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

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

Docker

Per avviare l'app di visualizzazione in questo tutorial

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Usa lo script di supporto CLI per chiamare. StartApp ViewApp

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

3. Chiama DescribeApp per verificare lo stato della tua app personalizzata.

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

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per avviare l'app di visualizzazione in questo tutorial

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* usando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Usa lo script di supporto CLI per chiamare. StartApp ViewApp

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

3. Chiama DescribeApp per verificare lo stato della tua app personalizzata.

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

Dopo lo stato dell'app personalizzata (istanza)STARTED, l'output di DescribeApp includerà l'indirizzo IP e il numero di porta dell'app personalizzata (istanza). Nell'output di esempio seguente, l'indirizzo IP è il valore di Address e il numero di porta è il valore di Actual nel EndpointInfo blocco.

```
{  
  "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

Il valore di `Declared` è il numero di porta a cui deve essere associato il codice dell'app. Il valore di `Actual` è il numero di porta che SimSpace Weaver i client possono connettersi alla tua app. SimSpace Weaver mappa la `Declared` porta alla `Actual` porta.

Note

Puoi utilizzare la [procedura del tutorial di avvio rapido](#) per ottenere l'indirizzo IP e il numero di porta di qualsiasi app personalizzata avviata, indipendentemente da questo flusso di lavoro.

Passaggio 9: avviare l'orologio

Quando crei la simulazione per la prima volta, c'è un orologio ma l'orologio non è in funzione. Quando l'orologio non è in funzione, la simulazione non aggiornerà il suo stato. Dopo aver avviato l'orologio, inizierà a inviare segni di spunta alle tue app. Ogni segno di spunta, le tue app spaziali esaminano le entità che possiedono e a cui inviano i risultati SimSpace Weaver

Note

L'avvio dell'orologio può richiedere 30-60 secondi.

⚠ Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Docker

Per avviare l'orologio

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai *path\project-name* utilizzando i valori che hai fornito quando hai creato il progetto. *project-folder*

Al prompt dei Windows comandi, immettete:

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

2. Utilizza lo script di supporto CLI per chiamare l'API. StartClock

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

i Note

L'StartClockAPI utilizza il tuo *simulation-name*, che puoi trovare utilizzando l'ListSimulationsAPI:

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

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per avviare l'orologio

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Utilizza lo script di supporto CLI per chiamare l'API. StartClock

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

Note

L'StartClockAPI utilizza il tuo *simulation-name*, che puoi trovare utilizzando l'ListSimulationsAPI:

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

Fase 10: Controllare i registri

SimSpace Weaverscrive i messaggi di gestione della simulazione e l'output della console dalle tue app su Amazon CloudWatch Logs. Per ulteriori informazioni sull'utilizzo dei log, consulta [Working with log groups and log stream](#) nella Amazon CloudWatch Logs User Guide.

Ogni simulazione che crei ha il proprio gruppo di log in Logs. CloudWatch Il nome del gruppo di log è specificato nello schema di simulazione. Nel frammento di schema

seguito, il valore di `log_destination_service` è `logs`. Ciò significa che il valore di `log_destination_resource_name` è il nome di un gruppo di log. In questo caso, il gruppo di log è `MySimulationLogs`.

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

Puoi anche utilizzare `DescribeSimulationAPI` per trovare il nome del gruppo di log per la simulazione dopo averlo avviato.

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva di SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni di SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK di SimSpace Weaver dell'app utilizzano il CLI di AWS. Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo profilo `default` o fornire il nome del tuo profilo IAM Identity Center per applicare gli script di SimSpace Weaver SDK con il parametro `AWS CLI --profile cli-profile-name`. Per ulteriori informazioni, consulta [Configurazione dell'uso di AWS IAM Identity Center nella Guida AWS CLI per l'utente](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente di AWS Command Line Interface.

Docker

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

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

L'esempio seguente mostra la parte dell'output DescribeSimulation che descrive la configurazione di registrazione. Il nome del gruppo di log viene visualizzato alla fine di LogGroupArn.

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

Ogni gruppo di log di simulazione contiene diversi flussi di log:

- Flusso di log di gestione: messaggi di gestione della simulazione prodotti dal servizio. SimSpace Weaver

```
/sim/management
```

- Flusso di registro degli errori: messaggi di errore prodotti dal SimSpace Weaver servizio. Questo flusso di log esiste solo in presenza di errori. SimSpace Weaver archivia gli errori scritti dalle tue app nei rispettivi flussi di log delle app (vedi i seguenti flussi di log).

```
/sim/errors
```

- Stream di log spaziali delle app (1 per ogni app spaziale su ciascun worker): output della console prodotto dalle app spaziali. Ogni app spaziale scrive nel proprio flusso di log. Sono *spatial-app-id* tutti i caratteri dopo la barra finale alla fine di *worker-id*

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

- Stream di log delle app personalizzati (1 per ogni istanza di app personalizzata): output della console prodotto da app personalizzate. Ogni istanza dell'app personalizzata scrive nel proprio flusso di log.

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

- Stream di log delle app di servizio (1 per ogni istanza dell'app di servizio): output della console prodotto dalle app di servizio. Ogni app di servizio scrive nel proprio flusso di log. Sono *service-app-id* tutti i caratteri dopo la barra finale alla fine di *service-app-name*

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

Note

L'applicazione di esempio non dispone di app di servizio.

Fase 11: Visualizza la simulazione

L'SDK SimSpace Weaver dell'app offre diverse opzioni per visualizzare l'applicazione di esempio. Puoi utilizzare il client della console di esempio se non disponi di alcun supporto locale per Unreal Engine lo sviluppo. Le istruzioni per il Unreal Engine client presuppongono che tu stia utilizzando Windows.

Il client della console visualizza un elenco di eventi di entità man mano che si verificano. Il client ottiene le informazioni sugli eventi dell'entità da `ViewApp`. Se il client della console visualizza l'elenco degli eventi, conferma la connettività di rete con l'attività `ViewApp` e della simulazione.

La PathfindingSample simulazione crea entità fisse e mobili su un piano bidimensionale. Le entità mobili si muovono attorno alle entità fisse. Il client Unreal Engine fornisce una visualizzazione degli eventi dell'entità.

Windows console client

Requisiti

- Microsoft Windows 10o superiore
- [Microsoft Visual Studio 2019](#)o più tardi, con il [Desktop development with C++](#)carico di lavoro installato
- [CMake3](#)
- [Git](#)

Per connettersi all'applicazione di esempio con il client della console di esempio

1. In una finestra del prompt dei comandi, vai alla cartella del client della console (nella cartella SDK dell'app).

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

2. Usa CMake3 per creare una Visual Studio soluzione in questa cartella.

```
cmake .
```

Note

Assicurati di includere lo spazio e il periodo alla fine.

Important

Tieni aperta la finestra del prompt dei comandi per ulteriori passaggi.

3. InVisual Studio, apri PathfindingSampleConsoleClient.sln quello che hai creato nel passaggio precedente.
4. Seleziona la configurazione di RelWithDebInfo.

5. Scegliete Build> Build Solution.
6. Nella finestra precedente del prompt dei comandi, vai alla cartella di output della build nella cartella del client della console.

```
cd RelWithDebInfo
```

7. Esegui il client con l'indirizzo IP e il numero di porta del tuoViewApp.

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

La finestra del prompt dei comandi dovrebbe mostrare i numeri per l'aggiornamento, l'eliminazione e la creazione di eventi dell'entità, simili all'output di esempio seguente.

Note

Gli indirizzi IP e i numeri di porta nell'output di esempio seguente sono segnaposto. Fornisci l'indirizzo IP e il numero di porta del tuo ViewApp al client della console. Fornisci il numero di Actual porta se desideri connetterti a un ViewApp sistema eseguito inCloud AWS. Fornisci l'indirizzo IP e il numero di porta 127.0.0. 1:7000 quando ti connetti a un sistema ViewApp che funziona sul tuo sistema locale. Per ulteriori informazioni, consulta [Sviluppo locale](#).

```
##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

Per una guida alla risoluzione dei problemi, vedere. [PathfindingSample il client della console non riesce a connettersi](#)

8. Premere CTRL+C per chiudere il client della console.

Linux console client

Important

Forniamo queste istruzioni per facilitare l'utente. Potrebbero non funzionare in alcuni Linux ambienti. Queste procedure non sono supportate.

Questa procedura presuppone che l'utente stia lavorando interamente all'interno di un Linux ambiente. È inoltre possibile visualizzare la simulazione utilizzando client integrati. Windows

Requisiti

- CMake3
- Ccompilatore (già incluso) Amazon Linux 2
- Git

Per connettersi all'applicazione di esempio con il client della console di esempio

1. Al prompt della Linux shell, vai alla cartella del client della console (nella cartella SDK dell'app).

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

2. Crea una cartella di compilazione.

```
mkdir build
```

3. Vai alla cartella build.

```
cd build
```

4. Usa CMake3 per creare il client.

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

Note

Assicurati di includere lo spazio e il periodo alla fine.

5. Esegui il client con l'indirizzo IP e il numero di porta del tuo ViewApp.

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

La finestra del prompt dei comandi dovrebbe mostrare i numeri per l'aggiornamento, l'eliminazione e la creazione di eventi dell'entità, simili all'output di esempio seguente.

Note

Gli indirizzi IP e i numeri di porta nell'output di esempio seguente sono segnaposto. Fornisci l'indirizzo IP e il numero di porta del tuo ViewApp al client della console. Fornisci il numero di Actual porta se desideri connetterti a un ViewApp sistema eseguito inCloud AWS. Fornisci l'indirizzo IP e il numero di porta 127.0.0. 1:7000 quando ti connetti a un sistema ViewApp che funziona sul tuo sistema locale. Per ulteriori informazioni, consulta [Sviluppo locale](#).

```
##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

Per una guida alla risoluzione dei problemi, vedere. [PathfindingSample il client della console non riesce a connettersi](#)

6. Premere CTRL+C per chiudere il client della console.

Unreal Engine on Windows

Requisiti

- Unreal Engine 5 ambiente di sviluppo
- Microsoft .NET Framework 4.8 Developer Pack
- Windowsclient di console (vedi la scheda client della Windows console in questa pagina)

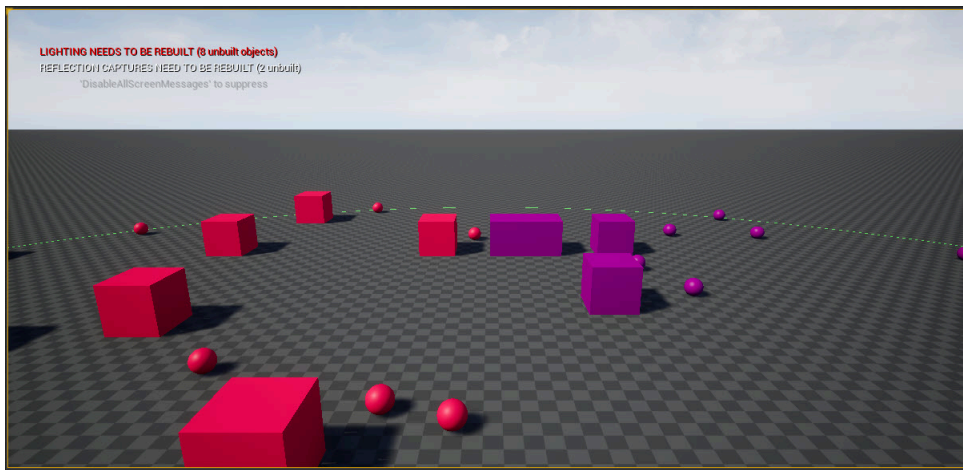
Important

Altre versioni di Unreal Engine e non .NET sono supportate e potrebbero causare problemi.

Per connettersi all'applicazione di esempio con il Unreal client di esempio

1. Il Unreal Engine client utilizza la NNG libreria del client della console. È necessario creare il client della console Windows se non l'hai già creato. Per ulteriori informazioni, consulta la scheda client della Windows console in questa pagina.
2. In una finestra di gestione dei file, vai a `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Aprire `UnrealClient.uproject`.
4. Se l'editor ti chiede se vuoi ricostruire i UnrealClient moduli, scegli yes.
5. In un editor di testo, apri `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`.
6. Aggiorna l'URL con l'indirizzo IP e il numero di porta dell'app View: `tcp://ip-address:port-number` (dovrebbe apparire così `tcp://198.51.100.135:1234`).
7. Nell'Unrealeditor, scegli play.

L'Unrealeditor dovrebbe mostrare una visualizzazione della simulazione, simile alla schermata seguente.



Note

A seconda della potenza del sistema di sviluppo locale, l'Unreal editor potrebbe impiegare alcuni minuti per visualizzare la simulazione. Durante questo periodo, il sistema potrebbe sembrare bloccato.

Usa i D tasti W AS,, per muoverti nel Unreal client. Tieni premuto il pulsante del mouse e trascina il mouse per ruotare.

Puoi premere il tasto [(parentesi quadra sinistra) per ridurre le dimensioni dell'area di iscrizione. Puoi premere il tasto] (parentesi quadra destra) per aumentare le dimensioni dell'area di sottoscrizione. La dimensione dell'area di sottoscrizione determina il numero di entità che appaiono nel client.

È possibile premere il C tasto per creare un'entità nella simulazione. Il client invia un CreateEntity comando all'app di visualizzazione. L'app di visualizzazione creerà quindi l'entità e la trasferirà nel dominio spaziale.

Puoi esaminare il codice per `ViewAppDriver::HandleEntityCreationRequests` [*project-folder*\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp](#) vedere come l'app implementa questo processo.


 Note

Se non conosci l'indirizzo IP e il numero di porta della tua app View, puoi utilizzare la [procedura del tutorial di avvio rapido](#) per ottenere tali informazioni.


Passaggio 12: interrompi e ripulisci la simulazione

È importante ripulire le simulazioni quando non ne hai più bisogno. SimSpace Weaver le risorse di simulazione vengono conteggiate ai fini delle quote (limiti) di servizio, anche se la simulazione viene interrotta. Continuerai a ricevere addebiti di fatturazione per le simulazioni in esecuzione. Potresti anche addebitare costi di fatturazione per l'archiviazione dei dati nei servizi di supporto, come Amazon CloudWatch Logs e Amazon Simple Storage Service. Per ulteriori informazioni sulle quote di SimSpace Weaver servizio, consulta [SimSpace Endpoint e quote Weaver](#)

Segui le procedure in questa sezione quando sei pronto per ripulire la simulazione.

 Important

Non è possibile riavviare una simulazione interrotta.

 Important

Non è possibile recuperare una simulazione eliminata.

Pulisci le risorse di simulazione in SimSpace Weaver

È necessario interrompere la simulazione prima di poterla eliminare. L'eliminazione di una simulazione rimuove solo le risorse. SimSpace Weaver È necessario eseguire passaggi separati per eliminare le risorse create dalla simulazione o utilizzate in altri servizi (vedere la sezione seguente).

Docker

Per ripulire la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del tuo progetto e della tua piattaforma. Stai `path\project-name` utilizzando i valori che hai fornito quando hai creato il progetto. `project-folder`

Al prompt dei Windows comandi, immettete:

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

2. Trova i nomi delle tue simulazioni.

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

3. Interrompi una simulazione.

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

4. Eliminare una simulazione interrotta.

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

WSL

 Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Per ripulire la simulazione

1. Se non ci sei già, vai alla cartella degli strumenti del progetto e della piattaforma. *project-folder* Stai *path/project-name* utilizzando i valori che hai fornito quando hai creato il progetto.

Al prompt della Linux shell, inserisci:

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

2. Trova i nomi delle tue simulazioni.

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

 Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'SDK dell'app. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

3. Interrompi una simulazione.

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

4. Eliminare una simulazione interrotta.

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

AWS Management Console

Per ripulire la simulazione

1. Apri la SimSpace Weaver console dalla [SimSpace Weaverconsole](#).
2. Dal riquadro di navigazione, scegli Simulations.
3. Dall'Simulationselecco, seleziona l'opzione accanto al nome della simulazione che desideri eliminare.
4. Se il Statusnome della simulazione selezionata è: STARTED
 - a. Scegli il Actionsmenu a discesa.
 - b. Scegli Stop.
 - c. Per confermare, inserisci il nome della simulazione.
 - d. Scegli Stop.
 - e. StatusAttendi che sia finita la simulazione. STOPPED
5. Scegli il Actionsmenu a discesa.
6. Scegli Delete.
7. Per confermare, scegli. Delete

Pulisci le risorse di simulazione nei servizi di supporto

Per supportare la simulazione, SimSpace Weaver crea risorse in altri servizi. SimSpace Weaver non elimina queste risorse quando elimini la simulazione. Puoi eliminare queste risorse aggiuntive se non ne hai bisogno.

Important

Potresti ricevere addebiti di fatturazione per ognuna di queste risorse che non elimini.

Per eliminare le risorse di supporto per il tuo progetto

1. Se hai finito con il progetto, elimina il relativo AWS CloudFormation stack. Per ulteriori informazioni sull'utilizzo di AWS CloudFormation, consulta [Eliminazione di uno stack sulla AWS CloudFormation console nella Guida per l'utente](#). AWS CloudFormation

- `weaver-nome-progetto -stack`

Important

Le simulazioni avviate dallo stesso progetto condividono risorse come il ruolo dell'app. Quando elimini lo AWS CloudFormation stack, elimini il ruolo dell'app. Non eliminare lo AWS CloudFormation stack se hai altre simulazioni che condividono le stesse risorse.

Note

Il tuo AWS CloudFormation stack verrà probabilmente segnalato DELETE_FAILED perché non può eliminare i bucket Amazon S3 che non sono vuoti. Eliminerai i bucket Amazon S3 nel passaggio successivo.

2. Se hai finito con il tuo progetto, elimina i relativi bucket Amazon S3. Per ulteriori informazioni sull'utilizzo dei bucket Amazon S3, consulta [Eliminazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

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

Ad esempio, il progetto indicato MyProject nell'account 111122223333 nella us-west-2 regione ha il seguente bucket:

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

Note

È necessario eliminare il contenuto di un bucket Amazon S3 prima di poter eliminare il bucket.

Note

SimSpace WeaverI progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- *weaver- - numero di conto -app-zips- regione lowercase-project-name*
- *weaver- - numero-conto -schemi- regione lowercase-project-name*

3. Se hai attivato la registrazione per la simulazione, elimina il gruppo Logs log. CloudWatch Per ulteriori informazioni sull'utilizzo dei CloudWatch log, consulta [Working with log groups and log stream](#) nella Amazon CloudWatch Logs User Guide.

Il nome del gruppo di log per la simulazione è specificato nel relativo schema (file di configurazione): *project-folder\tools\project-name.yaml*

Il nome del gruppo di log è il valore di `log_destination_resource_name`. Il seguente frammento di schema mostra che il gruppo di log per l'applicazione di esempio è.

MySimulationLogs

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

Warning

Se si avviano più simulazioni che specificano lo stesso gruppo di log, i dati di log per tutte le simulazioni verranno inseriti nello stesso gruppo di log. Se si elimina il gruppo di log, si eliminano i dati di registro per tutte le simulazioni che utilizzano quel gruppo di log. Se si elimina un gruppo di log per una simulazione in esecuzione, la simulazione avrà esito negativo.

⚠ Important

Se lo schema della simulazione è specifico `log_destination_service: "logs"` e un `log_destination_resource_name` ma non riesci a trovare il gruppo di log in CloudWatch Logs, assicurati di controllare lo stesso Regione AWS in cui è stata eseguita la simulazione.

Utilizzo di SimSpace Weaver

Questo capitolo fornisce informazioni e linee guida per aiutarti a creare le tue applicazioni in SimSpace Weaver.

Argomenti

- [Configurazione della simulazione](#)
- [Durata massima di un flusso di un flusso](#)
- [Sviluppo di app](#)
- [Sviluppo di applicazioni client](#)
- [Sviluppo locale](#)
- [AWS SimSpace Weaver SDK dell'app](#)
- [AWS SimSpace Weaver struttura dimostrativa](#)
- [Utilizzo delle quote di servizio](#)
- [Simulazioni di debug](#)
- [Contenitori personalizzati](#)
- [Uso di Python](#)
- [Support per altri motori](#)
- [Utilizzo di software concesso in licenza con AWS SimSpace Weaver](#)
- [Gestisci le tue risorse con AWS CloudFormation](#)
- [Snapshot](#)
- [Messaggistica](#)

Configurazione della simulazione

Uno schema (o schema) di simulazione è un file di testo in YAML formato elettronico che specifica la configurazione per una simulazione. È possibile utilizzare lo stesso schema per avviare più simulazioni. Il file dello schema si trova nella cartella del progetto per la simulazione. È possibile utilizzare qualsiasi editor di testo per modificare il file. SimSpace Weaver legge lo schema solo quando avvia la simulazione. Tutte le modifiche apportate a un file di schema influiscono solo sulle nuove simulazioni avviate dopo le modifiche.

Docker

Per configurare la simulazione, modifica il file dello schema di simulazione:

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

Lo schema di simulazione viene caricato quando si crea una nuova simulazione. Lo script di supporto per l'avvio rapido del progetto caricherà lo schema come parte del processo di creazione della simulazione:

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

Puoi anche utilizzare lo script di supporto allo schema di caricamento per il tuo progetto se non stai utilizzando lo script di avvio rapido per creare la tua simulazione:

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

WSL

Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

È possibile configurare la simulazione modificando il relativo file dello schema di simulazione:

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

Lo schema di simulazione viene caricato quando si crea una nuova simulazione. Lo script di supporto per l'avvio rapido del progetto caricherà lo schema come parte del processo di creazione della simulazione:

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

Puoi anche utilizzare lo script di supporto allo schema di caricamento per il tuo progetto se non stai utilizzando lo script di avvio rapido per creare la tua simulazione:

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

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o SimSpace Weaver successiva dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script SimSpace Weaver SDK con il parametro. AWS CLI --profile *cli-profile-name* Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

Parametri di configurazione della simulazione

Lo schema di simulazione contiene informazioni di bootstrap, tra cui:

- Proprietà di simulazione: [versione SDK e configurazione di calcolo \(tipo e numero di lavoratori\)](#)
- Orologi: frequenza di ticchettio e tolleranze
- Strategie di partizionamento spaziale: topologia spaziale (come una griglia), limiti e gruppi di posizionamento (raggruppamento delle partizioni spaziali sui lavoratori)
- Domini e relative app: app bucket, path e comandi di avvio

SimSpace Weaver utilizza la configurazione dello schema per configurare e disporre le partizioni spaziali, avviare app e far avanzare la simulazione alla frequenza specificata.

Note

Lo script di creazione del progetto nell'SDK dell'SimSpace Weaver app genererà automaticamente uno schema di simulazione basato sull'applicazione di esempio.

I seguenti argomenti descrivono i parametri dello schema di simulazione. Per una descrizione completa dello schema di simulazione, vedere. [SimSpace Weaver riferimento allo schema di simulazione](#)

Argomenti

- [Versione SDK](#)
- [Proprietà di simulazione](#)
- [Worker](#)
- [Orologio](#)
- [Strategie di partizionamento](#)
- [Domini](#)

Versione SDK

Il `sdk_version` campo specifica la versione per SimSpace Weaver cui è formattato lo schema. Valori validi: 1.16, 1.15, 1.14, 1.13, 1.12

Important

Il valore di `include sdk_version` solo il numero della versione principale e il numero della prima versione secondaria. Ad esempio, il valore 1.12 specifica tutte le versioni 1.12.x, ad esempio 1.12.01.12.1, e 1.12.2.

Proprietà di simulazione

La `simulation_properties` una sezione dello schema specifica la configurazione di registrazione e un tipo di dati per il campo indice (in genere la posizione spaziale) delle entità.

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

Il valore di `log_destination_service` determina l'interpretazione del valore di `log_destination_resource_name`. Attualmente, l'unico valore supportato è `logs`. Ciò

significa che il valore `ilog_destination_resource_name` è il nome di un gruppo di log in `AmazonCloudWatchRegistri`

Note

La registrazione è facoltativa. Se non configuri le proprietà di destinazione dei registri, la simulazione non produrrà registri.

È obbligatoria solo la proprietà `default_entity_index_key_type`. L'unico valore valido è `Vector3<f32>`.

Worker

La sezione `workers` specifica il tipo e il numero di lavoratori che desideri per la tua simulazione. SimSpace Weaver utilizza i propri tipi di worker mappati ai tipi di istanze Amazon EC2.

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

Abilitazione delle simulazioni con più operatori

È possibile creare una simulazione che utilizzi più di un lavoratore. Per impostazione predefinita, le simulazioni utilizzano 1 lavoratore. È necessario modificare lo schema di simulazione prima di iniziare la simulazione.

Note

Non puoi modificare una simulazione che è già iniziata. Se desideri abilitare multi-worker per una simulazione in esecuzione, devi prima interrompere ed eliminare la simulazione.

Per utilizzare più di un lavoratore, impostate il `desired` numero di istanze di calcolo con un valore maggiore di 1. Esiste un numero massimo di app per ogni lavoratore. Per ulteriori informazioni, vedere [SimSpace Endpoint e quote Weaver](#). SimSpace Weaver utilizzerà più di 1 lavoratore solo

quando il numero di app su un lavoratore supera questo limite. SimSpace Weaver può inserire un'app su uno qualsiasi dei lavoratori disponibili. Il posizionamento dell'app su un lavoratore specifico non è garantito.

Il seguente frammento di schema mostra una configurazione per una simulazione che richiede 2 lavoratori. SimSpace Weaver tenterà di allocare il secondo lavoratore se il numero di app supera il numero massimo di app per 1 lavoratore.

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

Orologio

La `clock` sezione specifica le proprietà dell'orologio di simulazione. Attualmente, è possibile configurare solo il tasso di cambio (il numero di tick al secondo che l'orologio invia alle app). La frequenza dei tick è la frequenza massima. La percentuale di tick effettiva potrebbe essere inferiore perché tutte le operazioni (come gli aggiornamenti delle entità) relative a un segno di spunta devono essere completate prima che possa iniziare il tick successivo. Il tick rate è anche chiamato tick rate frequenza.

I valori validi per `tick_rate` dipendono da `sdk_version` specificato nel tuo schema.

Valori validi per il tick rate

- Versioni precedenti a "1.14":
 - 10
 - 15
 - 30
- versione "1.14" o successiva:
 - "10"
 - "15"
 - "30"
 - "unlimited"

Per ulteriori informazioni, consulta [Tick rate illimitato](#).

Important

- Per schemi con `unsdk_version` prima di "1.14" il valore di `tick_rate` è un intero, ad esempio 30.
- Per schemi con `unsdk_version` di "1.14" o più tardi, il valore di `tick_rate` è una corda, ad esempio "30". Il valore deve includere le virgolette doppie.

Se converti una versione "1.12" o "1.13" da schema a versione "1.14" o più tardi, è necessario includere il valore di `tick_rate` tra virgolette doppie.

Tick rate illimitato

Puoi impostare il `tick_rate` a "unlimited" per consentire alla simulazione di funzionare alla stessa velocità di esecuzione del codice. Con una tariffa di selezione illimitata, SimSpace Weaver invia il segno di spunta successivo subito dopo che tutte le app hanno completato i commit per il segno di spunta corrente.

Important

La frequenza di selezione illimitata non è supportata in SimSpace Weaver versioni precedenti alla 1.14.0. Il valore minimo di `unsdk_version` nello schema è "1.14".

Tick rate illimitato in SimSpace Weaver Local

SimSpace Weaver Local implementa "unlimited" come se lo schema specificasse una frequenza di tick di 10 kHz (10000). L'effetto è lo stesso di una frequenza di tick illimitata in Cloud AWS. Specifica il `tick_rate`: "unlimited" nel tuo schema. Per ulteriori informazioni su SimSpace Weaver Local, consulta [Sviluppo locale](#).

Domande frequenti sull'orologio

D1. Posso modificare una simulazione STARTED per utilizzare un tick rate diverso?

Non è possibile modificare il tick rate di una simulazione già esistente in Cloud AWS in qualsiasi fase del suo ciclo di vita. Inoltre, non è possibile modificare la frequenza di tick di una simulazione in esecuzione in SimSpace Weaver Local. È possibile impostare il `tick_rate` nello schema e iniziare una nuova simulazione da quello schema.

D2. Posso eseguire la mia simulazione con un tick rate illimitato in una versione precedente alla 1.14?

No, il tick rate illimitato non è supportato nelle versioni precedenti alla 1.14.0.

Risoluzione degli errori dell'orologio

Se la simulazione non si avvia, puoi controllare il valore di "StartError" nell'output di DescribeSimulationAPI. Un non validotick_rateil valore nello schema produrrà i seguenti errori.

Note

L'output di errore mostrato qui viene visualizzato su più righe per migliorare la leggibilità. L'output di errore effettivo è una singola riga.

- Lasdk_versionè precedente a"1.14"e il valore ditick_rateè un numero intero non valido. Valori validi: 10, 15, 30

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

- IIsdk_versionè precedente a"1.14"e il valore ditick_rateè una stringa. Valori validi: 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\"}]"
```

- Lasdk_versionè"1.14"o più tardi e il valore ditick_rateè una stringa non valida. Valori validi: "10", "15", "30", "unlimited"

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

- Lasdk_versionè"1.14"o più tardi e il valore ditick_rateè un numero intero. Valori validi: "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\}]"

```

Strategie di partizionamento

La `partitioning_strategies` sezione specifica le proprietà di configurazione per le partizioni delle app spaziali. Fornisci il tuo nome per una strategia di partizionamento (un insieme di proprietà in questa sezione) e lo usi nella configurazione spaziale dell'app.

```

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

```

La `topology` proprietà specifica il tipo di sistema di coordinate utilizzato dalla simulazione. Il valore `Grid` specifica una griglia bidimensionale (2D).

Per un `Grid` topologia, lo spazio di simulazione è modellato come un riquadro di delimitazione allineato agli assi (AABB). Specificate i limiti delle coordinate per ciascun asse del vostro AABB nella `aabb_bounds` proprietà. Tutte le entità che esistono spazialmente nella simulazione devono avere una posizione all'interno dell'AABB.

Gruppi di posizionamento della griglia

Un gruppo di collocamento è una raccolta di partizioni spaziali di app che desideri `SimSpace Weaver` collocare sullo stesso lavoratore. Si specifica il numero e la disposizione dei gruppi di posizionamento (in una griglia) nel `grid_placement_groups` proprietà. `SimSpace Weaver` tenterà di distribuire uniformemente le partizioni tra i gruppi di posizionamento. Le aree di proprietà delle app spaziali con partizioni nello stesso gruppo di posizionamento saranno spazialmente adiacenti.

Ti consigliamo che $x * y$ sia uguale al numero di lavoratori desiderato. Se non è uguale, `SimSpace Weaver` cercherà di bilanciare i gruppi di collocamento tra i lavoratori disponibili.

Se non specifichi una configurazione del gruppo di posizionamento, SimSpace Weaverne calcolerà uno per te.

Domini

Fornisci un nome per un set di proprietà di configurazione per un dominio. L'impostazione di avvio per le app in un dominio determina il tipo di dominio:

- **launch_apps_via_start_app_call**— dominio personalizzato
- **launch_apps_by_partitioning_strategy**— dominio spaziale
- **launch_apps_per_worker**(non incluso nell'applicazione di esempio) — dominio del servizio

Important

SimSpace Weaversupporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

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

```
compute: 1
```

Note

SimSpace Weaver progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- `tessitore-lowercase-project-name-numero di conto-app zip-regione`
- `tessitore-lowercase-project-name-numero di conto-schemi-regione`

Argomenti

- [Configurazione dell'app](#)
- [Configurazione dei domini spaziali](#)
- [Endpoint di rete](#)
- [Configurazione dei domini di servizio](#)

Configurazione dell'app

Si specifica la configurazione di un'app (`app_config`) come parte della configurazione del relativo dominio. Tutti i tipi di domini utilizzano queste stesse proprietà di configurazione dell'app.

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

Note

SimSpace Weaver progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- `tessitore-lowercase-project-name-numero di conto-app zip-regione`
- `tessitore-lowercase-project-name-numero di conto-schemi-regione`

La proprietà `package` specifica l'URI S3 di un file zip in un bucket S3. Il file zip contiene l'eseguibile dell'app (chiamato anche binario) e qualsiasi altra risorsa di cui ha bisogno (come le librerie). Ogni istanza dell'eseguibile dell'app viene eseguita in un Docker contenitore su un lavoratore.

La proprietà `launch_command` specifica il nome dell'eseguibile e tutte le opzioni della riga di comando per eseguire l'app. Il valore di `launch_command` è un array. Ogni token dell'intera stringa di comando di avvio è un elemento dell'array.

Esempio

- Per il comando di avvio: `MyTestApp --option1 value1`
- Specifica: `launch_command: ["MyTestApp", "-option1", "value1"]`

La proprietà `required_resource_units` specifica il numero di unità di risorse di calcolo che SimSpace Weaver dovrebbe assegnare a questa app. Un'unità di risorse di calcolo è una quantità fissa di capacità di elaborazione (vCPU) e memoria (RAM) su un lavoratore. È possibile aumentare questo valore per aumentare la potenza di calcolo disponibile per l'app quando viene eseguita su un operatore. Esiste un numero limitato di unità di risorse di calcolo per ogni lavoratore. Per ulteriori informazioni, consulta [SimSpace Endpoint e quote Weaver](#).

Configurazione dei domini spaziali

Per i domini spaziali, è necessario specificare `partitioning_strategy`. Il valore di questa proprietà è il nome che hai dato a una strategia di partizionamento definita in un'altra parte dello schema.

```
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 progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- tessitore-*lowercase-project-name-numero di conto*-app zip-*regione*
- tessitore-*lowercase-project-name-numero di conto*-schemi-*regione*

Una strategia di partizionamento con unGridla topologia (l'unica topologia supportata in questa versione) dirigeSimSpace Weaverper disporre le partizioni spaziali delle app di questo dominio in una griglia. Lagrid_partitionproprietà specifica il numero di righe e colonne della griglia di partizione.

SimSpace Weaveravvierà 1 istanza dell'app spaziale per ogni cella nella griglia delle partizioni. Ad esempio, se un dominio spaziale hagrid_partitionvalorix: 2ey: 2, ci sono $2 \times 2 = 4$ partizioni nel dominio spaziale.SimSpace Weaveravvierà 4 istanze dell'app configurate nel dominio spaziale e assegnerà 1 partizione a ciascuna istanza dell'app.

Argomenti

- [Requisiti di risorse per i domini spaziali](#)
- [Domini spaziali multipli](#)
- [Domande frequenti sui domini spaziali](#)
- [Risoluzione dei problemi relativi ai domini spaziali](#)

Requisiti di risorse per i domini spaziali

Puoi assegnare fino a 17 unità di risorse di calcolo per ogni lavoratore. Si specifica il numero di unità di risorse di calcolo utilizzate da ciascuna app spaziale nelapp_configsezione del tuo dominio spaziale.

Example frammento di schema che mostra le unità di risorse di calcolo per un'app spaziale

```
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
```

Per calcolare il numero di unità di risorse di calcolo richieste da un dominio, moltiplica il numero di celle nella griglia (nel tuo `grid_partition,x*y`) in base al numero di unità di risorse di calcolo assegnate alle app spaziali.

Nell'esempio precedente, il dominio `MySpatialDomain` specifica:

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

La griglia per `MySpatialDomain` ha $2 * 2 = 4$ celle. Il dominio spaziale richiede $4 * 1 = 4$ unità di risorse di calcolo.

Il numero totale di unità di risorse di calcolo per tutti i domini specificati nello schema deve essere inferiore o uguale ad `desired_number` di lavoratori moltiplicato per il numero massimo di unità di risorse di calcolo per ogni lavoratore (17).

Domini spaziali multipli

Puoi configurare la tua simulazione per utilizzare più di un dominio spaziale. Ad esempio, puoi utilizzare 1 dominio spaziale per controllare gli attori principali di una simulazione (come persone e automobili) e un dominio spaziale diverso per controllare l'ambiente.

Puoi anche utilizzare più domini spaziali per assegnare risorse diverse a diverse parti della simulazione. Ad esempio, se la simulazione ha un tipo di entità che ha 10 volte più istanze di entità rispetto a un altro tipo, puoi creare domini diversi per gestire ogni tipo di entità e allocare più risorse per il dominio con più entità.

Important

SimSpace Weaver le versioni precedenti alla 1.14.0 non supportano più domini spaziali.

⚠ Important

AWS SimSpace Weaver Local attualmente non supporta più domini spaziali. Per ulteriori informazioni su SimSpace Weaver Local, consulta [Sviluppo locale](#).

⚠ Important

SimSpace Weaversupporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

Configurazione di più domini spaziali

Per configurare più di un dominio spaziale, aggiungi le altre definizioni del dominio spaziale come sezioni denominate separate nel tuo schema. Ogni dominio deve specificare il `launch_apps_by_partitioning_strategy` chiave. Vedi il seguente schema di esempio.

```
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

```

Unire i domini spaziali

In alcuni scenari, potresti voler posizionare le partizioni per un dominio spaziale sui worker accanto alle partizioni di un altro dominio. Ciò può migliorare le caratteristiche prestazionali se tali partizioni creano sottoscrizioni tra domini tra domini reciproci.

Aggiungi la chiave di primo livello `placement_constraints` al tuo schema per specificare quali domini SimSpace Weaver devono essere messi insieme. Il richiested `workers` la chiave deve fare riferimento a un nome `workers` configurazione nello schema.

Example frammento di schema che mostra i domini spaziali messi insieme

```

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

```

Important

- Se utilizzi gruppi di collocamento:
 - Assicurati che $x * y$ sia un multiplo del numero di lavoratori.

- Assicurati che i valori dei gruppi di posizionamento siano divisori comuni per le dimensioni della griglia dei domini che metti insieme.
- Se non usare gruppi di collocamento:
 - Assicurati che 1 asse delle griglie del dominio spaziale abbia un divisore comune uguale al numero di lavoratori.

Per ulteriori informazioni sui gruppi di collocamento, vedere [Strategie di partizionamento](#).

Domande frequenti sui domini spaziali

D1. Come posso aggiungere un altro dominio spaziale a una simulazione esistente?

- Per una simulazione di corsa— Non è possibile modificare la configurazione di una simulazione in esecuzione. Modifica la configurazione del dominio nello schema, carica lo schema e le zip dell'app e avvia una nuova simulazione.
- Per una nuova simulazione— Aggiungi la configurazione del dominio allo schema, carica lo schema e le zip dell'app e avvia la nuova simulazione.

Risoluzione dei problemi relativi ai domini spaziali

È possibile che venga visualizzato il seguente errore quando si tenta di avviare la simulazione ma la configurazione del dominio non è valida.

```
"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}"]"
```

Cause potenziali

- Lo schema alloca più unità di risorse di calcolo per le app di quante ne siano disponibili per i lavoratori.
- SimSpace Weaver non riesce a determinare una disposizione per mettere insieme i domini sui lavoratori. Ciò si verifica quando si specificano più domini spaziali ma non esiste un divisore o un multiplo comune tra le griglie di dominio, ad esempio tra una griglia 2x4 e una griglia 3x5).

Endpoint di rete

Le app personalizzate e di servizio possono disporre di endpoint di rete a cui i client esterni possono connettersi. Si specifica un elenco di numeri di porta come valore `peringress_portsnelendpoint_config`. Questi numeri di porta sono sia TCP che UDP. L'app personalizzata o di servizio deve essere associata ai numeri di porta specificati in `ingress_ports`. SimSpace Weaver alloca dinamicamente i numeri di porta in fase di esecuzione e associa queste porte alle porte dinamiche. Puoi chiamare il `describe-app` API dopo che le app hanno iniziato a trovare i numeri di porta dinamici (effettivi). Per ulteriori informazioni, vedere [Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#) dal tutorial di avvio rapido.

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

Note

SimSpace Weaver i progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:

- `tessitore-lowercase-project-name-numero di conto-app zip-regione`
- `tessitore-lowercase-project-name-numero di conto-schemi-regione`

Note

`endpoint_config` è una proprietà opzionale per app personalizzate e app di servizio. Se non specifichi un `endpoint_config` quindi l'app non avrà un endpoint di rete.

Configurazione dei domini di servizio

La presenza di `launch_apps_per_worker` in una configurazione di dominio indica che si tratta di un dominio di servizio con app di servizio. SimSpace Weaver avvia e interrompe le app di servizio per te. Quando SimSpace Weaver avvia e interrompe un'app, si ritiene che l'app abbia un ciclo di vita gestito. SimSpace Weaver attualmente supporta l'avvio di 1 o 2 app di servizio per ogni lavoratore.

Example Esempio di dominio configurato per avviare 1 app di servizio su ogni lavoratore

```
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 Esempio di dominio configurato per avviare 2 app di servizio su ogni lavoratore

```
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
```

Durata massima di un flusso di un flusso

Ogni simulazione in AWS SimSpace Weaver ha un'impostazione di durata massima che specifica il tempo massimo in cui la simulazione può essere eseguita. La durata massima viene fornita come parametro quando si avvia una simulazione. L'[interfaccia di programmazione dell'API StartSimulation](#) ha un parametro opzionale `MaximumDuration`. Il valore del parametro è un numero di minuti (m o M), ore (h o H) o giorni (d o D). Ad esempio, `1h` o `1H` significa 1 ora. SimSpace Weaver mette di creare un flusso di flusso quando raggiunge questo limite.

Valore massimo

Il valore valido più alto per `MaximumDuration` è `14D` o il suo equivalente in ore (`336H`) o minuti (`20160M`).

Valore predefinito

Il parametro `MaximumDuration` è facoltativo. Se non fornisci un valore, SimSpace Weaver utilizza un valore di `14D`.

Valore minimo

Il valore valido più basso per `MaximumDuration` è un valore numericamente equivalente a `0`. Ad esempio, i valori `0M0H`, `e0D`, sono tutti numericamente equivalenti a `0`.

Se fornisci il valore minimo per la durata massima, la simulazione passa immediatamente allo `STOPPING` stato non appena raggiunge lo `STARTED` stato.

Avvio di una simulazione utilizzando gli script SDK SimSpace Weaver dell'app

È possibile fornire un valore per il `maximum-duration` parametro quando si utilizza uno dei seguenti script per avviare una simulazione:

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

Ogni script trasmette il valore di `maximum-duration` all'API `StartSimulation`.

⚠ Important

Se non fornisci un valore per `maximum-duration`, SimSpace Weaver utilizza il [valore predefinito](#) (14D).

Avvio di una simulazione utilizzando la console

È possibile fornire un valore per la durata massima quando si avvia una simulazione nella [SimSpace Weaver console](#). Inserisci il valore nel campo Durata massima del modulo Impostazioni di simulazione quando scegli Avvia simulazione.

⚠ Important

Se non fornisci un valore per la durata massima, SimSpace Weaver utilizza il [valore predefinito](#) (14D).

Lo stato di una simulazione che raggiunge la sua durata massima

Quando interrompe SimSpace Weaver automaticamente una simulazione che raggiunge la sua durata massima, lo stato della simulazione è `STOPPING` (se in corso) o `STOPPED`. Nella [SimSpace Weaver console](#), lo stato di destinazione della simulazione è fisso `STARTED`, poiché quello era l'ultimo stato richiesto da un utente.

Sviluppo di app

SimSpace Weaver lo sviluppo richiede un Amazon Linux 2 (AL2) ambiente per creare app perché le simulazioni vengono eseguite Amazon Linux in AWS Cloud. Se lo utilizzi Windows, puoi utilizzare gli script nell'SDK dell'SimSpace Weaver app per creare e avviare un Docker contenitore che viene eseguito AL2 con le dipendenze necessarie per creare SimSpace Weaver app. È inoltre possibile avviare un AL2 ambiente utilizzando Windows Subsystem for Linux (WSL) o utilizzare un AL2 sistema nativo. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

Note

Indipendentemente da come configuri il tuo ambiente di sviluppo locale, le tue app vengono eseguite in Docker contenitori quando le carichi per essere eseguite in Cloud AWS. Le app non hanno accesso diretto al sistema operativo host.

Flusso generale di un SimSpace Weaver app

1. Crea un'applicazione di .
2. Ciclo:
 - a. Inizia l'aggiornamento creando un `Transaction`.
 - Esci dal ciclo se la simulazione si sta chiudendo.
 - b. Elabora gli eventi relativi alla sottoscrizione e all'entità proprietaria.
 - c. Aggiorna la simulazione.
 - d. Impegna `Transaction` a terminare l'aggiornamento.
3. Distruggi l'applicazione.

App spaziali

Ogni app spaziale ha un'area di proprietà che è un'area spaziale del mondo della simulazione. Le entità situate nell'area di proprietà di un'app spaziale vengono archiviate nella partizione assegnata all'app. La singola app spaziale ha la piena proprietà (autorizzazioni di lettura e scrittura) su tutte le entità all'interno della partizione assegnata. Nessun'altra app può scrivere su tali entità. L'app spaziale fa avanzare lo stato delle sue entità. Ogni app spaziale possiede solo 1 partizione. SimSpace Weaver utilizza la posizione spaziale di un'entità per indicizzarla e assegnarla a una partizione spaziale dell'app.

L'SimSpace Weaver SDK fornisce un'applicazione di esempio. Puoi trovare il codice sorgente dell'app spaziale dell'applicazione di esempio nella seguente cartella:

Docker

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

WSL

Important

Forniamo queste istruzioni per comodità dell'utente. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

App personalizzate

Crea e usa app personalizzate per interagire con la simulazione.

Le app personalizzate possono

- Creazione di entità
- Sottoscrizione ad altre partizioni
- Esegui il commit delle

Flusso generale di un'app personalizzata

1. Crea un'applicazione di .
2. Abbonati a una regione specifica della simulazione:
 - a. Crea un `Transaction` per iniziare il primo aggiornamento.
 - b. Crea un abbonamento per la regione specifica.
 - c. Impegna `Transaction` a terminare il primo aggiornamento.
3. Ciclo:
 - a. Crea un `Transaction` file per iniziare l'aggiornamento.
 - Esci dal ciclo se la simulazione si sta chiudendo.
 - b. Modifiche allo stato del processo.
 - c. Impegna `Transaction` a terminare l'aggiornamento.

4. Distruggi l'applicazione.

Dopo che un'app personalizzata ha creato un'entità, deve trasferirla in un dominio spaziale affinché l'entità possa esistere spazialmente all'interno della simulazione. SimSpace Weaver utilizza la posizione spaziale dell'entità per posizionare l'entità nella partizione spaziale dell'app appropriata. L'app personalizzata che ha creato l'entità non può aggiornare o eliminare l'entità dopo averla trasferita in un dominio spaziale.

L'SimSpace Weaver app SDK fornisce un'applicazione di esempio. È possibile utilizzare le app personalizzate incluse nell'applicazione di esempio come modelli per le proprie app personalizzate. Puoi trovare il codice sorgente dell'app di visualizzazione (un'app personalizzata) dell'applicazione di esempio nella seguente cartella:

Docker

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

WSL

Important

Forniamo queste istruzioni per comodità dell'utente. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

Sviluppo di applicazioni client

Alcuni dei motivi per cui potresti voler connettere un client a una simulazione includono:

- Inserisci informazioni sul traffico in tempo reale in una simulazione su scala urbana.
- Crea human-in-the-loop simulazioni, in cui un operatore umano controlla alcuni aspetti della simulazione.
- Consenti agli utenti di interagire con la simulazione, ad esempio per una simulazione di allenamento.

Le app personalizzate in questi esempi fungono da interfaccia tra lo stato della simulazione e il mondo esterno. I client si connettono alle app personalizzate per interagire con la simulazione.

SimSpace Weaver non gestisce le applicazioni client e la loro comunicazione con le app personalizzate. Sei responsabile della progettazione, della creazione, del funzionamento e della sicurezza delle applicazioni client e della loro comunicazione con le tue app personalizzate. SimSpace Weaver espone solo un indirizzo IP e un numero di porta per ciascuna delle tue app personalizzate in modo che i client possano connettersi ad esse.

L'SimSpace Weaver app SDK fornisce client per la sua applicazione di esempio. È possibile utilizzare questi client come modelli per le proprie applicazioni client. È possibile trovare il codice sorgente per i client dell'applicazione di esempio nella seguente cartella:

Docker

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

WSL

Important

Sono anche presenti qui per comodità dell'utente. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

Per ulteriori informazioni sulla creazione e sull'utilizzo dei client applicativi di esempio, consulta il tutorial [Fase 5: Visualizza la simulazione](#) di avvio rapido in questa guida.

Sviluppo locale

Puoi distribuire SimSpace Weaver le tue applicazioni localmente per test e debug rapidi. SimSpace Weaver Local è supportato solo per la creazione. Microsoft Windows

⚠ Important

Per informazioni sullo sviluppo con Unity and Unreal Engine, consulta [Support per altri motori](#).

⚠ Important

Se stai lavorando nella versione 1.15.3 con C++,,, oppure Python Unity Unreal Engine, vedi. [Differenze di sviluppo locale nella versione 1.15.3](#)

Requisiti

- Microsoft Windows 10o superiore
- [Microsoft Visual Studio 2019](#)o più tardi, con il [Desktop development with C++](#)carico di lavoro installato

Argomenti

- [Crea la tua simulazione per SimSpace Weaver Local](#)
- [Esegui la tua simulazione con SimSpace Weaver Local](#)
- [Visualizza la tua simulazione locale](#)
- [Interrompi la simulazione locale](#)
- [Esegui il debug di simulazioni locali](#)
- [Differenze di sviluppo locale nella versione 1.15.3](#)

Crea la tua simulazione per SimSpace Weaver Local

Per imparare a SimSpace Weaver Local usarla, puoi usare la stessa applicazione Pathfinding Sample che hai eseguito nel cloud durante [Nozioni di base su SimSpace Weaver](#) i tutorial, ma questa volta sul tuo hardware locale.

Per creare l'applicazione di esempio per SimSpace Weaver Local

1. Al prompt dei comandi, vai `project-folder\tools\local`.
2. Esegui `generate_visual_studio_project.bat`.

3. Apri `project-folder\buildlocal\PathfindingSampleLocal.sln` in Visual Studio.
4. Imposta la configurazione di build su RelWithDebInfo.
5. Scegli Build> Build Solution.

Visual Studio inserirà i tuoi artefatti da costruzione in:

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

All'interno di quella cartella, dovresti vedere i seguenti file eseguibili:

- `PathfindingSampleLocalSpatial.exe`
- `PathfindingSampleLocalView.exe`

Esegui la tua simulazione con SimSpace Weaver Local

Puoi utilizzarla SimSpace Weaver Local per eseguire qualsiasi combinazione di un massimo di 24 app spaziali o personalizzate sul tuo computer locale. L'orologio di simulazione inizia dopo l'avvio di tutte le app spaziali definite nello schema.

Per eseguire le tue app con SimSpace Weaver Local

1. In una finestra di selezione dei file, vai a `project-folder\buildlocal\out\RelWithDebInfo`.
2. SimSpace Weaver Localle app richiedono un file di schema denominato `schema.yaml` nella directory di lavoro delle app. Qualsiasi mancata lettura delle informazioni richieste dallo schema interrompe le app.

Lo schema di SimSpace Weaver Local non deve essere necessariamente identico a `project-folder\tools\project-name-schema.yaml`, ma è possibile utilizzarlo come punto di partenza.

Scegli una delle seguenti opzioni:

- Copia quello schema in `project-folder\buildlocal\out\RelWithDebInfo\schema.yaml`.
- Imposta la variabile `WEAVERLOCAL_SCHEMA_PATH` di ambiente sul nome di un file di schema con un percorso o un nome di file diverso.

Example Esempio

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

Note

Se imposti la variabile di ambiente dalla riga di comando, quella variabile di ambiente (con quel valore) è accessibile solo da quella sessione del prompt dei comandi (finestra della console).

3. Lo schema per l'applicazione di esempio definisce una griglia 2x2, che crea 4 partizioni. Verrà eseguito uno script che avvierà 4 istanze dell'app spaziale, in modo che corrisponda al numero di app spaziali specificato nello schema. Lo script avvierà anche 1 app di visualizzazione. Dopo l'avvio di tutte le app spaziali e l'assegnazione di una partizione, la simulazione inizierà automaticamente a funzionare.

Per avviare le tue app

- a. Al prompt dei comandi, accedi alla cartella degli strumenti locale del tuo progetto.

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

- b. Esegui lo script per avviare le app.

```
launch_simulation_locally.bat
```

Note

Se si imposta WEAVERLOCAL_SCHEMA_PATH il nome di un file di schema, è necessario avviare le app spaziali dalla riga di comando nella stessa sessione (finestra) in cui è stata impostata la variabile di ambiente.

⚠ Important

Se viene visualizzato un pop-up di sicurezza di Windows, scegli Allow Access in modo da poterti connettere all'app di visualizzazione per visualizzare la simulazione.

ℹ Note

Puoi anche avviare le app spaziali e visualizzarle manualmente. Per fare ciò, devi avviare manualmente 4 istanze dell'app spatial e un'app di visualizzazione.

- App spaziale: `start PathfindingSampleLocalSpatial.exe`
- Visualizza l'app: `start PathfindingSampleLocalView.exe`

Visualizza la tua simulazione locale

Per visualizzare la simulazione locale, puoi utilizzare uno qualsiasi dei client inclusi in.

SimSpaceWeaverAppSdkDistributable Per ulteriori informazioni sulla creazione e l'utilizzo dei client di esempio, consulta il tutorial [Fase 5: Visualizza la simulazione](#) di avvio rapido.

È necessario aggiornare l'indirizzo IP e il numero di porta nel client per connettersi all'app View per la simulazione locale. Usa sempre i seguenti valori con SimSpace Weaver Local:

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

A seconda del client selezionato, è possibile aggiornare l'indirizzo IP e il numero di porta come segue:

- Unreal: modifica l'URL sulla riga 1 di `view_app_url.txt`
- Console: avvia il client con l'indirizzo IP e il numero di porta (URL) come parametro

Interrompi la simulazione locale

La simulazione locale continuerà a funzionare se le app spaziali locali sono attive. Se chiudi una delle finestre dell'app spaziale, interromperai l'intera simulazione. Chiudi tutte le altre finestre per ripulire il resto della simulazione.

Puoi chiudere manualmente ogni finestra dell'app oppure puoi usare il seguente script per chiuderle tutte automaticamente:

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

Note

Anche se la chiusura di una finestra spaziale dell'app interrompe la simulazione, assicurati di chiudere le altre finestre dell'app. Non sarai in grado di avviare correttamente un'altra simulazione locale se sono ancora aperte le finestre di una simulazione precedente.

Eseguire il debug di simulazioni locali

Puoi eseguire il debug delle tue SimSpace Weaver Local app con Microsoft Visual Studio [Per ulteriori informazioni su come eseguire il debug con Visual Studio, consulta Microsoft Visual Studio documentation](#)

Per eseguire il debug della simulazione locale

1. Assicurati che `schema.yaml` sia nella tua directory di lavoro.
2. In Visual Studio, apri il menu contestuale per ogni app di cui desideri eseguire il debug (ad esempio `PathfindingSampleLocalSpatial` o `PathfindingSampleLocalView`) e imposta la directory di lavoro nella sezione di debug.
3. Apri il menu contestuale dell'app di cui desideri eseguire il debug e seleziona `Imposta come progetto di avvio`.
4. Scegli `F5` di avviare il debug dell'app.

I requisiti per eseguire il debug di una simulazione sono gli stessi necessari per eseguire una simulazione normalmente. È necessario avviare il numero di app spaziali specificato nello schema. Ad esempio, se lo schema specifica una griglia 2x2 e si avvia un'app spaziale in modalità debug, la

simulazione non verrà eseguita finché non si avviano altre 3 app spaziali (in modalità debug o non in modalità debug).

Per eseguire il debug di un'app personalizzata, devi prima avviare le app spaziali e quindi avviare l'app personalizzata nel debugger.

Tieni presente che la simulazione viene eseguita in blocco. Non appena un'app raggiunge un punto di interruzione, tutte le altre app verranno messe in pausa. Dopo aver proseguito da quel punto di interruzione, le altre app continueranno.

Differenze di sviluppo locale nella versione 1.15.3

Questa sezione descrive le modifiche nello sviluppo a SimSpace Weaver Local partire dalla versione 1.15.3. Queste modifiche influiscono sui flussi di lavoro per i SimSpace Weaver Local progetti in C++, ePython. Unity Unreal Engine

Argomenti

- [Modifiche ai file](#)
- [Aggiorna un progetto C++ esistente alla versione 1.15.3 SimSpace Weaver Local](#)
- [Esegui un nuovo progetto Python in 1.15.3 SimSpace Weaver Local](#)
- [Aggiorna un progetto Unity esistente alla versione SimSpace Weaver Local 1.15.3](#)
- [Aggiorna un progetto Unreal Engine esistente alla versione 1.15.3 SimSpace Weaver Local](#)
- [Domande frequenti sulla versione 1.15.3 SimSpace Weaver Local](#)

Modifiche ai file

- SimSpaceWeaverAppSdkLocal i file di libreria sono ora denominati `weaver_app_sdk_cxx_v1_full_local`.
 - Puoi trovare questi file in `sdk-folder\SimSpaceWeaverAppSdk-1.15.3\lib\weaverlocal\windows`.
- I tuoi progetti non devono più collegarsi o includere `SimSpaceWeaverAppSDK-1.15.1\include\aws\weaverruntime\local_ffi`.
- Gli script Python SimSpace Weaver Local non richiedono più. `cmake`
- Esistono script Python nuovi o rinominati per: SimSpace Weaver Local
 - `build-local`— Crea l'app Python per il lancio. Mette tutto il codice Python e i file della SimSpace Weaver Local libreria nella `buildlocal` directory.

- `local-config`— Definisce le variabili di ambiente per gli script locali.
- `start-python-locally`— Configura l'app PythonPath e avvia Python.
- I seguenti script sono equivalenti agli script cloud con nomi simili:
 - `quick-start-local`
 - `start-simulation-local`
 - `stop-simulation-local`

Aggiorna un progetto C++ esistente alla versione 1.15.3 SimSpace Weaver Local

Requisiti

- [SDK dell'app versione 1.15.3 SimSpace Weaver](#)
- Un progetto C++ esistente per SimSpace Weaver Local la versione 1.15.2 o precedente

Per aggiornare il tuo progetto

1. Assicurati che il tuo progetto includa l'intestazione richiesta:

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

Note

Di solito è incluso in `aws\weaverruntime\detail.h`

2. Assicurati che il tuo progetto non includa la seguente intestazione obsoleta:

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

Note

Di solito è incluso in `aws\weaverruntime\detail.h`

3. Nel file CMake o nello script di build, sostituisci i nomi delle librerie `SimSpaceWeaverAppSdkLocal` statiche con il nuovo `weaver_app_sdk_cxx_v1_full_local` nome.

4. Se non l'hai eseguito `sdk-folder\docker-create-image.bat` dalla versione 1.15.3, eseguila ora. Devi farlo solo 1 volta.
5. Segui le [normali procedure](#) per creare ed eseguire il tuo progetto.

Risoluzione dei problemi

Vengono visualizzati errori del linker (simbolo esterno non risolto)

Il compilatore emette errori del linker come i seguenti (interruzioni di riga aggiunte per motivi di leggibilità).

```
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
```

Per risolvere il problema

- Assicurati che il tuo progetto non includa `<aws/weaverruntime/local_ffi/Bridge.h>`.

Viene visualizzato un errore di incompatibilità dei tipi

È possibile che venga visualizzato un errore come il seguente (interruzioni di riga aggiunte per motivi di leggibilità).

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

Per risolvere il problema

1. Aggiorna i tuoi file di SimSpace Weaver installazione con le versioni 1.15.3.
2. Assicurati che qualsiasi utilizzo di `rust::cxxbridge1::String` sia simile al seguente:

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

Esegui un nuovo progetto Python in 1.15.3 SimSpace Weaver Local

Requisiti

- [SDK dell'app versione 1.15.3 SimSpace Weaver](#)
- Una nuova SimSpace Weaver Local versione del progetto 1.15.3 PythonBubblesSample

Per eseguire il tuo progetto

1. In una finestra del prompt dei comandi, vai `project-folder\tools\local\windows`.
2. Esegui `quick-start-local.bat`

4 app spaziali, 1 app di visualizzazione e 1 client devono essere avviate localmente. Il client dovrebbe visualizzare delle bolle.

Risoluzione dei problemi

Viene visualizzato un errore che dice che **GLIBCXX** non è stato trovato

Il seguente errore indica l'impossibilità di importare l'SDK dell'app SimSpace Weaver Python. Una causa probabile è che le tue librerie C++ non sono aggiornate.

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

Per risolvere il problema

1. Aggiorna il codice sorgente delle tue `libstdc++` librerie (ad esempio `gcc omsvc`) a una versione che includa la versione specificata di `GLIBCXX` (nell'esempio fornito, la versione è `3.4.29`).
2. Assicuratevi che la variabile di ambiente di sistema `LD_LIBRARY_PATH` sia impostata sul percorso corretto di `lib64`

Lo script di avvio rapido ha esito negativo

Lo script di avvio rapido fallisce con un messaggio come il seguente.

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

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

Per risolvere il problema

1. Vai a *project-folder*\tools\local\windows e modifica `local-config.bat`.
2. Assicurati che le seguenti variabili di ambiente siano impostate sui percorsi corretti per il tuo sistema locale:
 - `TOOLS_DIR` è impostato sul tuo *project-folder*\tools
 - `TOOLS_DIR_WINDOWS` è impostato sul tuo *project-folder*\tools\local\windows
 - `PROJECT_ROOT` è impostato sul tuo *project-folder*
 - `BUILD_DIR` è impostato sul tuo *project-folder*\buildlocal
 - `APP_SDK_DIR` è impostato sul tuo *sdk-folder*

Aggiorna un progetto Unity esistente alla versione SimSpace Weaver Local 1.15.3

Requisiti

- [SDK dell'app versione 1.15.3 SimSpace Weaver](#)
- Un progetto Unity esistente per la SimSpace Weaver Local versione 1.15.2 o precedente

Per aggiornare il tuo progetto

1. In Unity, rimuovi il pacchetto AWS SimSpace Weaver esistente.
 - a. Apri il tuo progetto Unity esistente.
 - b. Nella finestra dell'editor, scegliete Finestra > Package Manager.
 - c. In Pacchetti - Unity Technologies, seleziona AWS SimSpace Weaver e scegli Rimuovi.
2. Nella tua versione *sdk-folder* 1.15.3, esegui `download-unity-package.bat`
3. Segui le istruzioni `Unity_SDK_for_AWS_SimSpace_Weaver.pdf` per aggiungere il pacchetto appena scaricato `SimSpaceWeaverUnityPackage.zip` come pacchetto AWS SimSpace Weaver nell'editor Unity.

Aggiorna un progetto Unreal Engine esistente alla versione 1.15.3 SimSpace Weaver Local

Requisiti

- [SDK dell'app versione 1.15.3 SimSpace Weaver](#)
- Un progetto Unreal Engine esistente per SimSpace Weaver Local la versione 1.15.2 o precedente

Per aggiornare il tuo progetto

1. Chiudi tutte le finestre di Unreal Project e Code Editor.
2. In una finestra del prompt dei comandi, vai alla tua versione 1.15.3. *sdk-folder*
3. Esegui `update-unreal-project.bat --path project-folder --name project-name`.

Note

Questo sostituisce il plug-in esistente con il nuovo plug-in. Tutte le modifiche vengono cancellate.

4. Segui le istruzioni `AWS_SimSpace_Weaver_Unreal_Guide.pdf` per creare il tuo progetto per SimSpace Weaver Local

Risoluzione dei problemi

Mancata eliminazione dei file del plugin esistenti

Potresti ricevere errori simili ai seguenti.

```
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
```

Per risolvere il problema

1. Assicurati che la finestra dell'editor del progetto Unreal e gli editor di codice siano chiusi.
2. Esegui `update-unreal-project.bat`.

Domande frequenti sulla versione 1.15.3 SimSpace Weaver Local

- Q1: Come posso modificare le mie variabili di SimSpace Weaver Local ambiente per il mio progetto Python?
 - Modifica `project-folder\tools\windows\local-config` per il tuo progetto Python.

AWS SimSpace WeaverSDK dell'app

L'SDK SimSpace Weaver dell'app fornisce API che puoi utilizzare per controllare le entità della simulazione e rispondere agli eventi. SimSpace Weaver Include il seguente namespace:

- API: definizioni principali dell'API e del suo utilizzo

Collegamento con la seguente libreria:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

La libreria è disponibile per il collegamento dinamico quando esegui le app in. Cloud AWS
Non è necessario caricarla con le app.

Note

Le API SDK SimSpace Weaver dell'app controllano i dati all'interno della simulazione. Queste API sono separate dalle API di SimSpace Weaver servizio, che controllano le risorse del SimSpace Weaver servizio (come simulazioni, app e orologi) in. AWS Per ulteriori informazioni, consulta [SimSpace WeaverRiferimenti API](#).

Argomenti

- [I metodi API restituiscono un Result](#)
- [Interazione con l'SDK dell'app al livello più alto](#)
- [Gestione della simulazione](#)
- [Sottoscrizioni](#)
- [Entità](#)
- [Eventi dell'entità](#)
- [Resulte gestione degli errori](#)
- [Generici e tipi di dominio](#)
- [Operazioni varie dell'SDK dell'app](#)

I metodi API restituiscono un Result

La maggior parte delle funzioni SimSpace Weaver API ha un tipo di ritorno `Aws::WeaverRuntime::Result<T>`. Se la funzione è stata eseguita correttamente, `Result` contiene `T`. Altrimenti, `Result` contiene un file `Aws::WeaverRuntime::ErrorCode` che rappresenta un codice di errore proveniente da Rust App SDK.

Example Esempio

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

Questo metodo:

- Restituisce `Transaction` se `BeginUpdate()` viene eseguito correttamente.
- Restituisce `Aws::WeaverRuntime::ErrorCode` se `BeginUpdate()` fallisce.

Interazione con l'SDK dell'app al livello più alto

Ciclo di vita

- L'SDK SimSpace Weaver dell'app gestisce il ciclo di vita dell'app. Non è necessario leggere o scrivere lo stato del ciclo di vita di un'app.

Partizioni

- Usalo `Result <PartitionSet> AssignedPartitions(Transaction& txn);` per ottenere partizioni di proprietà.
- Usa `Result <PartitionSet> AllPartitions(Transaction& txn);` per ottenere tutte le partizioni della simulazione.

Gestione della simulazione

Questa sezione descrive le soluzioni per le attività più comuni di gestione della simulazione.

Argomenti

- [Avvia una simulazione](#)
- [Aggiorna una simulazione](#)
- [Termina una simulazione](#)

Avvia una simulazione

Utilizzalo `CreateApplication()` per creare un'app.

Example Esempio

```
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()));
```


Aggiorna una simulazione

Utilizza le seguenti BeginUpdate funzioni per aggiornare l'app:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`— ti dice se `BeginUpdate()` bloccherà o non bloccherà.

Si usa `Result<void> Commit(Transaction& txn)` per confermare le modifiche.

Example Esempio

```
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();
}
```

Termina una simulazione

Utilizzare `Result<void> DestroyApplication(Application&& app)` per terminare l'app e la simulazione.

Altre app scoprono che la simulazione si interrompe quando ricevono `ErrorCode::ShuttingDown` chiamate verso `o.BeginUpdateWillBlock()` `BeginUpdate()` Quando un'app riceve `ErrorCode::ShuttingDown`, può chiamare `Result<void> DestroyApplication(Application&& app)` per terminarsi da sola.

Example Esempio

```
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

Chiama solo `Result<void> DestroyApplication(Application&& app)` dopo `Api::Commit()`. La distruzione di un'applicazione durante un aggiornamento può causare un comportamento indefinito.

⚠ Important

È necessario effettuare una chiamata `DestroyApplication()` prima della chiusura del programma per assicurarsi che l'applicazione riporti la chiusura con successo.

La mancata chiamata alla `DestroyApplication()` chiusura del programma farà sì che lo stato venga considerato uguale. FATAL

Sottoscrizioni

Crei un abbonamento con un'area di abbonamento e un ID di dominio. L'ID di dominio rappresenta il dominio che possiede quell'area di abbonamento. A `BoundingBox2F32` descrive l'area di sottoscrizione. Utilizza la seguente funzione per creare un abbonamento:

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

Example Esempio

```
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();
}
```

È possibile utilizzare il comando `Api::SubscriptionHandle` restituito da `CreateSubscriptionBoundingBox2F32()` per modificare l'abbonamento. Lo si passa come argomento alle seguenti funzioni:

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

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

Entità

Le Load API Store and vengono richiamate utilizzando l'evento `Api:Entity` di cambio di proprietà `Result<Api::Entity>` restituito da `CreateEntity()` o da un evento di cambio di proprietà quando un'entità entra nell'area di sottoscrizione dell'app (per ulteriori informazioni, consulta [Eventi dell'entità](#)). Ti consigliamo di tracciare i tuoi `Api::Entity` oggetti in modo da poterli utilizzare con queste API.

Argomenti

- [Crea entità](#)
- [Trasferisci un'entità in un dominio spaziale](#)
- [Scrivi e leggi i dati dei campi dell'entità](#)
- [Memorizza la posizione di un'entità](#)
- [Carica la posizione di un'entità](#)

Crea entità

Usa `CreateEntity()` per creare un'entità. Siete voi a definire il significato di `Api::TypeId` ciò che passate a questa funzione.

```
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

I valori 0-511 per `Api::BuiltinTypeId` sono riservati. L'entità `TypeID` (`k_entityTypeId` in questo esempio) deve avere un valore pari o superiore a 512.

Trasferisci un'entità in un dominio spaziale

Dopo che un'app o un'app di servizio personalizzata ha creato un'entità, l'app deve trasferire l'entità in un dominio spaziale affinché l'entità esista spazialmente nella simulazione. Le entità in un dominio spaziale possono essere lette da altre app e aggiornate da un'app spaziale. Usa l'`ModifyEntityDomain()` API per trasferire un'entità in un dominio spaziale.

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

Se `DomainId` non corrisponde a quello assegnato `Partition` dall'app chiamante, `DomainId` deve essere per un `DomainType::SpatialDomain`. Il trasferimento della proprietà alla nuova `Domain` avviene durante `Commit(Transaction&&)`.

Parametri

txn

La correnteTransaction.

entity

L'obiettivo Entity per il cambio diDomain.

domainId

DomainIdLa destinazione Domain perEntity.

Questa API restituisce Success se il dominio dell'entità è stato modificato con successo.

Scrivi e leggi i dati dei campi dell'entità

Tutti i campi di dati delle entità sono di tipo blob. È possibile scrivere fino a 1.024 byte di dati su un'entità. Ti consigliamo di mantenere i blob il più piccoli possibile perché dimensioni maggiori ridurranno le prestazioni. Quando si scrive su un blob, si passa SimSpace Weaver un puntatore ai dati e alla loro lunghezza. Quando si legge da un blob, SimSpace Weaver fornisce un puntatore e una lunghezza da leggere. Tutte le letture devono essere completate prima che l'app effettui una chiamata. Commit() I puntatori restituiti da una chiamata di lettura vengono invalidati quando l'app chiama. Commit()

Important

- La lettura da un puntatore blob memorizzato nella cache dopo a non Commit() è supportata e può causare il fallimento della simulazione.
- La scrittura su un puntatore blob restituito da una chiamata di lettura non è supportata e può causare il fallimento della simulazione.

Argomenti

- [Memorizza i dati del campo di un'entità](#)
- [Carica i dati del campo di un'entità](#)
- [Caricamento dei dati del campo delle entità rimosse](#)

Memorizza i dati del campo di un'entità

Gli esempi seguenti mostrano come è possibile archiviare (scrivere nello State Fabric) i dati del campo di un'entità di proprietà dell'app. Questi esempi utilizzano la seguente funzione:

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

Il `Api::TypeId keyTypeId` parametro rappresenta il tipo di dati dei dati trasmessi.

Il `Api::TypeId keyTypeId` parametro deve ricevere il modulo corrispondente `Api::TypeIdApi::BuiltinTypeId`. Se non esiste una conversione appropriata, puoi usare `Api::BuiltinTypeId::Dynamic`.

Per tipi di dati complessi, usa `Api::BuiltinTypeId::Dynamic`.

Note

Il valore di `FieldIndex index` deve essere maggiore di 0. Il valore 0 è riservato alla chiave dell'indice (vedi `StoreEntityIndexKey()`).

Example Esempio di utilizzo di tipi di dati primitivi

```
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 Esempio di utilizzo di struct a per contenere i dati

```

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));
}

```


Carica i dati del campo di un'entità

Gli esempi seguenti mostrano come caricare (leggere dallo state fabric) i dati del campo di un'entità. Questi esempi utilizzano la seguente funzione:

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

Il `Api::TypeId keyTypeId` parametro dovrebbe ricevere il corrispondente `Api::TypeId` da `Api::BuiltinTypeId`. Se non esiste una conversione appropriata, puoi usare `Api::BuiltinTypeId::Dynamic`.

Note

Il valore dell'`FieldIndex` indice deve essere maggiore di 0. Il valore 0 è riservato alla chiave dell'indice (vedi `StoreEntityIndexKey()`).

Example Esempio di utilizzo di tipi di dati primitivi

```
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 Esempio di utilizzo di struct a per contenere i dati

```

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);
}

```

Caricamento dei dati del campo delle entità rimosse

Non puoi caricare (leggere dallo State Fabric) i dati dei campi di entità per le entità che sono state rimosse dalle aree di proprietà e abbonamento dell'app. L'esempio seguente genera un errore perché richiama `Api::LoadIndexKey()` un'entità come risultato di `unApi::ChangeListAction::Remove`. Il secondo esempio mostra un modo corretto per archiviare e caricare i dati delle entità direttamente nell'app.

Example Esempio di codice errato

```
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 Esempio di un modo corretto per archiviare e caricare i dati delle entità nell'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;
}
}
}

/* ... */
}

```

Memorizza la posizione di un'entità

È possibile memorizzare (scrivere nel tessuto statale) la posizione di un'entità utilizzando una struttura di dati intera. Questi esempi utilizzano la seguente funzione:

```

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

```

Note

È necessario fornire `Api::BuiltinTypeId::Vector3F32` a `Api::StoreEntityIndexKey()`, come illustrato negli esempi seguenti.

Example Esempio di utilizzo di un array per rappresentare la posizione

```
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 Esempio di utilizzo di struct a per rappresentare la posizione

```
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));
}

```

Carica la posizione di un'entità

È possibile caricare (leggere dalla struttura degli stati) la posizione di un'entità utilizzando una struttura di dati intera. Questi esempi utilizzano la seguente funzione:

Note

È necessario fornire `Api::BuiltinTypeId::Vector3F32` a `Api::LoadEntityIndexKey()`, come illustrato negli esempi seguenti.

Example Esempio di utilizzo di un array per rappresentare la posizione

```

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 Esempio di utilizzo di struct a per rappresentare la posizione

```

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);
}
```

Eventi dell'entità

Puoi utilizzare le seguenti funzioni nell'SDK dell'SimSpace Weaverapp per ottenere tutti gli eventi di proprietà e abbonamento:

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

È possibile utilizzare il framework SimSpace Weaver demo se è necessaria l'elaborazione di eventi di entità basata su callback. Per ulteriori informazioni, consulta il seguente file di intestazione:

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

Puoi anche creare l'elaborazione degli eventi della tua entità.

Argomenti

- [Esegui iterazioni tra gli eventi per le entità di proprietà](#)
- [Esegui un'iterazione tra gli eventi per le entità sottoscritte](#)

- [Esegui un'iterazione degli eventi di cambio di proprietà delle entità](#)

Esegui iterazioni tra gli eventi per le entità di proprietà

`OwnershipChanges()` Utilizzalo per ottenere un elenco di eventi per le entità di proprietà (entità nell'area di proprietà dell'app). La funzione ha la seguente firma:

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

Quindi esegui un'iterazione tra le entità con un ciclo, come illustrato nell'esempio seguente.

Example Esempio

```
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;  
  }  
}
```

Event types (Tipi di evento)

- None— L'entità si trova nell'area e i dati relativi alla posizione e al campo non sono stati modificati.
- Remove— L'entità è stata rimossa dall'area.
- Add— L'entità è stata aggiunta all'area.
- Update— L'entità si trova nell'area ed è stata modificata.
- Reject— L'app non è riuscita a rimuovere l'entità dall'area.

Note

In caso di Reject evento, l'app tenterà nuovamente il trasferimento con il segno di spunta successivo.

Esegui un'iterazione tra gli eventi per le entità sottoscritte

Utilizzalo `AllSubscriptionEvents()` per ottenere un elenco di eventi per le entità sottoscritte (entità nell'area di iscrizione dell'app). La funzione ha la seguente firma:

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

Quindi esegui un'iterazione tra le entità con un ciclo, come illustrato nell'esempio seguente.

Example Esempio

```
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;
}
}
```

Event types (Tipi di evento)

- None— L'entità si trova nell'area e i dati relativi alla posizione e al campo non sono stati modificati.
- Remove— L'entità è stata rimossa dall'area.
- Add— L'entità è stata aggiunta all'area.
- Update— L'entità si trova nell'area ed è stata modificata.
- Reject— L'app non è riuscita a rimuovere l'entità dall'area.

Note

In caso di Reject evento, l'app tenterà nuovamente il trasferimento con il segno di spunta successivo.

Esegui un'iterazione degli eventi di cambio di proprietà delle entità

Per visualizzare gli eventi in cui un'entità si sposta tra un'area di proprietà e un'area di sottoscrizione, confronta le modifiche tra gli eventi di proprietà e sottoscrizione dell'entità correnti e precedenti.

Puoi gestire questi eventi leggendo:

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

È quindi possibile confrontare le modifiche con i dati precedentemente memorizzati.

L'esempio seguente mostra come gestire gli eventi di modifica della proprietà delle entità. Questo esempio presuppone che per le entità che passano dall'essere entità sottoscritte a entità possedute (in entrambe le direzioni), si verifichi prima l'evento di rimozione/aggiunta della proprietà, seguito dall'evento di rimozione/aggiunta della sottoscrizione nel segno di spunta successivo.

Example Esempio

```
Result<void> ProcessOwnershipEvents(Transaction& transaction)
{
    using EntityIdsByAction =
        std::unordered_map<Api::ChangeListAction,
            std::vector<Api::EntityId>>;
    using EntityIdSetByAction =
        std::unordered_map<Api::ChangeListAction,
            std::unordered_set<Api::EntityId>>;

    static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

    EntityIdSetByAction entityIdSetByAction;

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

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

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

                {
                    entityIdSetByAction[action].insert(
                        event.entity.descriptor->id);
                    break;
                }
        }
    }
}
```

```

    case Api::ChangeListAction::None:
    case Api::ChangeListAction::Update:
    case Api::ChangeListAction::Reject:
        {
            break;
        }
    }
}

EntityIdsByAction entityIdsByAction;

/**
 * Enumerate Api::OwnershipChangeList items
 * and store Add and Remove events.
 */

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

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

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

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

```

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

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

m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;
```

```

    return Success();
}

```

Resulte gestione degli errori

La `Aws::WeaverRuntime::Result<T>` classe utilizza una `Outcome` libreria di terze parti. È possibile utilizzare lo schema seguente per verificare gli errori `Result` e `catch` restituiti dalle chiamate API.

```

void DoBeginUpdate(Application& app)
{
    Result<Transaction> transactionResult = Api::BeginUpdate(app);

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

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

```

Resultistruzione di controllo, macro

All'interno di una funzione con un tipo restituito `Aws::WeaverRuntime::Result<T>`, è possibile utilizzare la `WEAVERRUNTIME_TRY` macro anziché lo schema di codice precedente. La macro eseguirà la funzione che le è stata passata. Se la funzione passata fallisce, la macro farà in modo che la funzione di inclusione restituisca un errore. Se la funzione passata ha esito positivo, l'esecuzione passa alla riga successiva. L'esempio seguente mostra una riscrittura della funzione precedente. `DoBeginUpdate()` Questa versione utilizza la `WEAVERRUNTIME_TRY`

macro anziché la struttura di if-else controllo. Nota che il tipo restituito dalla funzione è `Aws::WeaverRuntime::Result<void>`.

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

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

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

Se `BeginUpdate()` fallisce, la macro `DoBeginUpdate()` restituisce anticipatamente con un errore. È possibile utilizzare la `WEAVERRUNTIME_EXPECT_ERROR` macro per ottenere il file `Aws::WeaverRuntime::ErrorCode` da `BeginUpdate()`. L'esempio seguente mostra come la `Update()` funzione chiama `DoBeginUpdate()` e ottiene il codice di errore in caso di errore.

```
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);
    }
}
```



```

    }
}

```

È possibile rendere `BeginUpdate()` disponibile il codice di errore per una funzione che chiama `Update()` modificando il tipo restituito da `Update()` to `Aws::WeaverRuntime::Result<void>`. È possibile ripetere questo processo per continuare a inviare il codice di errore più in basso nello stack di chiamate.

Generici e tipi di dominio

L'SDK SimSpace Weaver dell'app fornisce i tipi di dati a precisione singola `Api::Vector2F32` e `Api::BoundingBox2F32`, a doppia precisione e `Api::Vector2F64` `Api::BoundingBox2F64`. Questi tipi di dati sono strutture di dati passive prive di metodi pratici. Tieni presente che l'API utilizza solo `Api::Vector2F32` e `Api::BoundingBox2F32`. Puoi utilizzare questi tipi di dati per creare e modificare abbonamenti.

Il framework SimSpace Weaver demo fornisce una versione minima della libreria AzCore matematica, che contiene `Vector3` e `Aabb`. Per ulteriori informazioni, consultate i file di intestazione in:

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

Operazioni varie dell'SDK dell'app

Argomenti

- [AllSubscriptionEventse OwnershipChanges contengono gli eventi dell'ultima chiamata](#)
- [Rilascia i blocchi di lettura dopo l'elaborazione SubscriptionChangeList](#)
- [Crea un'istanza di app autonoma per il test](#)

AllSubscriptionEventse OwnershipChanges contengono gli eventi dell'ultima chiamata

I valori restituiti dalle chiamate `Api::AllSubscriptionEvents()` e `Api::OwnershipChanges()` contengono gli eventi dell'ultima chiamata, non l'ultimo segno di spunta. Nell'esempio seguente, `secondSubscriptionEvents` e `secondOwnershipChangeList` sono vuoti perché le relative funzioni vengono richiamate immediatamente dopo le prime chiamate.

Se attendi 10 tick e poi chiami `Api::AllSubscriptionEvents()` and `Api::OwnershipChanges()`, i risultati conterranno sia gli eventi che le modifiche rispetto agli ultimi 10 tick (non all'ultimo segno di spunta).

Example Esempio

```
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

La funzione **AllSubscriptionEvents()** è implementata ma la funzione non **SubscriptionEvents()** è implementata.

Rilascia i blocchi di lettura dopo l'elaborazione SubscriptionChangeList

Quando si avvia un aggiornamento, sono presenti segmenti di memoria condivisa per i dati salvati in altre partizioni relative al segno di spunta precedente. Questi segmenti di memoria condivisa potrebbero essere bloccati dai lettori. Un'app non può eseguire il commit completo finché tutti i lettori non hanno rilasciato i lucchetti. Come ottimizzazione, un'app dovrebbe chiamare `Api::ReleaseReadLeases()` per rilasciare i blocchi dopo aver

`Api::SubscriptionChangelist` elaborato gli elementi. Ciò riduce le contese al momento del commit. `Api::Commit()` rilascia i lease di lettura per impostazione predefinita, ma è consigliabile rilasciarli manualmente dopo l'elaborazione degli aggiornamenti dell'abbonamento.

Example Esempio

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

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

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Crea un'istanza di app autonoma per il test

Puoi utilizzarla `Api::CreateStandaloneApplication()` per creare un'app autonoma per testare la logica dell'app prima di eseguire il codice in una simulazione effettiva.

Example Esempio

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

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

    ...
}
```

AWS SimSpace Weaver struttura dimostrativa

Il framework AWS SimSpace Weaver demo (framework demo) è una libreria di utilità che puoi utilizzare per sviluppare SimSpace Weaver app.

Il framework demo fornisce

- Esempi di codice e schemi di programmazione da utilizzare ed esaminare
- Astrazioni e funzioni di utilità che semplificano lo sviluppo di app semplici
- Un modo più semplice per testare le funzionalità sperimentali dell' SDK SimSpace Weaver app

Abbiamo progettato l' SDK SimSpace Weaver dell'app con accesso di basso livello alle SimSpace Weaver API per offrire prestazioni più elevate. Al contrario, abbiamo progettato il framework demo per fornire astrazioni di livello superiore e accesso alle API SimSpace Weaver più facili da usare. Il costo della facilità d'uso è un livello di prestazioni inferiore rispetto all'utilizzo diretto dell' SDK SimSpace Weaver dell'app. Le simulazioni in grado di tollerare prestazioni inferiori (come quelle senza requisiti di prestazioni in tempo reale) potrebbero essere buone candidate per utilizzare il framework demo. Ti consigliamo di utilizzare la funzionalità nativa nell' SDK dell' SDK SimSpace Weaver app per applicazioni complesse perché il framework demo non è un toolkit completo.

Il framework demo include

- Esempi di codice funzionante che supportano e dimostrano:
 - Gestione del flusso delle app
 - Elaborazione di eventi di entità basata sul callback
- Una serie di librerie di utilità di terze parti:
 - spdlog (una libreria di registrazione)
 - Una versione minima di AZCore (una libreria matematica) che contiene solo:
 - Vector3
 - Aabb
 - cxxopts (una libreria di parser di opzioni a riga di comando)
- Funzioni di utilità specifiche per SimSpace Weaver

Il framework demo è composto da una libreria, file sorgente e CMakeLists. I file sono inclusi nel pacchetto distribuibile SimSpace Weaver dell' SDK dell'app.

Utilizzo delle quote di servizio

In questa sezione viene descritto come lavorare con le quote di servizio per SimSpace Weaver. Le quote sono anche chiamate limiti. Per un elenco delle quote di servizio, consulta [SimSpace Endpoint e quote Weaver](#). Le API in questa sezione fanno parte del set di API delle app. Le API delle app sono diverse dalle API dei servizi. Le API delle app fanno parte dell'SDK SimSpace Weaver dell'app. Puoi trovare la documentazione per le API delle app nella cartella SDK dell'app sul tuo sistema locale:

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

Argomenti

- [Scopri i limiti di un'app](#)
- [Ottieni la quantità di risorse utilizzate da un'app](#)
- [Reimpostazione delle metriche](#)
- [Superamento di un limite](#)
- [In esaurimento della memoria](#)
- [Best practice](#)

Scopri i limiti di un'app

Puoi utilizzare l'API `RuntimeLimits` dell'app per interrogare i limiti di un'app.

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

Parametri

Application app

Un riferimento all'app.

LimitType tipo

Un enum con i seguenti tipi di limite:

```
enum LimitType {  
    Unset = 0,
```

```

EntitiesPerPartition = 1,
RemoteEntityTransfers = 2,
LocalEntityTransfers = 3
};

```

L'esempio seguente interroga il limite del numero di entità.

```

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

```

Ottieni la quantità di risorse utilizzate da un'app

Puoi chiamare l'API `RuntimeMetrics` dell'app per ottenere la quantità di risorse utilizzate da un'app:

```

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

```

Parametri

Applicatione app

Un riferimento all'app.

L'API restituisce un riferimento a unstruct che contiene le metriche. Una metrica del contatore contiene un valore totale corrente e aumenta solo. Una metrica dell'indicatore contiene un valore che può aumentare o diminuire. Il runtime dell'applicazione aggiorna un contatore ogni volta che un evento aumenta il valore. Il runtime aggiorna gli indicatori solo quando si chiama l'API. SimSpace Weaver garantisce che il riferimento sia valido per tutta la durata dell'app. Le chiamate ripetute all'API non modificheranno il riferimento.

```

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
}

```

Reimpostazione delle metriche

L'API `ResetRuntimeMetrics` dell'app reimposta i valori in `AppRuntimeMetricsStruct`.

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

L'esempio seguente mostra come è possibile effettuare chiamate `ResetRuntimeMetrics` nella propria app.

```

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

    ticks_since_last_report = 0;

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

```

Superamento di un limite

Una chiamata all'API dell'app che supera un limite restituirà un `ErrorCode::CapacityExceeded`, ad eccezione dei trasferimenti di entità. SimSpace Weaver gestisce i trasferimenti di entità in modo asincrono come parte delle operazioni di `Commit` e dell'API dell'`BeginUpdateapp`, quindi non esiste un'operazione specifica che restituisca un errore se un trasferimento fallisce a causa del limite di trasferimento dell'entità. Per rilevare gli errori di trasferimento, è possibile confrontare i valori correnti di `rejected_incoming_transfer_counter` e `rejected_outgoing_transfer_counter` (nel `AppRuntimeMetricsStruct`) con i valori precedenti. Le entità rifiutate non saranno presenti nella partizione, ma l'app può comunque simularle.

In esaurimento della memoria

SimSpace Weaver utilizza un processo di raccolta dei rifiuti per ripulire e rilasciare la memoria liberata. È possibile scrivere dati più velocemente di quanto il garbage collector possa rilasciare memoria. In questo caso, le operazioni di scrittura potrebbero superare il limite di memoria riservata dell'app. SimSpace Weaver restituirà un errore interno con un messaggio che contiene `OutOfMemory` (e dettagli aggiuntivi). Per ulteriori informazioni, consulta [Diffondi le scritture nel tempo](#).

Best practice

Le seguenti best practice sono linee guida generali per progettare le app in modo da evitare di superare i limiti. Potrebbero non essere applicabili al design specifico della tua app.

Monitora frequentemente e rallenta

Dovresti monitorare frequentemente le tue metriche e rallentare le operazioni che stanno per raggiungere un limite.

Evita di superare i limiti di abbonamento e i limiti di trasferimento

Se possibile, progetta la simulazione per ridurre il numero di abbonamenti remoti e trasferimenti di entità. È possibile utilizzare i gruppi di posizionamento per posizionare più partizioni sullo stesso lavoratore e ridurre la necessità di trasferimenti di entità remote tra lavoratori.

Diffondi le scritture nel tempo

Il numero e la dimensione degli aggiornamenti in un segno di spunta possono avere un impatto significativo sul tempo e sulla memoria necessari per eseguire una transazione. I requisiti di memoria elevati possono causare l'esaurimento della memoria del runtime dell'applicazione. Puoi distribuire le scritture nel tempo per ridurre la dimensione totale media degli aggiornamenti per segno di spunta. Questo può aiutare a migliorare le prestazioni ed evitare il superamento dei limiti. Ti consigliamo di non scrivere più di una media di 12 MB su ogni segno di spunta o 1,5 KB per ciascuna entità.

Simulazioni di debug

È possibile utilizzare i seguenti metodi per ottenere informazioni sulle simulazioni.

Argomenti

- [Usa SimSpace Weaver Local e guarda l'output della console](#)

- [Guarda i tuoi log in Amazon CloudWatch Logs](#)
- [Usa le chiamate describe API](#)
- [Connect a client](#)

Usa SimSpace Weaver Local e guarda l'output della console

Ti consigliamo di sviluppare prima le simulazioni localmente e poi di eseguirle Cloud AWS in. Puoi visualizzare l'output della console direttamente quando esegui con SimSpace Weaver Local. Per ulteriori informazioni, consulta [Sviluppo locale](#).

Guarda i tuoi log in Amazon CloudWatch Logs

Quando esegui la simulazione nella console, Cloud AWS l'output delle tue app viene inviato ai flussi di log in Amazon CloudWatch Logs. La tua simulazione scrive anche altri dati di log. È necessario abilitare la registrazione nello schema di simulazione se si desidera che la simulazione scriva dati di registro. Per ulteriori informazioni, consulta [SimSpace Weaver accessi in Amazon CloudWatch Logs](#).

Warning

La simulazione può produrre grandi quantità di dati di registro. I dati di registro possono crescere molto rapidamente. È necessario monitorare attentamente i log e interrompere le simulazioni quando non è più necessario eseguirle. La registrazione può comportare costi elevati.

Usa le chiamate describe API

Puoi utilizzare le seguenti API di servizio per ottenere informazioni sulle tue simulazioni in. Cloud AWS

Important

Se utilizzi AWS IAM Identity Center o nomi profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva SimSpace Weaver dell'SDK dell'app. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK SimSpace Weaver dell'app utilizzano il. AWS CLI Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo default profilo o fornire il nome del tuo profilo IAM Identity Center per applicare gli script

SimSpace Weaver SDK con il parametro. AWS CLI `--profile cli-profile-name` Per ulteriori informazioni, consulta [Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente. AWS Command Line Interface

- ListSimulations— ottieni un elenco di tutte le tue simulazioni in. Cloud AWS

Example Esempio

```
tools\windows\weaver-MyProject-cli.bat list-simulations
```

- DescribeSimulation— ottieni dettagli su una simulazione.

Example Esempio

```
tools\windows\weaver-MyProject-cli.bat describe-simulation --simulation MySimulation
```

- DescribeApp— ottieni dettagli su un'app.

Example Esempio

```
tools\windows\weaver-MyProject-cli.bat describe-app --simulation MySimulation --  
domain MyCustomDomain --app MyCustomApp
```

Per ulteriori informazioni sulle SimSpace Weaver API, consulta [SimSpace Weaver Riferimenti API](#).

Connect a client

Puoi connettere un client a un'app personalizzata o di servizio endpoint_config in esecuzione definita con uno schema di simulazione. L'SDK SimSpace Weaver dell'app include client di esempio che puoi utilizzare per visualizzare l'applicazione di esempio. Puoi esaminare il codice sorgente di questi client di esempio e l'applicazione di esempio per vedere come creare i tuoi client. Per ulteriori informazioni su come creare ed eseguire i client di esempio, consulta [Fase 5: Visualizza la simulazione](#).

Puoi trovare il codice sorgente per i client di esempio nella seguente cartella:

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

Eseguire il debug di simulazioni locali

Puoi eseguire il debug delle tue SimSpace Weaver Local app con Microsoft Visual Studio [Per ulteriori informazioni su come eseguire il debug con Visual Studio, consulta. Microsoft Visual Studio documentation](#)

Per eseguire il debug della simulazione locale

1. Assicurati che `schema.yaml` sia nella tua directory di lavoro.
2. In Visual Studio, apri il menu contestuale per ogni app di cui desideri eseguire il debug (ad esempio `PathfindingSampleLocalSpatial` o `PathfindingSampleLocalView`) e imposta la directory di lavoro nella sezione di debug.
3. Apri il menu contestuale dell'app di cui desideri eseguire il debug e seleziona `Imposta come progetto di avvio`.
4. Scegli `F5` di avviare il debug dell'app.

I requisiti per eseguire il debug di una simulazione sono gli stessi necessari per eseguire una simulazione normalmente. È necessario avviare il numero di app spaziali specificato nello schema. Ad esempio, se lo schema specifica una griglia 2x2 e si avvia un'app spaziale in modalità debug, la simulazione non verrà eseguita finché non si avviano altre 3 app spaziali (in modalità debug o non in modalità debug).

Per eseguire il debug di un'app personalizzata, devi prima avviare le app spaziali e quindi avviare l'app personalizzata nel debugger.

Tieni presente che la simulazione viene eseguita in blocco. Non appena un'app raggiunge un punto di interruzione, tutte le altre app verranno messe in pausa. Dopo aver proseguito da quel punto di interruzione, le altre app continueranno.

Contenitori personalizzati

AWS SimSpace Weaver le app vengono eseguite in ambienti containerizzati Amazon Linux 2 (AL2). Nel Cloud AWS, SimSpace Weaver esegue le simulazioni in contenitori Docker creati a partire da un'immagine `amazonlinux:2` fornita da Amazon Elastic Container Registry (Amazon ECR). Puoi creare un'immagine Docker personalizzata, archivarla in Amazon ECR e utilizzare quell'immagine per la tua simulazione anziché l'immagine Docker predefinita che forniamo.

Puoi utilizzare un contenitore personalizzato per gestire le dipendenze software e includere componenti software aggiuntivi che non sono presenti nell'immagine Docker standard. Ad esempio, puoi aggiungere al contenitore le librerie software disponibili pubblicamente utilizzate dall'app e inserire il codice personalizzato solo nel file zip dell'app.

Important

Supportiamo solo immagini Docker AL2 ospitate nei repository Amazon ECR, in Amazon ECR Public Gallery o nel tuo registro Amazon ECR privato. Non supportiamo immagini Docker ospitate al di fuori di Amazon ECR. Per ulteriori informazioni su Amazon ECR, consulta la [documentazione di Amazon Elastic Container Registry](#).

Argomenti

- [Crea un contenitore personalizzato](#)
- [Modifica un progetto per utilizzare un contenitore personalizzato](#)
- [Domande frequenti sui contenitori personalizzati](#)
- [Risoluzione dei problemi relativi ai contenitori personalizzati](#)

Crea un contenitore personalizzato

Queste istruzioni presuppongono che tu sappia usare Docker e Amazon Elastic Container Registry (Amazon ECR). Per ulteriori informazioni su Amazon ECR, consulta la [Guida per l'utente di Amazon ECR](#).

Prerequisiti

- L'identità IAM (uso o ruolo) che usi per eseguire queste azioni dispone delle autorizzazioni corrette per utilizzare Amazon ECR
- Docker è installato sul tuo sistema locale

Per creare un contenitore personalizzato

1. Crea il tuo `Dockerfile`.

Un'AWS SimSpace Weaver app `Dockerfile` per eseguire inizia con l'Amazon Linux 2 immagine in Amazon ECR.

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

2. Crea il tuo `Dockerfile`
3. Carica l'immagine del contenitore su Amazon ECR.
 - [Usa il. AWS Management Console](#)
 - [Usa ilAWS Command Line Interface.](#)

Note

Se ricevi un `AccessDeniedException` errore quando tenti di caricare l'immagine del contenitore su Amazon ECR, la tua identità IAM (utente o ruolo) potrebbe non disporre delle autorizzazioni necessarie per utilizzare Amazon ECR. Puoi collegare la policy `AmazonEC2ContainerRegistryPowerUser` AWS gestita alla tua identità IAM e riprovare. Per ulteriori informazioni su come allegare una policy, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM](#) nella Guida per l'AWS Identity and Access Managementutente.

Modifica un progetto per utilizzare un contenitore personalizzato

Queste istruzioni presuppongono che tu sappia già come usarle AWS SimSpace Weaver e che desideri rendere Cloud AWS più efficienti i flussi di lavoro di archiviazione e sviluppo delle app.

Prerequisiti

- Stai modificando un SimSpace Weaver progetto esistente creato dallo `create-project.bat` script.
- Hai un contenitore personalizzato in Amazon Elastic Container Registry (Amazon ECR). Per ulteriori informazioni sulla creazione di un contenitore personalizzato, consulta [Crea un contenitore personalizzato](#).

Per modificare il progetto in modo da utilizzare un contenitore personalizzato

1. Aggiungi le autorizzazioni al ruolo dell'app di simulazione del tuo progetto per utilizzare Amazon ECR.

- a. Se non disponi già di una policy IAM con le seguenti autorizzazioni, crea la policy. Ti suggeriamo il nome `simspaceweaver-ecr` della policy. Per ulteriori informazioni su come creare una policy IAM, consulta [Creazione di policy IAM](#) nella Guida per l'AWS Identity and Access Management.

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

- b. Trova il nome del ruolo dell'app di simulazione del tuo progetto:
 - i. In un editor di testo, apri il AWS CloudFormation modello per il tuo progetto:

```
project-folder\cloudformation\weaver-project-name-stack.yaml
```

- ii. Trova la `RoleName` proprietà sotto `WeaverAppRole`. Il valore è il nome del ruolo dell'app di simulazione del progetto.

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. Allega la `simspaceweaver-ecr` policy al ruolo dell'app di simulazione del progetto. Per ulteriori informazioni su come allegare una policy, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida](#) per l'AWS Identity and Access Management.
2. Specificate le immagini del contenitore nello schema di simulazione del progetto.
 - Puoi aggiungere la `default_image` proprietà opzionale in `simulation_properties` per specificare un'immagine contenitore personalizzata predefinita per tutti i domini.
 - Aggiungi la `image` proprietà nel campo `app_config` per il dominio in cui desideri utilizzare un'immagine contenitore personalizzata. Specificare l'URI del repository Amazon ECR come valore. Puoi specificare un'immagine diversa per ogni dominio.
 - Se `image` non è specificato per un dominio e `default_image` viene specificato, le app di quel dominio utilizzano l'immagine predefinita.
 - Se `image` non è specificato per un dominio e `default_image` non è specificato, le app di quel dominio vengono eseguite in un SimSpace Weaver contenitore standard.

Example Frammento di schema che include impostazioni personalizzate del contenitore

```

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. Crea e carica il tuo progetto come al solito.

Domande frequenti sui contenitori personalizzati

D1. Cosa devo fare se voglio modificare il contenuto del mio contenitore?

- Per una simulazione in esecuzione: non è possibile cambiare il contenitore per una simulazione di esecuzione. È necessario creare un nuovo contenitore e avviare una nuova simulazione che utilizzi quel contenitore.
- Per una nuova simulazione: crea un nuovo contenitore, caricalo su Amazon Elastic Container Registry (Amazon ECR) e avvia una nuova simulazione che utilizzi quel contenitore.

D2. Come posso modificare l'immagine del contenitore per la mia simulazione?

- Per una simulazione in esecuzione: non è possibile modificare il contenitore per una simulazione in esecuzione. È necessario avviare una nuova simulazione che utilizzi il nuovo contenitore.
- Per una nuova simulazione: specifica la nuova immagine del contenitore nello schema di simulazione del progetto. Per ulteriori informazioni, consulta [Modifica un progetto per utilizzare un contenitore personalizzato](#).

Risoluzione dei problemi relativi ai contenitori personalizzati

Argomenti

- [AccessDeniedException quando carichi l'immagine su Amazon Elastic Container Registry \(Amazon ECR\)](#)
- [Una simulazione che utilizza un contenitore personalizzato non riesce ad avviarsi](#)

AccessDeniedException quando carichi l'immagine su Amazon Elastic Container Registry (Amazon ECR)

Se ricevi un `AccessDeniedException` errore quando tenti di caricare l'immagine del contenitore su Amazon ECR, la tua identità IAM (utente o ruolo) potrebbe non disporre delle autorizzazioni necessarie per utilizzare Amazon ECR. Puoi collegare la policy `AmazonEC2ContainerRegistryPowerUser` AWS gestita alla tua identità IAM e riprovare. Per ulteriori informazioni su come allegare una policy, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM](#) nella Guida per l'AWS Identity and Access Management utente.

Una simulazione che utilizza un contenitore personalizzato non riesce ad avviarsi

Suggerimenti per la risoluzione dei problemi

- Se la registrazione è abilitata per la simulazione, controlla i log degli errori. [Per ulteriori informazioni, consulta il tutorial dettagliato.](#)
- Testa la tua simulazione senza un contenitore personalizzato.
- Testa la tua simulazione localmente. Per ulteriori informazioni, consulta [Sviluppo locale.](#)

Uso di Python

Puoi usare Python per `SimSpace Weaver` app e client. Il kit di sviluppo software Python (Python SDK) è incluso come parte dello standard `SimSpace Weaver` pacchetto distribuibile per app SDK. Lo sviluppo con Python funziona in modo simile allo sviluppo negli altri linguaggi supportati.

Important

`SimSpace Weaver` supporta solo la versione 3.9 di Python.

⚠ Important

SimSpace Weaver il supporto per Python richiede SimSpace Weaver versione 1.15.0 o successiva.

Argomenti

- [Creare un progetto Python](#)
- [Avvio di una simulazione Python](#)
- [Il client Python di esempio](#)
- [Scrivere i propri script di build](#)
- [Domande frequenti sull'uso di Python](#)
- [Risoluzione dei problemi relativi a Python](#)

Creare un progetto Python

Usi il `create-project.bat` script per creare un progetto Python, allo stesso modo in cui si crea un progetto non Python. Puoi usare il `PythonBubblesSample` modello come punto di partenza per il tuo progetto Python. Vedi [Crea un progetto Python](#), di seguito.

Contenitore personalizzato Python

Per eseguire il tuo sistema basato su Python SimSpace Weaver simulazione in Cloud AWS, è possibile creare un contenitore personalizzato che includa le dipendenze necessarie. Per ulteriori informazioni, consulta [Contenitori personalizzati](#).

Un contenitore personalizzato Python deve includere quanto segue:

- gcc
- openssl-devel
- bzip2-devel
- libffi-devel
- wget
- tar
- gzip

- `make`
- Python (versione 3.9)

Se si utilizza il `PythonBubblesSample` modello per creare il tuo progetto, puoi eseguire il `create-custom-container.bat` script (che si trova nella `tools` cartella del progetto) per creare un'immagine Docker con le dipendenze necessarie. Lo script carica l'immagine su Amazon Elastic Container Registry (Amazon ECR).

Lo script `create-custom-container.bat` utilizza quanto segue `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
```

È possibile aggiungere le proprie dipendenze al `Dockerfile`:

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

Il file `requirements.txt` contiene un elenco di pacchetti Python richiesti per la simulazione di esempio `PythonBubblesSample`:

```
Flask==2.1.1
```

È possibile aggiungere le proprie dipendenze dei pacchetti Python al `requirements.txt`:

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

Il `Dockerfile` e il `requirements.txt` sono nella `tools` cartella del tuo progetto.

⚠ Important

Devi scappare `create-custom-container.bat` dopo eventuali modifiche al `Dockerfile` o `requirements.txt`.

⚠ Important

Tecnicamente non è necessario utilizzare un contenitore personalizzato con la simulazione Python, ma consigliamo vivamente di utilizzare un contenitore personalizzato. Il contenitore Amazon Linux 2 (AL2) standard che forniamo non ha Python. Pertanto, se non utilizzi un contenitore personalizzato con Python (come l'immagine del contenitore creata da `create-custom-container.bat` script), devi includere Python e le dipendenze richieste in ogni file zip dell'app in cui carichi SimSpace Weaver.

Crea un progetto Python

La procedura seguente è per Microsoft Windows. Se si utilizza Windows Subsystem for Linux (WSL), utilizzare il `.sh` versione di `.bat` script invece. È necessario completare la configurazione di Amazon Elastic Container Registry (Amazon ECR) per utilizzare questa procedura. Per ulteriori informazioni, consulta [Configurazione con Amazon ECR](#) nella Guida per l'utente di Amazon ECR.

Per creare un progetto Python

1. In una finestra del prompt dei comandi, vai al tuo SimSpace Weaver cartella SDK.

```
cd sdk-folder
```

2. Esegui `create-project.bat` con il `PythonBubblesSample` modello.

```
.\create-project.bat --name project-name --path project-folder-parent-path --  
template PythonBubblesSample
```

3. Vai al `tools` cartella nella cartella del progetto. La tua *cartella-progetto* è `project-folder-parent-path\project-name`.

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

4. Crea il contenitore personalizzato.

```
.\create-custom-container.bat
```

Avvio di una simulazione Python

Puoi avviare la tua simulazione basata su Python allo stesso modo di una normale SimSpace Weaversimulazione, entrambe in SimSpace Weaver Locale in SimSpace Weaver nel Cloud AWS. Per ulteriori informazioni, consulta gli argomenti seguenti:

SimSpace Weaver Local

- [Sviluppo locale](#)

Cloud AWS

- [Passaggio 3: Esegui lo script di avvio rapido](#) nel tutorial di avvio rapido
- [Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio](#)

Il `PythonBubblesSample` include il proprio client di esempio in Python. Per ulteriori informazioni, consulta [Il client Python di esempio](#).

Il client Python di esempio

Se si utilizza il `PythonBubblesSample` modello per creare un progetto, il progetto contiene un client di esempio Python. È possibile utilizzare il client di esempio per visualizzare il `PythonBubblesSample` simulazione. Puoi anche usare il client di esempio come punto di partenza per creare il tuo client Python.

La procedura seguente presuppone che sia stato creato un `PythonBubblesSample` progetto e ne ha avviato la simulazione.

Per avviare il client Python

1. In un finestra del prompt dei comandi, vai al `src\PythonBubblesSample\bin` cartella del tuo progetto.

```
cd project-folder\src\PythonBubblesSample\bin
```

2. Esegui il client Python.

```
python bubbles_tkinter_client.py --host ip-address --port port-number --  
simsizes max-entitites
```

Parametri

host

L'indirizzo IP della simulazione. Per una simulazione iniziata nelCloud AWS, puoi trovare l'indirizzo IP della tua simulazione nel[SimSpace Weaverplancia](#)oppure usa la procedura in[Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#)del tutorial di avvio rapido. Per una simulazione locale, usa`127.0.0.1`come indirizzo IP.

port

Il numero di porta della simulazione. Per una simulazione iniziata nelCloud AWS, questa è laAttuale numero di porta. Puoi trovare il numero di porta della tua simulazione nel[SimSpace Weaverplancia](#)oppure usa la procedura in[Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#)del tutorial di avvio rapido. Per una simulazione locale, usa`7000`come numero di porta.

simsizes

Il numero massimo di entità da visualizzare nel client.

Scrivere i propri script di build

Puoi scrivere i tuoi script di compilazione per la tua simulazione in Python. Per avere una build di successo, devono essere eseguiti i seguenti passaggi:

1. Copia il contenuto di`src/PythonBubblesSample/nelBuild/out`elenco.
2. Copia il contenuto di`${WEAVER_SDK_DIRECTORY}/lib/weaver/weaver_python_app_sdk_v1nelBuild/out/lib/weaver_app_sdk_v1`elenco.
3. Copia`${WEAVER_SDK_DIRECTORY}/lib/weaver/libweaver_app_sdk_python_v1_39.so`nelBuild/out/lib/weaver_app_sdk_v1elenco.
4. Rinomina `Build/out/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_39.so` come `libweaver_app_sdk_python_v1.so`.
5. Comprimi il contenuto di`Build/out/rubrica`.

6. Ripeti il processo zip per ogni app zip specificato nello schema della simulazione. Per il `PythonBubblesSample`, lo schema prevede un `project-nameSpatial.zip` e un `project-nameView.zip`.

Dopo questi passaggi, i file zip sono pronti per essere caricati nel bucket Amazon S3 del progetto.

Domande frequenti sull'uso di Python

D1. Quali versioni di Python sono supportate?

SimSpace Weaver supporta solo la versione 3.9 di Python.

Risoluzione dei problemi relativi a Python

Argomenti

- [Errore durante la creazione di contenitori personalizzati](#)
- [La tua simulazione in Python non si avvia](#)
- [Un client di simulazione o visualizzazione in Python genera un `ModuleNotFoundError`](#)

Errore durante la creazione di contenitori personalizzati

Se viene visualizzato un errore `no basic auth credentials` dopo aver corso `create-custom-container.bat` allora potrebbe esserci un problema con le tue credenziali temporanee per Amazon ECR. Esegui il seguente comando con il tuo `Region` `AWSID` e `AWS` numero di conto:

```
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

Assicurati che la Regione AWS che specifichi è lo stesso che usi per la simulazione. Usa uno dei Regioni AWS che SimSpace Weavers supporta. Per ulteriori informazioni, consulta [SimSpace Endpoint e quote Weaver](#).

Dopo aver eseguito il `aws ecs create-custom-container` di nuovo.

Altre risorse per la risoluzione dei problemi da verificare

- [Risoluzione dei problemi relativi ai contenitori personalizzati](#)
- [Risoluzione dei problemi di Amazon ECR](#) nel Guida per l'utente di Amazon ECR
- [Configurazione con Amazon ECR](#) nel Guida per l'utente di Amazon ECR

La tua simulazione in Python non si avvia

Potresti vedere un `Unable to start` app errore nel registro di gestione della simulazione. Questo può accadere se la creazione del contenitore personalizzato non è riuscita. Per ulteriori informazioni, consulta [Errore durante la creazione di contenitori personalizzati](#). Per ulteriori informazioni sui log, consulta [SimSpace Weaver accessi in Amazon CloudWatch Logs](#).

Se sei sicuro che non ci sia nulla di sbagliato nel tuo contenitore, controlla il codice sorgente Python dell'app. Puoi usare `SimSpace Weaver Local` per testare la tua app. Per ulteriori informazioni, consulta [Sviluppo locale](#).

Un client di simulazione o visualizzazione in Python genera un `ModuleNotFoundError`

Python lancia un `ModuleNotFoundError` quando non riesce a trovare un pacchetto Python richiesto.

Se la simulazione è in Cloud AWS, assicurati che il tuo contenitore personalizzato abbia tutte le dipendenze richieste elencate nel tuo `requirements.txt`. Ricordati di correre `create-custom-container` di nuovo se modifichi `requirements.txt`.

Se viene visualizzato l'errore relativo al `PythonBubblesSample` cliente, usa `pip` per installare il pacchetto indicato:

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


Support per altri motori

Puoi usare il tuo C++ motore personalizzato con SimSpace Weaver. Attualmente stiamo sviluppando il supporto per i seguenti motori. Esiste una documentazione separata per ciascuno di questi motori.

Important

Le integrazioni con i motori elencati qui sono sperimentali. Sono disponibili in anteprima.

Motori

- [Unità](#) (versione minima 2021.3.7f1)
- [Unreal Engine](#) (versione minima 5.0)

Unità

È necessario che l'ambiente di Unity sviluppo sia già installato prima di creare SimSpace Weaver simulazioni con Unity. Scaricate il modulo separato Unity SDK per AWS SimSpace Weaver (Unity SDK) e seguite le istruzioni contenute nel pacchetto.

Important

È necessario utilizzare la versione più recente dell'SDK SimSpace Weaver dell'app. L'ultima versione è la 1.16.0. Per ulteriori informazioni, consulta [Versioni AWS SimSpace Weaver](#).

Per scaricare e utilizzare Unity SDK

1. Al prompt dei comandi di Windows, vai alla cartella *sdk*.
2. Esegui lo script di download. Sostituisci la *regione* con quella Regione AWS in cui inizierai la simulazione (ad esempio, *us-west-2*).

```
.\download-unity-package.bat --region region
```

Lo script viene scaricato e decompresso `SimSpaceWeaverUnityPackage.zip` nella cartella corrente.

3. Leggi la sezione `SimSpaceWeaverUnityPackage\Release\Documentation\Unity_SDK_for_AWS_SimSpace_Weaver.pdf`.

Important

Se ricevi un errore in Unity relativo a uno spazio dei nomi mancante per `JsonProperty` or `JsonAttribute`, segui questi passaggi per aggiungere il pacchetto: `Newtonsoft.Json`

1. Nell'editor Unity, scegli Finestra > Package Manager dalla barra dei menu.
2. Nella finestra Package Manager, scegli il pulsante + (più) nella parte superiore della finestra.
3. Seleziona Aggiungi pacchetto da git URL.
4. Inserisci i seguenti dati:

```
com.unity.nuget.newtonsoft-json
```

5. Scegli Aggiungi.

Important

Unity SDKNon supporta i profili denominati per AWS Command Line Interface (AWS CLI). Se utilizzi AWS CLI i profili AWS IAM Identity Center or, devi copiare o rinominare il tuo profilo denominato nel default profilo prima di utilizzare ilUnity SDK. Per ulteriori informazioni, vedere [Configurazione di AWS CLI da utilizzare AWS IAM Identity Center nella Guida per l'AWS Command Line Interface](#)utente e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'AWS Command Line Interfaceutente.

Unreal Engine

È necessario creare un server Unreal Engine dedicato a partire dal codice sorgente.

`SimSpaceWeaverAppSdkDistributable` Include una versione del `PathfindingSample` moduloUnreal Engine. Per ulteriori informazioni, consulta le istruzioni separate:

```
sdk-folder\AWS_SimSpace_Weaver_Unreal_Guide.pdf
```

Utilizzo di software concesso in licenza con AWS SimSpace Weaver

AWS SimSpace Weaver ti consente di creare simulazioni con il motore e i contenuti di simulazione che preferisci. In relazione all'utilizzo di SimSpace Weaver, l'utente è responsabile dell'ottenimento, della manutenzione e del rispetto dei termini di licenza di qualsiasi software o contenuto utilizzato nelle simulazioni. Verifica che il contratto di licenza ti consenta di distribuire il software e i contenuti in un ambiente ospitato virtuale.

Gestisci le tue risorse con AWS CloudFormation

Puoi usarlo AWS CloudFormation per gestire le tue AWS SimSpace Weaver risorse. AWS CloudFormation è un AWS servizio separato che consente di specificare, fornire e gestire AWS l'infrastruttura come codice. [Con AWS CloudFormation te crei un file JSON o YAML, chiamato modello](#). Il tuo modello specifica i dettagli della tua infrastruttura. AWS CloudFormation utilizza il modello per fornire l'infrastruttura come una singola unità, chiamata [stack](#). Quando si elimina lo stack, è possibile AWS CloudFormation eliminare contemporaneamente tutti gli elementi in esso contenuti. Puoi gestire il tuo modello utilizzando processi standard di gestione del codice sorgente (ad esempio, tracciandolo in un sistema di controllo delle versioni come [Git](#)). Per ulteriori informazioni su AWS CloudFormation, consulta la [Guida per l'utente di AWS CloudFormation](#).

La tua risorsa di simulazione

In AWS, una risorsa è un'entità utilizzabile. Esempi sono un'istanza Amazon EC2, un bucket Amazon S3 o un ruolo IAM. La tua SimSpace Weaver simulazione è una risorsa. Nelle configurazioni, di solito si specifica una AWS risorsa nel modulo `AWS::service::resource`. Per SimSpace Weaver, specifichi la tua risorsa di simulazione come `AWS::SimSpaceWeaver::Simulation`. Per ulteriori informazioni sulla risorsa di simulazione in AWS CloudFormation, consulta la [SimSpace Weaver](#) sezione della Guida per l'AWS CloudFormation utente.

Come posso utilizzare AWS CloudFormation con SimSpace Weaver?

È possibile creare un AWS CloudFormation modello che specifichi le AWS risorse di cui intendi effettuare il provisioning. Il tuo modello può specificare un'intera architettura, parte di un'architettura o una piccola soluzione. Ad esempio, puoi specificare un'architettura per la tua SimSpace Weaver soluzione che includa bucket Amazon S3, autorizzazioni IAM, un database di supporto in Amazon Relational Database Service o Amazon DynamoDB e la tua risorsa. `Simulation` È quindi possibile AWS CloudFormation utilizzare per fornire tutte queste risorse come unità e contemporaneamente.

Example modello che crea risorse IAM e avvia una simulazione

Il seguente modello di esempio crea un ruolo IAM e le autorizzazioni SimSpace Weaver necessarie per eseguire azioni nel tuo account. Gli script SDK dell'SimSpace Weaverapp creano il ruolo e le autorizzazioni in uno specifico Regione AWS momento della creazione di un progetto, ma puoi utilizzare un AWS CloudFormation modello per distribuire la simulazione su un altro Regione AWS senza eseguire nuovamente gli script. Ad esempio, puoi farlo per configurare una simulazione di backup per scopi di disaster recovery.

In questo esempio, il nome originale della simulazione è `MySimulation`. Un bucket per lo schema esiste già in Regione AWS where AWS CloudFormation will build the stack. Il bucket contiene una versione dello schema configurata correttamente per eseguire la simulazione al suo interno. Regione AWS Ricorda che lo schema specifica la posizione dei file zip dell'app, che è un bucket Amazon S3 nello Regione AWS stesso della simulazione. Il bucket e i file dell'app zips devono già esistere nello stack Regione AWS quando AWS CloudFormation crea lo stack, altrimenti la simulazione non verrà avviata. Nota che il nome del bucket in questo esempio include Regione AWS, ma ciò non determina dove si trova effettivamente il bucket. Devi assicurarti che il bucket sia effettivamente all'interno Regione AWS (puoi controllare le proprietà del bucket nella console Amazon S3, con le API di Amazon S3 o con i comandi Amazon S3 nel). AWS CLI

Questo esempio utilizza alcune funzioni e parametri incorporati AWS CloudFormation per eseguire la sostituzione delle variabili. Per ulteriori informazioni, vedere [Riferimento alla funzione intrinseca e Riferimento agli pseudoparametri nella Guida](#) per l'utente. 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'

Utilizzo delle istantanee con AWS CloudFormation

Un'[istantanea](#) è un backup di una simulazione. L'esempio seguente avvia una nuova simulazione da un'istantanea anziché da uno schema. L'istantanea in questo esempio è stata creata da una simulazione di progetto SDK SimSpace Weaver dell'app. AWS CloudFormation crea la nuova risorsa di simulazione e la inizializza con i dati dell'istantanea. La nuova simulazione può avere una simulazione MaximumDuration diversa da quella originale.

Ti consigliamo di creare e utilizzare una copia del ruolo dell'app della simulazione originale. Il ruolo dell'app della simulazione originale potrebbe essere eliminato se elimini lo stack di AWS CloudFormation quella simulazione.

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

      SnapshotS3Location:
        BucketName: "weaver-mytestsimulation-111122223333-artifacts-us-west-2"
        ObjectKey: "snapshot/MyTestSimulation_22-12-15_12_00_00-230428-1207-13.zip"
```

Snapshot

Puoi creare un'istantanea per eseguire il backup dei dati dell'entità di simulazione in qualsiasi momento. SimSpace Weaver crea un file.zip in un bucket Amazon S3. Puoi creare una nuova simulazione con uno snapshot. SimSpace Weaver inizializza lo State Fabric della nuova simulazione con i dati delle entità memorizzati nell'istantanea, avvia le app spaziali e di servizio in esecuzione al momento della creazione dell'istantanea e imposta l'orologio sul segno di spunta appropriato. SimSpace Weaver ottiene la configurazione della simulazione dall'istantanea anziché da un file di schema. I file.zip della tua app devono trovarsi nella stessa posizione in Amazon S3 in cui si trovavano nella simulazione originale. È necessario avviare le app personalizzate separatamente.

Argomenti

- [Casi d'uso per uno snapshot](#)
- [Usa l'SimSpace Weaver app SDK per lavorare con le istantanee](#)
- [Usa la SimSpace Weaver console per lavorare con le istantanee](#)
- [Usa il AWS CLI per lavorare con le istantanee](#)
- [Usa le SimSpace Weaver API per lavorare con le istantanee](#)
- [Utilizzo delle istantanee con AWS CloudFormation](#)

- [Domande frequenti sugli snapshot](#)

Casi d'uso per uno snapshot

Torna a uno stato precedente ed esplora gli scenari di ramificazione

Puoi creare un'istantanea della tua simulazione per salvarla in uno stato specifico. È quindi possibile creare più nuove simulazioni da quell'istantanea ed esplorare diversi scenari che potrebbero derivare da quello stato.

Best practice del disaster recovery e della sicurezza

Ti consigliamo di eseguire regolarmente il backup della simulazione, in particolare per le simulazioni che durano più di 1 ora o utilizzano più lavoratori. I backup possono aiutarti a riprenderti da disastri e incidenti di sicurezza. Le istantanee forniscono un modo per eseguire il backup della simulazione. Le istantanee richiedono che i file.zip dell'app esistano nella stessa posizione in Amazon S3 in cui si trovavano prima. Se devi poter spostare i file.zip della tua app in un'altra posizione, devi utilizzare una soluzione di backup personalizzata.

Per ulteriori informazioni su altre best practice, consulta [Le migliori pratiche quando si lavora con SimSpace Weaver](#) e [Best practice relative alla sicurezza di SimSpace Weaver](#).

Estendi la durata della tua simulazione

La tua risorsa di simulazione è la rappresentazione della tua simulazione in SimSpace Weaver. Tutte le risorse di simulazione hanno un' `MaximumDuration` impostazione. Una risorsa di simulazione si arresta automaticamente quando raggiunge la sua `MaximumDuration`. Il valore massimo di `MaximumDuration` è 14D (14 giorni).

Se hai bisogno che la tua simulazione duri più a lungo della relativa risorsa `MaximumDuration` di simulazione, puoi creare un'istantanea prima che la risorsa di simulazione raggiunga la sua `MaximumDuration`. Puoi avviare una nuova simulazione (creare una nuova risorsa di simulazione) con la tua istantanea. SimSpace Weaver inizializza i dati dell'entità dall'istantanea, avvia le stesse app spaziali e di servizio eseguite in precedenza e ripristina l'orologio. Puoi avviare le tue app personalizzate ed eseguire qualsiasi inizializzazione personalizzata aggiuntiva. Puoi impostare la nuova risorsa `MaximumDuration` di simulazione su un valore diverso all'avvio.

Usa l'SimSpace Weaver SDK per lavorare con le istantanee

Puoi utilizzare gli script forniti nell'SimSpace Weaver SDK (versione minima 1.13) per creare e utilizzare istantanee.

L'SimSpace Weaver SDK organizza le simulazioni per progetto. Puoi avviare più simulazioni da un singolo progetto. Ognuna di queste simulazioni utilizza lo stesso schema e gli stessi file.zip dell'app. Gli script SimSpace Weaver dell'app SDK collocano le risorse per una simulazione in uno specifico bucket Amazon S3 in base al nome, al numero e al numero del progetto. Account AWS Regione AWS Gli script funzionano con file di istantanee che si trovano in una snapshot cartella nella radice di quel bucket. L'URI Amazon S3 della snapshot cartella ha il seguente formato:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Esempio

- Nome progetto: MyProject
- Account AWSnumero: 111122223333
- Regione AWS: us-west-2
- Cartella istantanee Amazon S3 URI: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot

Se desideri utilizzare un bucket Amazon S3 diverso, consulta:

Altri modi per lavorare con gli snapshot

- [Console SimSpace Weaver](#)
- [AWS CLI](#)
- [API SimSpace Weaver](#)

Argomenti

- [Usa l'SimSpace Weaver SDK per creare un'istananea](#)
- [Usa l'SimSpace Weaver SDK per avviare una simulazione da un'istananea](#)
- [Usa l'SimSpace Weaver SDK per avviare rapidamente una simulazione da un'istananea](#)
- [Usa l'SimSpace Weaver SDK per elencare le istantanee di un progetto](#)

Usa l'SimSpace Weaver SDK per creare un'istantanea

Per creare un'istantanea, la simulazione deve essere nello STARTED stato. La creazione dell'istantanea inizia dopo il completamento del segno di spunta corrente. SimSpace Weaver interrompe l'invio di segni di spunta alle app ma lo stato dell'orologio lo indica STARTED comunque. Lo stato della simulazione cambia in. SNAPSHOT_IN_PROGRESS Al termine dell'istantanea, lo stato della simulazione torna STARTED e le app ricevono nuovamente i segni di spunta.

Per creare una snapshot

1. Al prompt dei comandi di Windows, vai alla cartella degli strumenti del tuo progetto.

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

2. Se non conosci il nome della tua simulazione, chiama l'`list-simulationsAPI` per visualizzare un elenco delle tue risorse di simulazione. Assicurati che lo stato della simulazione sia STARTED.

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

3. Esegui lo `create-snapshot` script per il tuo progetto.

```
.\create-snapshot-project-name.bat --simulation simulation-name
```

Esempio

```
.\create-snapshot-MyProject.bat --simulation MyProjectSimulation_23-04-29_12_00_00
```

SimSpace Weaver crea il file di istantanea nel bucket degli artefatti per il tuo progetto.

Esempio

- Nome progetto: MyProject
- Account AWS numero: 111122223333
- Regione AWS: us-west-2
- Cartella istantanee Amazon S3 URI: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot
- Nome della simulazione: MyProjectSimulation_23-04-29_12_00_00

- Ora dell'istantanea: 29 aprile 2023, 15:30:27 UTC
- Nome del file dell'istantanea:
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
- File istantaneo (URI Amazon S3): s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot/MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip

Usa l'App SimSpace Weaver SDK per avviare una simulazione da un'istantanea

Quando si utilizza uno script SDK dell'app per avviare una simulazione da un'istantanea, gli script creano un nuovo nome di simulazione nello stesso modo in cui avviano una simulazione senza un'istantanea.

Il file snapshot deve esistere nella posizione della copia istantanea in Amazon S3 con il seguente URI Amazon S3:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

I file.zip dell'app devono trovarsi nella stessa posizione in cui si trovavano al momento della creazione dell'istantanea.

SimSpace Weaver crea una nuova risorsa di simulazione, inizializza State Fabric con i dati delle entità memorizzati nell'istantanea, avvia nuove istanze delle stesse app spaziali e di servizio in esecuzione al momento della creazione dell'istantanea e imposta l'orologio sul segno di spunta appropriato. È necessario avviare le app personalizzate separatamente tramite la normale procedura.

Lo `start-from-snapshot` script è la versione istantanea dello `start-simulation` script. Proprio come lo `start-simulation` script, lo `start-from-snapshot` script non avvia l'orologio per te. È necessario avviare l'orologio separatamente.

Per avviare una simulazione da un'istantanea

1. Al prompt dei comandi di Windows, vai alla cartella degli strumenti del tuo progetto.

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

2. Eseguire lo script `start-from-snapshot`.

```
.\start-from-snapshot-project-name.bat --snapshot-s3-file snapshot-file-name
```

Esempio

```
.\start-from-snapshot-MyProject.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Usa l'SimSpace Weaverapp SDK per avviare rapidamente una simulazione da un'istantanea

È possibile avviare rapidamente una simulazione da un'istantanea. È simile a un avvio rapido senza istantanea.

Il file snapshot deve esistere nella posizione della copia istantanea in Amazon S3 con il seguente URI Amazon S3:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

I file.zip dell'app devono trovarsi nella stessa posizione in cui si trovavano al momento della creazione dell'istantanea.

SimSpace Weaver crea una nuova risorsa di simulazione, inizializza State Fabric con i dati delle entità memorizzati nell'istantanea, avvia nuove istanze delle stesse app spaziali e di servizio in esecuzione al momento della creazione dell'istantanea e imposta l'orologio sul segno di spunta appropriato. È necessario avviare le app personalizzate separatamente tramite la normale procedura.

Lo `quick-start-from-snapshot` script è la versione istantanea dello `quick-start` script. Proprio come lo `quick-start` script, lo `quick-start-from-snapshot` script avvia l'orologio per te. Avvia anche l'app di visualizzazione per il progetto di esempio di pathfinding.

Per avviare rapidamente una simulazione da un'istantanea

1. Al prompt dei comandi di Windows, vai alla cartella degli strumenti del tuo progetto.

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

2. Eseguire lo script `quick-start-from-snapshot`.

```
.\quick-start-from-snapshot-project-name-cli.bat --snapshot-s3-file snapshot-file-name
```

Esempio

```
.\quick-start-from-snapshot-MyProject-cli.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Usa l'SimSpace Weaver SDK per elencare le istantanee di un progetto

È possibile utilizzare lo `list-snapshots` script per elencare le istantanee di un progetto. Questo script elenca i file nella snapshot cartella del progetto. I progetti sono esclusivi dell'SimSpace Weaver SDK, quindi puoi farlo solo con gli script dell'app SDK e solo per i progetti. Lo script presuppone che tutti i file nella snapshot cartella su Amazon S3 siano file snapshot. Se sposti o elimini file dalla cartella, tali file non verranno visualizzati nell'elenco.

Per elencare gli snapshot di un progetto

1. Al prompt dei comandi di Windows, vai alla cartella degli strumenti del tuo progetto.

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

2. Eseguire lo script `list-snapshots`.

```
.\list-snapshots-project-name.bat
```

Esempio

```
.\list-snapshots-MyProject.bat
```

Usa la SimSpace Weaver console per lavorare con le istantanee

Puoi utilizzare la SimSpace Weaver console per creare uno snapshot della simulazione.

Altri modi per lavorare con gli snapshot

- [SimSpace Weaverscript SDK per app](#)
- [AWS CLI](#)
- [API SimSpace Weaver](#)

Argomenti

- [Usa la console per creare uno snapshot](#)
- [Usa la console per avviare una simulazione da un'istantanea](#)

Usa la console per creare uno snapshot

Per creare una snapshot

1. Accedi AWS Management Console e connettiti alla [SimSpace Weaverconsole](#).
2. Scegliere Simulazioni dal riquadro di navigazione.
3. Seleziona il pulsante di opzione accanto al nome della simulazione. Lo stato della tua simulazione deve essere Avviato.
4. Nella parte superiore della pagina, scegli Crea uno snapshot.
5. In Impostazioni istantanee, per la destinazione Snapshot, inserisci l'URI Amazon S3 di un bucket o di un bucket e della cartella in cui desideri SimSpace Weaver creare la tua istantanea. Puoi scegliere Browse S3 se preferisci sfogliare i bucket disponibili e selezionare una località.

Important

Il bucket Amazon S3 deve trovarsi nella Regione AWS stessa della simulazione.

Note

SimSpace Weaver crea una snapshot cartella all'interno della destinazione dell'istantanea selezionata. SimSpace Weaver crea il file snapshot .zip in quella snapshot cartella.

6. Scegli Create snapshot (Crea snapshot).

Usa la console per avviare una simulazione da un'istantanea

Per avviare una simulazione da un'istantanea, il file.zip dello snapshot deve esistere in un bucket Amazon S3 a cui la simulazione possa accedere. La tua simulazione utilizza le autorizzazioni definite nel ruolo dell'app selezionato all'avvio della simulazione. Tutti i file.zip dell'app della simulazione originale devono esistere nelle stesse posizioni in cui sono state create l'istantanea.

Per avviare una simulazione da un'istantanea

1. Accedi AWS Management Console e connettiti alla [SimSpace Weaverconsole](#).
2. Scegliere Simulazioni dal riquadro di navigazione.
3. Nella parte superiore della pagina, scegli Avvia simulazione.
4. Sotto Impostazioni della simulazione inserisci un nome e una descrizione facoltativa della simulazione. Il nome della simulazione deve essere univoco nel tuo Account AWS
5. Per il metodo di avvio della simulazione, scegli Usa uno snapshot in Amazon S3.
6. Per l'URI Amazon S3 per snapshot, inserisci l'URI Amazon S3 del tuo file snapshot oppure scegli Browse S3 per sfogliare e selezionare il file.

Important

Il bucket Amazon S3 deve trovarsi nella Regione AWS stessa della simulazione.

7. Per il ruolo IAM, seleziona il ruolo dell'app che verrà utilizzato dalla simulazione.
8. In Durata massima, inserisci la quantità massima di tempo per cui la risorsa di simulazione deve essere eseguita. Il valore massimo è 14D. Per ulteriori informazioni sulla durata massima, consulta [_](#).
9. In Tag - opzionale, scegli Aggiungi nuovo tag se desideri aggiungere un tag.
10. Scegli Avvia simulazione.

Usa il AWS CLI per lavorare con le istantanee

È possibile utilizzare il AWS CLI per chiamare le SimSpace Weaver API da un prompt dei comandi. È necessario che AWS CLI sia installato e configurato correttamente. Per ulteriori informazioni, consulta [AWS Command Line Interface Istruzioni per l'installazione o aggiornamento della più recente della AWS CLI](#) della.

Altri modi per lavorare con gli snapshot

- [SimSpace Weaverscript SDK per app](#)
- [Console SimSpace Weaver](#)
- [API SimSpace Weaver](#)

Argomenti

- [Usa il AWS CLI per creare un'istantanea](#)
- [Usa il AWS CLI per avviare una simulazione da un'istantanea](#)

Usa il AWS CLI per creare un'istantanea

Per creare una snapshot

- Al prompt dei comandi, chiama l'CreateSnapshotAPI.

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

Parametri

simulazione

Il nome di una simulazione avviata. Puoi usarlo `aws simspaceweaver list-simulations` per vedere i nomi e gli stati delle tue simulazioni.

destinazione

Una stringa che specifica il bucket Amazon S3 di destinazione e il key prefix oggetto opzionale per il file snapshot. Il key prefix oggetto è in genere una cartella nel bucket. SimSpace Weaver crea la tua istantanea all'interno di una snapshot cartella in questa destinazione.

Important

Il bucket Amazon S3 deve trovarsi nella Regione AWS stessa della simulazione.

Esempio

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

Per ulteriori informazioni sull'CreateSnapshotAPI, consulta la [CreateSnapshot](#) Guida di riferimento dell'AWS SimSpace WeaverAPI.

Usa il AWS CLI per avviare una simulazione da un'istantanea

Per avviare una simulazione da un'istantanea

- Al prompt dei comandi, chiama l'StartSimulationAPI.

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

Parametri

name

Il nome della nuova simulazione. Il nome della simulazione deve essere univoco nel tuo Account AWS. Puoi usarlo `aws simspaceweaver list-simulations` per vedere i nomi delle tue simulazioni esistenti.

filato di ruolo

L'Amazon Resource Name (ARN) del ruolo dell'app che verrà utilizzato dalla simulazione.

snapshot-s3-posizione

Una stringa che specifica il bucket Amazon S3 e la chiave dell'snapshot.

Important

Il bucket Amazon S3 deve trovarsi nella Regione AWS stessa della simulazione.

Esempio

```
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
```


Per ulteriori informazioni sull'`StartSimulationAPI`, consulta la [StartSimulation](#) Guida di riferimento dell'AWS SimSpace WeaverAPI.

Usa le SimSpace Weaver API per lavorare con le istantanee

Puoi chiamare direttamente le SimSpace Weaver API per lavorare con le istantanee. Per informazioni sulle API, consulta l'[AWS SimSpace WeaverAPI Reference](#).

Altri modi per lavorare con gli snapshot

- [SimSpace Weaverscript SDK per app](#)
- [Console SimSpace Weaver](#)
- [AWS CLI](#)

Creazione di una snapshot

Puoi chiamare l'`CreateSnapshotAPI` per creare un'istananea per una simulazione. Lo stato della simulazione deve essere `STARTED`. SimSpace Weaver crea il file snapshot in una snapshot cartella nel bucket Amazon S3 e nel prefisso oggetto che hai specificato. Per ulteriori informazioni, consulta [CreateSnapshot](#) nella documentazione di riferimento dell'API AWS SimSpace Weaver.

Avvia una simulazione da uno snapshot

Puoi fornire un'istananea quando chiami l'`StartSimulationAPI` per avviare una nuova simulazione. Fornisci una stringa JSON come argomento per il `SnapshotS3Location` parametro. La stringa specifica il nome del bucket Amazon S3 e la chiave dell'oggetto del file snapshot. Se fornisci un `SnapshotS3Location`, non puoi fornire un `SchemaS3Location`. Per ulteriori informazioni, consulta [StartSimulation](#) nella documentazione di riferimento dell'API AWS SimSpace Weaver.

Domande frequenti sugli snapshot

La mia simulazione continua a funzionare durante un'istananea?

Le tue risorse di simulazione continuano a funzionare durante un'istananea e continui a ricevere addebiti di fatturazione per quel periodo. Il tempo conta per la durata massima della simulazione. Le tue app non ricevono segni di spunta mentre l'istananea è in corso. Se lo stato dell'orologio era `STARTED` quando è iniziata la creazione dell'istananea, l'orologio indicherà comunque `STARTED`

lo stato. Le app ricevono nuovamente i segni di spunta al termine della snapshot. Se lo stato dell'orologio era STOPPED, lo stato dell'orologio rimarrà STOPPED. Nota che una simulazione con uno STARTED stato è in esecuzione anche se lo stato dell'orologio lo è STOPPED.

Cosa succede se è in corso un'istantanea e la mia simulazione raggiunge la durata massima?

La simulazione terminerà l'istantanea e si interromperà non appena il processo di creazione dell'istantanea terminerà (con successo o senza successo). Ti consigliamo di testare prima il processo di snapshot per scoprire quanto tempo impiega, le dimensioni del file di istantanea che puoi aspettarti e se deve essere completato correttamente.

Cosa succede se interrompo una simulazione con un'istantanea in corso?

Un'istantanea in corso si interrompe immediatamente quando si interrompe la simulazione. Non creerà un file di istantanee.

Come posso interrompere un'istantanea in corso?

L'unico modo per interrompere un'istantanea in corso è interrompere la simulazione. Non è possibile riavviare una simulazione dopo averla arrestata.

Quanto tempo ci vorrà per completare uno snapshot?

Il tempo necessario per creare un'istantanea dipende dalla simulazione. Ti consigliamo di testare prima il processo di creazione delle istantanee per scoprire quanto tempo ci vorrà per la simulazione.

Quanto sarà grande il mio file di istantanea?

La dimensione di un file di istantanea dipende dalla simulazione. Ti consigliamo di testare prima il processo di creazione di istantanee per scoprire quanto potrebbe essere grande il file per la tua simulazione.

Messaggistica

L'API di messaggistica semplifica la comunicazione tra applicazioni all'interno della simulazione. Le API per inviare e ricevere messaggi fanno parte dell'SDK dell'SimSpace Weaverapp. La messaggistica attualmente utilizza un approccio ottimale per inviare e ricevere messaggi. SimSpace Weaver tenta di inviare/ricevere messaggi alla prossima simulazione, ma non ci sono garanzie sull'orario di consegna, ordinazione o arrivo.

Argomenti

- [Casi d'uso per la messaggistica](#)
- [Utilizzo delle API di messaggistica](#)
- [Quando usare la messaggistica](#)
- [Suggerimenti per l'utilizzo della messaggistica](#)
- [Errori di messaggistica e risoluzione dei problemi](#)

Casi d'uso per la messaggistica

Comunica tra applicazioni di simulazione

Utilizza l'API di messaggistica per comunicare tra le applicazioni della simulazione. Usala per modificare lo stato delle entità a distanza, modificare il comportamento delle entità o trasmettere informazioni all'intera simulazione.

Conferma della ricezione di un messaggio

I messaggi inviati contengono informazioni sul mittente nell'intestazione del messaggio. Utilizza queste informazioni per inviare una risposta di conferma alla ricezione di un messaggio.

Inoltra i dati ricevuti da un'app personalizzata ad altre app all'interno della simulazione

La messaggistica non sostituisce il modo in cui i client si connettono alle app personalizzate in esecuzione in SimSpace Weaver. Tuttavia, la messaggistica consente agli utenti di inoltrare i dati dalle app personalizzate che ricevono i dati dei client ad altre app che non dispongono di una connessione esterna. Il flusso di messaggi può funzionare anche in senso inverso, consentendo alle app senza connessione esterna di inoltrare i dati a un'app personalizzata e quindi a un client.

Utilizzo delle API di messaggistica

Le API di messaggistica sono contenute nell'SDK dell'App SimSpace Weaver (versione minima 1.16.0). La messaggistica è supportata in C++, Python e nelle nostre integrazioni con Unreal Engine 5 e Unity.

Esistono due funzioni che gestiscono le transazioni di messaggi: `SendMessage` e `ReceiveMessages`. Tutti i messaggi inviati contengono una destinazione e un payload.

L'API `ReceiveMessages` restituisce un elenco di messaggi attualmente presenti nella coda dei messaggi in entrata di un'app.

C++

Invia messaggio

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

Ricevere messaggi

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

Python

Inviare un messaggio

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

Ricevere messaggi

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

Argomenti

- [Invio di messaggi](#)
- [Ricezione di messaggi](#)
- [Rispondere al mittente](#)

Invio di messaggi

I messaggi sono costituiti da una transazione (simile ad altre chiamate dell'API Weaver), un payload e una destinazione.

Payload del messaggio

Il payload dei messaggi è una struttura dati flessibile fino a 256 byte. Consigliamo quanto segue come best practice per creare i payload dei messaggi.

Per creare il payload dei messaggi

1. Crea una struttura di dati (ad esempio `struct` in C++) che definisca il contenuto del messaggio.
2. Crea il payload del messaggio che contiene i valori da inviare nel messaggio.
3. Crea l'`MessagePayload` oggetto.

Destinazione del messaggio

La destinazione di un messaggio è definita dall'`MessageEndpoint` oggetto. Ciò include sia un tipo di endpoint che un ID di endpoint. L'unico tipo di endpoint attualmente supportato è `Partition`, che consente di indirizzare messaggi ad altre partizioni nella simulazione. L'ID endpoint è l'ID di partizione della destinazione di destinazione.

Puoi fornire solo 1 indirizzo di destinazione in un messaggio. Crea e invia più messaggi se desideri inviare messaggi a più di 1 partizione contemporaneamente.

Per indicazioni su come risolvere un endpoint di messaggio da una posizione, vedere. [Suggerimenti per l'utilizzo della messaggistica](#)

Inviare il messaggio

È possibile utilizzare l'`SendMessageAPI` dopo aver creato gli oggetti di destinazione e di payload.

C++

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

Python

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

Esempio completo di invio di messaggi

L'esempio seguente mostra come è possibile costruire e inviare un messaggio generico. Questo esempio invia 16 messaggi singoli. Ogni messaggio contiene un payload con un valore compreso tra 0 e 15 e il segno di spunta di simulazione corrente.

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

```

Ricezione di messaggi

SimSpace Weaver consegna i messaggi nella coda dei messaggi in entrata di una partizione. Utilizza l'ReceiveMessagesAPI per ottenere un MessageList oggetto che contiene i messaggi dalla coda. Elabora ogni messaggio con l'ExtractMessageAPI per ottenere i dati del messaggio.

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

```

Rispondere al mittente

Ogni messaggio ricevuto contiene un'intestazione del messaggio con informazioni sul mittente originale del messaggio. Puoi usare `message.header.source_endpoint` per inviare una risposta.

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
```

Quando usare la messaggistica

La messaggistica in SimSpace Weaver offre un altro modello per lo scambio di informazioni tra applicazioni di simulazione. Gli abbonamenti forniscono un meccanismo pull per leggere i dati da applicazioni o aree specifiche della simulazione; i messaggi forniscono un meccanismo push per inviare dati ad applicazioni o aree specifiche della simulazione.

Di seguito sono riportati due casi d'uso in cui è più utile inviare dati utilizzando la messaggistica piuttosto che estrarre o leggere i dati tramite un abbonamento.

Example 1: Invio di un comando a un'altra app per modificare la posizione di un'entità

```
// 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);
```

Sul lato ricevente, l'app aggiorna la posizione dell'entità e la scrive nello 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: Invio di un messaggio di creazione dell'entità a un'app spaziale

```
struct WeaverMessage
{
    const Aws::WeaverRuntime::Api::TypeId messageTypeId;
};
```

```

const Aws::WeaverRuntime::Api::TypeId k_createEntityMessageTypeId = { 1 };

struct CreateEntityMessage : WeaverMessage
{
    const Vector3 position;
    const Aws::WeaverRuntime::Api::TypeId typeId;
};

CreateEntityMessage messageData {
    k_createEntityMessageTypeId,
    Vector3{ position.GetX(), position.GetY(), position.GetZ() },
    Api::TypeId { 0 }
}

WEAVERRUNTIME_TRY(Api::MessageEndpoint destination,
    Api::Utils::MessageEndpointResolver::ResolveFromPosition(
        transaction, "MySpatialDomain", DemoFramework::ToVector2F32(position)
    ));

Api::MessagePayload payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const uint8_t*>(&messageData),
    sizeof(CreateEntityMessage));

Api::SendMessage(transaction, payload, destination);

```

Sul lato ricevente, l'app crea una nuova entità nello State Fabric e ne aggiorna la posizione.

```

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)));
}

```

Suggerimenti per l'utilizzo della messaggistica

Risolvi un endpoint da una posizione o dal nome dell'app

È possibile utilizzare la `AllPartitions` funzione per ottenere i limiti spaziali e l'ID di dominio necessari per determinare gli ID delle partizioni dei messaggi e le destinazioni dei messaggi. Tuttavia, se conosci la posizione in cui desideri inviare il messaggio, ma non il relativo ID di partizione, puoi utilizzare la funzione. `MessageEndpointResolver`

```

/**
 * 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);
};

```

Serializzazione e deserializzazione del payload del messaggio

È possibile utilizzare le seguenti funzioni per creare e leggere i payload dei messaggi. Per ulteriori informazioni, consulta `MessagingUtils.h` nella libreria SDK dell'app sul tuo sistema locale.

```

/**
 * 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());
}

```

Errori di messaggistica e risoluzione dei problemi

È possibile che si verifichino i seguenti errori quando si utilizzano le API di messaggistica.

Errori di risoluzione degli endpoint

Questi errori possono verificarsi prima che un'app invii un messaggio.

Controllo del nome di dominio

L'invio di un messaggio a un dispositivo non valido genera il seguente errore:

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

Questo può accadere quando si tenta di inviare un messaggio a un'app personalizzata e tale app personalizzata non ha ancora partecipato alla simulazione. Usa l'`DescribeSimulationAPI` per assicurarti che l'app personalizzata sia stata avviata prima di inviarle un messaggio. Questo comportamento è lo stesso in SimSpace Weaver Local e inCloud AWS.

Controllo della posizione

Il tentativo di risolvere un endpoint con un nome di dominio valido ma una posizione non valida genera il seguente errore.

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

Ti consigliamo di utilizzarlo `MessageEndpointResolver` nella `MessageUtils` libreria contenuta nell'SDK dell'`SimSpace Weaverapp`.

Errori di invio dei messaggi

I seguenti errori possono verificarsi quando un'app invia un messaggio.

È stato superato il limite di invio di messaggi per app, per segno di spunta

Il limite attuale per il numero di messaggi che possono essere inviati per app per segno di simulazione è 128. Le chiamate successive con lo stesso segno di spunta falliranno con il seguente errore:

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


SimSpace Weaver tenta di inviare messaggi non inviati con il segno di spunta successivo. Riduci la frequenza di invio per risolvere il problema. Combina payload di messaggi inferiori al limite di 256 byte per ridurre il numero di messaggi in uscita.

Questo comportamento è lo stesso in SimSpace Weaver Local e in Cloud AWS

Il limite di dimensione del payload dei messaggi è stato superato

Il limite attuale per la dimensione del payload dei messaggi è di 256 byte sia in che SimSpace Weaver Local in Cloud AWS L'invio di un messaggio con un payload superiore a 256 byte genera il seguente errore:

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

SimSpace Weaver controlla ogni messaggio e rifiuta solo quelli che superano il limite. Ad esempio, se l'app tenta di inviare 10 messaggi e 1 non supera il controllo, solo quel messaggio viene rifiutato. SimSpace Weaver invia gli altri 9 messaggi.

Questo comportamento è lo stesso in SimSpace Weaver Local e in Cloud AWS.

La destinazione è la stessa della fonte

Le app non possono inviare messaggi alle partizioni di cui sono proprietarie. Se un'app invia un messaggio a una partizione di sua proprietà, viene visualizzato il seguente errore.

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

Questo comportamento è lo stesso in SimSpace Weaver Local e in Cloud AWS

Messaggistica con il massimo impegno

SimSpace Weaver non garantisce la consegna dei messaggi. Il servizio cercherà di completare la consegna dei messaggi al successivo segno di spunta di simulazione, ma i messaggi potrebbero andare persi o subire ritardi.

Le migliori pratiche quando si lavora con SimSpace Weaver

Consigliamo le seguenti best practice quando si lavora con SimSpace Weaver.

Argomenti

- [Imposta gli allarmi di fatturazione](#)
- [Utilizzare SimSpace Weaver Local](#)
- [Interrompi le simulazioni che non ti servono](#)
- [Elimina le risorse che non ti servono](#)
- [Avere dei backup](#)

Imposta gli allarmi di fatturazione

È facile allocare le risorse AWS e lasciarle sempre attive, anche quando non sono più necessarie. Ciò può comportare costi eccessivi che possono sorprendere una volta ricevuta la fattura. Puoi configurare un allarme in Amazon CloudWatch che si attiverà e ti avviserà quando i costi superano una soglia da te impostata. Puoi esaminare i costi utilizzando strumenti di gestione dei costi. Per ulteriori informazioni, consultare:

- [Crea un allarme di fatturazione per monitorare gli addebiti stimati AWS](#)
- [Cos'è AWS Cost Management?](#)

Utilizzare SimSpace Weaver Local

Ti consigliamo di utilizzare SimSpace Weaver Local per sviluppare e testare le tue simulazioni prima di caricarle sul SimSpace Weaver servizio in Cloud AWS. I vantaggi dello sviluppo con SimSpace Weaver Local includono:

- Non c'è bisogno di aspettare caricamenti di grandi dimensioni
- Nessun limite al numero di simulazioni locali che puoi creare
- Non ti viene addebitato alcun costo per il tempo di elaborazione sul tuo computer locale
- Accesso diretto all'output della console dalle tue app
- Modifica, ricostruisci e riavvia la simulazione locale senza doverla ricreare nel Cloud AWS

Interrompi le simulazioni che non ti servono

Ti verranno addebitati i costi di fatturazione per una simulazione mentre è in esecuzione. È necessario interrompere una simulazione per non ricevere più addebiti. Le simulazioni in esecuzione vengono conteggiate anche ai fini della quota per il numero massimo di simulazioni. Una simulazione in esecuzione con registrazione configurata può anche generare grandi quantità di log, per i quali vengono addebitati anche costi di fatturazione. È necessario interrompere qualsiasi simulazione non necessaria per evitare costi aggiuntivi.

Important

L'interruzione dell'orologio di simulazione non interrompe la simulazione, l'orologio interrompe semplicemente la pubblicazione di segni di spunta nelle tue app. Non puoi riavviare una simulazione dopo averla interrotta.

Elimina le risorse che non ti servono

Ogni simulazione creata crea SimSpace Weaver anche risorse in altri AWS servizi. Puoi ottenere addebiti di fatturazione per risorse e dati in questi altri servizi. Le simulazioni in esecuzione e quelle con esito negativo vengono conteggiate ai fini della quota per il numero massimo di simulazioni. È necessario eliminare le simulazioni non riuscite non necessarie in modo da poter avviare nuove simulazioni. Quando si elimina una simulazione, è possibile che le risorse per la simulazione presenti in altri AWS servizi non vengano eliminate. Ad esempio, tutti i dati di log di simulazione in Amazon CloudWatch Logs rimarranno lì finché non li elimini. Ti verranno addebitati i costi di fatturazione per quei dati di log. Dovresti pulire tutte le risorse associate per le tue simulazioni se non ne hai più bisogno. Per ulteriori informazioni, consulta il [Passaggio 6: interrompi e ripulisci la simulazione](#) tutorial di avvio rapido.

Avere dei backup

È una buona idea disporre di backup e piani di backup per tutto. Non dovresti dare per scontato che solo perché ci sono i tuoi dati non AWS sia necessario eseguirne il backup. È necessario creare il proprio sistema se è necessario eseguire il backup dello stato di simulazione. Valuta la possibilità di utilizzarne più di uno Regioni AWS e di predisporre un piano per poter passare rapidamente il carico di lavoro di produzione a un altro, Regione AWS se necessario. Per ulteriori informazioni su Regioni AWS tale supporto SimSpace Weaver, consulta [SimSpace Endpoint e quote Weaver](#).

Sicurezza dell'AWS SimSpace Weaver

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il [modello di responsabilità condivisa](#) fa riferimento ad una sicurezza del cloud e nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che esegue i servizi AWS in Cloud AWS. AWS fornisce inoltre i servizi che è possibile utilizzare in modo sicuro. Revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei [Programmi di conformità AWS](#). Per informazioni sui programmi di conformità applicabili a AWS SimSpace Weaver, consulta [Servizi AWS scoperti dal programma di conformità](#).
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal servizio AWS che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione serve a facilitare la comprensione dell'applicazione del modello di responsabilità condivisa quando si utilizza l'AWS SimSpace Weaver. I seguenti argomenti illustrano come configurare l'AWS SimSpace Weaver per soddisfare gli obiettivi di sicurezza e conformità. Scoprirai anche come utilizzare altri servizi di AWS per monitorare e proteggere le risorse AWS SimSpace Weaver.

Argomenti

- [Protezione dei dati dell'AWS SimSpace Weaver](#)
- [Identity and Access Management per AWS SimSpace Weaver](#)
- [Registrazione e monitoraggio degli eventi di sicurezza AWS SimSpace Weaver](#)
- [Convalida della conformità per AWS SimSpace Weaver](#)
- [Resilienza nell'AWS SimSpace Weaver](#)
- [Sicurezza dell'infrastruttura in AWS SimSpace Weaver](#)
- [Analisi della configurazione e delle vulnerabilità in AWS SimSpace Weaver](#)
- [Best practice relative alla sicurezza di SimSpace Weaver](#)

Protezione dei dati dell'AWS SimSpace Weaver

Il [modello di responsabilità condivisa](#) di AWS si applica alla protezione dei dati in AWS SimSpace Weaver. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Inoltre, sei responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS che utilizzi. Per ulteriori informazioni sulla privacy dei dati, vedi [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog [AWS Shared Responsibility Model and GDPR](#) nel Blog sulla sicurezza AWS.

Per garantire la protezione dei dati, ti suggeriamo di proteggere le credenziali Account AWS e di configurare singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse AWS. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura la registrazione di log sulle attività di API e utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti in Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se necessiti di moduli crittografici convalidati FIPS 140-2 quando accedi ad AWS attraverso un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Questo vale quando si lavora con l'SimSpace Weaver e altri Servizi AWS utilizzando la console, l'API, la AWS CLI o gli SDK di AWS. I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia dei dati a riposo

I dati vengono considerati inattivi quando si trovano in un archivio di dati non volatile (persistente), ad esempio un disco. I dati che si trovano in archivi di dati volatili, come memoria e registri, non sono considerati inattivi.

Quando si utilizza SimSpace Weaver, gli unici dati inattivi sono:

- App e schemi caricati su Amazon Simple Storage Service (Amazon S3)
- Dati del registro di simulazione archiviati in Amazon CloudWatch

Gli altri dati SimSpace Weaver utilizzati internamente non persistono dopo l'interruzione della simulazione.

Per informazioni su come crittografare i dati inattivi, consulta:

- [Crittografa i tuoi dati in Amazon S3](#)
- [Crittografa i dati di registro](#)

Crittografia in transito

Le tue connessioni all'API SimSpace Weaver tramite AWS Command Line Interface (AWS CLI), AWS SDK e SimSpace Weaver app SDK utilizzano la crittografia TLS con il processo di [firma Signature Version 4](#). AWS gestisce l'autenticazione utilizzando le politiche di accesso definite da IAM per le credenziali di sicurezza utilizzate per la connessione.

Internamente, SimSpace Weaver utilizza TLS per connettersi agli altri AWS servizi che utilizza.

Important

Le comunicazioni tra le tue app e i loro clienti non implicano. SimSpace Weaver È tua responsabilità crittografare le comunicazioni con i client di simulazione, se necessario. Ti consigliamo di creare una soluzione per crittografare tutti i dati in transito tra le connessioni client.

Per saperne di più sui AWS servizi in grado di supportare le tue soluzioni di crittografia, consulta [il AWS Security Blog](#).

Riservatezza del traffico Internet

SimSpace Weaver le risorse di calcolo risiedono all'interno di 1 Amazon VPC condiviso da tutti i clienti. SimSpace Weaver Tutto il traffico interno dei SimSpace Weaver servizi rimane all'interno della AWS rete e non viaggia su Internet. La comunicazione tra i client di simulazione e le tue app viaggia su Internet.

Identity and Access Management per AWS SimSpace Weaver

AWS Identity and Access Management (IAM) è un Servizio AWS che consente agli amministratori di controllare in modo sicuro l'accesso alle risorse AWS. Gli amministratori IAM controllano chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) a utilizzare risorse SimSpace Weaver. IAM è un Servizio AWS che è possibile utilizzare senza alcun costo aggiuntivo.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Funzionamento di AWS SimSpace Weaver con IAM](#)
- [Esempi di policy basate su identità per AWS SimSpace Weaver](#)
- [Autorizzazioni create per te SimSpace Weaver](#)
- [Prevenzione del problema "confused deputy" tra servizi](#)
- [Risoluzione dei problemi di identità e accesso in AWS SimSpace Weaver](#)

Destinatari

Le modalità di utilizzo di AWS Identity and Access Management (IAM) cambiano in base alle operazioni eseguite in SimSpace Weaver.

Utente del servizio: se utilizzi il servizio SimSpace Weaver per eseguire il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità SimSpace Weaver utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette

all'amministratore. Se non riesci ad accedere a una funzionalità di SimSpace Weaver, consulta [Risoluzione dei problemi di identità e accesso in AWS SimSpace Weaver](#).

Amministratore del servizio: se sei il responsabile delle risorse SimSpace Weaver presso la tua azienda, probabilmente disponi dell'accesso completo a SimSpace Weaver. Il tuo compito è determinare le caratteristiche e le risorse SimSpace Weaver a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con SimSpace Weaver, consulta [Funzionamento di AWS SimSpace Weaver con IAM](#).

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dei dettagli su come scrivere policy per gestire l'accesso a SimSpace Weaver. Per visualizzare policy basate su identità di SimSpace Weaver di esempio che puoi utilizzare in IAM, consulta [Esempi di policy basate su identità per AWS SimSpace Weaver](#).

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS con le credenziali di identità. Devi essere autenticato (connesso a AWS) come utente root Utente root dell'account AWS, come utente IAM o assumere un ruolo IAM.

Puoi accedere ad AWS come identità federata utilizzando le credenziali fornite attraverso un'origine di identità. Gli utenti AWS IAM Identity Center (Centro identità IAM), l'autenticazione Single Sign-On (SSO) dell'azienda e le credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Se accedi ad AWS tramite la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere alla AWS Management Console o al portale di accesso AWS. Per ulteriori informazioni sull'accesso ad AWS, consulta la sezione [Come accedere al tuo Account AWS](#) nella Guida per l'utente di Accedi ad AWS.

Se accedi ad AWS in modo programmatico, AWS fornisce un Software Development Kit (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare le richieste personalmente. Per ulteriori informazioni sulla firma delle richieste, consulta [Firma delle richieste AWS](#) nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare l'autenticazione a più fattori

(MFA) per aumentare la sicurezza dell'account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente di IAM.

Utente root di un Account AWS

Quando crei un Account AWS, inizi con una singola identità di accesso che ha accesso completo a tutti i Servizi AWS e le risorse nell'account. Tale identità è detta utente root Account AWS ed è possibile accedervi con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente di IAM.

Identità federata

Come best practice, richiedi agli utenti umani, compresi quelli che richiedono l'accesso di amministratore, di utilizzare la federazione con un provider di identità per accedere a Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory degli utenti aziendali, un provider di identità Web, AWS Directory Service, la directory Identity Center o qualsiasi utente che accede ai Servizi AWS utilizzando le credenziali fornite tramite un'origine di identità. Quando le identità federate accedono agli Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. È possibile creare utenti e gruppi in IAM Identity Center oppure connettersi e sincronizzarsi con un gruppo di utenti e gruppi nell'origine di identità per utilizzarli in tutte le applicazioni e gli Account AWS. Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center.

Utenti e gruppi IAM

Un [utente IAM](#) è una identità all'interno del tuo Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente di IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità all'interno di un Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un ruolo IAM nella AWS Management Console mediante lo [scambio di ruoli](#). È possibile assumere un ruolo chiamando un'azione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center.
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, per alcuni dei Servizi AWS, è possibile collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per

informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

- **Accesso multi-servizio:** alcuni Servizi AWS utilizzano funzionalità in altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Inoltro delle sessioni di accesso (FAS):** quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, tale utente o ruolo viene considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) assunto da un servizio per eseguire operazioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** è possibile utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 che eseguono richieste di AWS CLI o dell'API AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo dell'istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Per controllare l'accesso a AWS è possibile creare policy e collegarle a identità o risorse AWS. Una policy è un oggetto in AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste policy quando un principale IAM (utente, utente root o sessione ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata in AWS sotto forma di documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWSJSON per specificare l'accesso ai diversi elementi. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dalla AWS Management Console, la AWS CLI o l'API AWS.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy autonome che possono essere collegate a più utenti, gruppi e ruoli in Account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy gestite da AWS da IAM in una policy basata su risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3, AWS WAF e Amazon VPC sono esempi di servizi che supportano le ACL. Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta altri tipi di policy meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Policy di controllo dei servizi (SCP):** le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata degli Account

AWSmultipli di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. La SCP limita le autorizzazioni per le entità negli account membri, compreso ogni Utente root dell'account AWS. Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.

- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWSdetermina se consentire una richiesta quando sono coinvolti più tipi di policy, consultare [Logica di valutazione delle policy](#) nella Guida per l'utente di IAM.

Funzionamento di AWS SimSpace Weavercon IAM

Prima di utilizzare IAM per gestire l'accesso a SimSpace Weaver, scopri quali funzionalità di IAM sono disponibili per l'uso con SimSpace Weaver.

Funzionalità IAM che è possibile utilizzare con AWS SimSpace Weaver

Funzionalità IAM	Supporto di SimSpace Weaver
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì

Funzionalità IAM	Supporto di SimSpace Weaver
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per ottenere un quadro generale del funzionamento di SimSpace Weaver e altri servizi AWS con la maggior parte delle caratteristiche di IAM, consulta [Servizi AWS supportati da IAM](#) nella Guida per l'utente IAM.

Policy basate su identità per SimSpace Weaver

Supporta le policy basate su identità	Sì
---------------------------------------	----

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

Esempi di policy basate su identità per SimSpace Weaver

Per visualizzare esempi di policy basate su identità SimSpace Weaver, consulta [Esempi di policy basate su identità per AWS SimSpace Weaver](#).

Policy basate su risorse all'interno di SimSpace Weaver

Supporta le policy basate su risorse

No

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando l'entità principale e la risorsa si trovano in diversi Account AWS, un amministratore IAM nell'account attendibile deve concedere all'entità principale (utente o ruolo) anche l'autorizzazione per accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente IAM.

Operazioni di policy per SimSpace Weaver

Supporta le azioni di policy

Sì

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni di policy hanno spesso lo stesso nome dell'operazione API AWS. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di SimSpace Weaver, consulta [Operazioni definite da AWS SimSpace Weaver](#) nella Guida di riferimento per l'autorizzazione del servizio.

Le operazioni delle policy in SimSpace Weaver utilizzano il seguente prefisso prima dell'operazione:

```
simspaceweaver
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "simspaceweaver:action1",  
  "simspaceweaver:action2"  
]
```

Per visualizzare esempi di policy basate su identità SimSpace Weaver, consulta [Esempi di policy basate su identità per AWS SimSpace Weaver](#).

Risorse relative alle policy per SimSpace Weaver

Supporta le risorse di policy	Si
-------------------------------	----

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco di tipi di risorse SimSpace Weaver e dei relativi ARN, consulta [Risorse definite da AWS SimSpace Weaver](#) nella Guida di riferimento sull'autorizzazione del servizio. Per

informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la sezione [Operazioni definite da AWS SimSpace Weaver](#).

Per visualizzare esempi di policy basate su identità SimSpace Weaver, consulta [Esempi di policy basate su identità per AWS SimSpace Weaver](#).

Chiavi di condizione delle policy per SimSpace Weaver

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì
---	----

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione OR logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche per il servizio. Per visualizzare tutte le chiavi di condizione globali di AWS, consulta [Chiavi di contesto delle condizioni globali di AWS](#) nella Guida per l'utente di IAM.

Per visualizzare un elenco di chiavi di condizione SimSpace Weaver, consulta [Chiavi di condizione per AWS SimSpace Weaver](#) nella Guida di riferimento sull'autorizzazione del servizio. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta la sezione [Operazioni definite da AWS SimSpace Weaver](#).

Per visualizzare esempi di policy basate su identità SimSpace Weaver, consulta [Esempi di policy basate su identità per AWS SimSpace Weaver](#).

Liste di controllo degli accessi (ACL) in SimSpace Weaver

Supporta le ACL

No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato su attributi (ABAC) con SimSpace Weaver

Supporta ABAC (tag nelle policy)

Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, tali attributi sono denominati tag. È possibile collegare dei tag alle entità IAM (utenti o ruoli) e a numerose risorse AWS. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta [Che cos'è ABAC?](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con SimSpace Weaver

Supporta le credenziali temporanee	Si
------------------------------------	----

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, inclusi i Servizi AWS che funzionano con le credenziali temporanee, consulta [Servizi AWS supportati da IAM](#) nella Guida per l'utente IAM.

Le credenziali temporanee sono utilizzate se si accede alla AWS Management Console utilizzando qualsiasi metodo che non sia la combinazione di nome utente e password. Ad esempio, quando accedi ad AWS utilizzando il collegamento Single Sign-On (SSO) della tua azienda, tale processo crea in automatico credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Cambio di un ruolo \(console\)](#) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando la AWS CLI o l'API AWS. È quindi possibile utilizzare tali credenziali temporanee per accedere ad AWS. AWS consiglia di generare le credenziali temporanee dinamicamente anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

Autorizzazioni del principale tra servizi per SimSpace Weaver

Supporta sessioni di accesso diretto (FAS)	Si
--	----

Quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, si viene considerati un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Inoltro sessioni di accesso](#).

Ruoli di servizio per SimSpace Weaver

Supporta i ruoli di servizio	Si
------------------------------	----

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di SimSpace Weaver. Modifica i ruoli di servizio solo quando SimSpace Weaver fornisce le indicazioni per farlo.

Gli script SDK SimSpace Weaver dell'app utilizzano un AWS CloudFormation modello per creare risorse in altri AWS servizi a supporto della simulazione. Una di queste risorse è il ruolo dell'app per la simulazione. SimSpace Weaver assume il ruolo dell'app di eseguire azioni per conto dell'utente Account AWS, ad esempio scrivere i dati di registro in Logs. CloudWatch Per ulteriori informazioni sul ruolo dell'app, consulta [Autorizzazioni create per te SimSpace Weaver](#)

Ruoli collegati ai servizi per l'SimSpace Weaver

Supporta i ruoli collegati ai servizi

No

Un ruolo collegato ai servizi è un tipo di ruolo di servizio che è collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate su identità per AWS SimSpace Weaver

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse SimSpace Weaver. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, l'AWS Command Line Interface (AWS CLI) o l'API AWS. Per concedere agli utenti l'autorizzazione per eseguire operazioni sulle risorse di cui hanno bisogno, un

amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da SimSpace Weaver, incluso il formato degli ARN per ogni tipo di risorsa, consulta [Operazioni, risorse e chiavi di condizione per AWS SimSpace Weaver](#) nella Guida di riferimento per l'autorizzazione del servizio.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console di SimSpace Weaver](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Consenti agli utenti di creare ed eseguire simulazioni](#)

Best practice per le policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse SimSpace Weaver nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Nozioni di base sulle policy gestite da AWS e passaggio alle autorizzazioni con privilegio minimo: per le informazioni di base su come concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite da AWS che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente di AWS specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile

scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi inoltre utilizzare le condizioni per concedere l'accesso alle operazioni di servizio, ma solo se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente di IAM.
- Richiesta dell'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o utenti root nel tuo Account AWS, attiva MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console di SimSpace Weaver

Per accedere alla console AWS SimSpace Weaver è necessario disporre di un insieme di autorizzazioni minimo. Queste autorizzazioni devono consentire di elencare e visualizzare i dettagli relativi alle risorse SimSpace Weaver nel tuo Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario concedere le autorizzazioni minime della console agli utenti che effettuano chiamate solo alla AWS CLI o all'API AWS. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che gli utenti e i ruoli possano ancora utilizzare la SimSpace Weaver console, allega anche la policy SimSpace Weaver *ConsoleAccess* o la policy *ReadOnly* AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono allegate alla relativa identità utente. La policy include le autorizzazioni per completare questa azione sulla console o a livello di programmazione utilizzando la AWS CLI o l'API AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```


Consenti agli utenti di creare ed eseguire simulazioni

Questo esempio di policy IAM fornisce le autorizzazioni di base necessarie per creare ed eseguire simulazioni in 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"
        }
    }
}
]
```

Autorizzazioni create per te SimSpace Weaver

Quando crei un SimSpace Weaver progetto, il servizio creerà un ruolo AWS Identity and Access Management (IAM) con il nome `weaver-project-name-app-role` e una policy di fiducia IAM. La politica di fiducia SimSpace Weaver consente di assumere il ruolo in modo che possa eseguire operazioni per te.

Politica sulle autorizzazioni dei ruoli dell'app

Il ruolo dell'app di simulazione ha i seguenti criteri di autorizzazione.

```
{
  "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": "*"
    }
  ]
}

```

Politica di attendibilità dei ruoli dell'app

SimSpace Weaver aggiunge una relazione di fiducia al ruolo dell'app di simulazione come [politica di fiducia](#). SimSpace Weaver crea una politica di fiducia per ogni simulazione, simile all'esempio seguente.

```

{
  "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 questo esempio, il numero di conto è 111122223333 e il nome della simulazione è. MySimName Questi valori sono diversi nelle vostre politiche di fiducia.

Prevenzione del problema "confused deputy" tra servizi

Il [problema della confusione secondaria](#) è un problema di sicurezza in cui un'entità che non è autorizzata a eseguire un'azione può indurre un'entità con maggiori privilegi a eseguire l'azione. In AWS, la rappresentazione tra servizi può determinare un problema "confused deputy". La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Ti consigliamo di utilizzare le chiavi di contesto delle condizioni globali [aws:SourceArn](#) e [aws:SourceAccount](#) nelle policy delle risorse per limitare le autorizzazioni con cui AWS SimSpace Weaver fornisce un altro servizio alla risorsa. Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio il nome della risorsa Amazon (ARN) di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il valore di `aws:SourceArn` deve utilizzare l'ARN dell'estensione.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non conosci l'ARN completo dell'estensione o specifichi più estensioni, utilizza la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:simspaceweaver:*:111122223333:*`.

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in SimSpace Weaver per prevenire il problema confused deputy. Questa politica consentirà di SimSpace Weaver assumere il ruolo solo quando la richiesta proviene dall'account di origine specificato e viene fornita con l'ARN specificato. *In questo caso, SimSpace Weaver può assumere il ruolo di responsabile delle richieste provenienti da simulazioni solo nell'account del richiedente (111122223333) e solo nella regione specificata (us-west-2).*

```
{
```

```

"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/*"
      }
    }
  }
]
}

```

Un modo più sicuro per scrivere questa politica consiste nell'includere il nome della simulazione in `aws:SourceArn`, come mostrato nell'esempio seguente, che limita la politica a una simulazione denominata: `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"
    }
}
]
}

```

Quando includi `aws:SourceArn` esplicitamente un numero di account, puoi omettere l'Conditionamento test per `aws:SourceAccount` (consulta la [IAM User Guide](#) per maggiori informazioni), ad esempio nella seguente policy semplificata:

```

{
  "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"
        }
      }
    }
  ]
}

```

Risoluzione dei problemi di identità e accesso in AWS SimSpace Weaver

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di SimSpace Weaver e di IAM.

Argomenti

- [Non sono autorizzato a eseguire un'operazione in SimSpace Weaver](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)

- [Desidero visualizzare le mie chiavi di accesso](#)
- [Sono un amministratore e desidero consentire ad altri utenti di accedere a SimSpace Weaver](#)
- [Voglio consentire alle persone esterne al mio account Account AWS di accedere alle mie risorse SimSpace Weaver](#)

Non sono autorizzato a eseguire un'operazione in SimSpace Weaver

Se la AWS Management Console indica che non hai l'autorizzazione a eseguire un'operazione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` IAM prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa `my-example-widget` fittizia ma non dispone di autorizzazioni `simspaceweaver:GetWidget` fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
simspaceweaver:GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa `my-example-widget` utilizzando l'operazione `simspaceweaver:GetWidget`.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a SimSpace Weaver.

Alcuni Servizi AWS consentono di passare un ruolo esistente a tale servizio, invece di creare un nuovo ruolo di servizio o un ruolo collegato ai servizi. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in SimSpace Weaver. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Desidero visualizzare le mie chiavi di accesso

Dopo aver creato le chiavi di accesso utente IAM, è possibile visualizzare il proprio ID chiave di accesso in qualsiasi momento. Tuttavia, non è possibile visualizzare nuovamente la chiave di accesso segreta. Se perdi la chiave segreta, dovrai creare una nuova coppia di chiavi di accesso.

Le chiavi di accesso sono composte da due parti: un ID chiave di accesso (ad esempio AKIAIOSFODNN7EXAMPLE) e una chiave di accesso segreta (ad esempio, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). Come un nome utente e una password, è necessario utilizzare sia l'ID chiave di accesso sia la chiave di accesso segreta insieme per autenticare le richieste dell'utente. Gestisci le tue chiavi di accesso in modo sicuro mentre crei il nome utente e la password.

Important

Non fornire le chiavi di accesso a terze parti, neppure per aiutare a [trovare l'ID utente canonico](#). Se lo facessi, daresti a qualcuno accesso permanente al tuo Account AWS.

Quando crei una coppia di chiavi di accesso, ti viene chiesto di salvare l'ID chiave di accesso e la chiave di accesso segreta in una posizione sicura. La chiave di accesso segreta è disponibile solo al momento della creazione. Se si perde la chiave di accesso segreta, è necessario aggiungere nuove chiavi di accesso all'utente IAM. È possibile avere massimo due chiavi di accesso. Se se ne hanno già due, è necessario eliminare una coppia di chiavi prima di crearne una nuova. Per visualizzare le istruzioni, consulta [Gestione delle chiavi di accesso](#) nella Guida per l'utente di IAM.

Sono un amministratore e desidero consentire ad altri utenti di accedere a SimSpace Weaver

Per consentire ad altri utenti di accedere ad SimSpace Weaver, devi creare un'entità IAM (utente o ruolo) per la persona o l'applicazione che richiede l'accesso. Tale utente o applicazione utilizzerà le credenziali dell'entità per accedere ad AWS. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in SimSpace Weaver.

Per iniziare immediatamente, consulta [Creazione dei primi utenti e gruppi delegati IAM](#) nella Guida per l'utente di IAM.

Voglio consentire alle persone esterne al mio account Account AWS di accedere alle mie risorse SimSpace Weaver

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per capire se SimSpace Weaver supporta queste funzionalità, consulta [Funzionamento di AWS SimSpace Weaver con IAM](#).
- Per informazioni su come garantire l'accesso alle risorse negli Account AWS che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS in tuo possesso](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso alle risorse ad Account AWS di terze parti, consulta [Fornire l'accesso agli Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente IAM.

Registrazione e monitoraggio degli eventi di sicurezza AWS SimSpace Weaver

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni di SimSpace Weaver e delle soluzioni AWS. È necessario raccogliere i dati sul monitoraggio da tutte le parti della soluzione AWS per consentire un debug più facile di eventuali guasti in più punti.

AWS e SimSpace Weaver forniscono diversi strumenti per monitorare le tue risorse di simulazione e rispondere a potenziali incidenti.

Accedi in Amazon CloudWatch

SimSpace Weaver memorizza i dati di accesso. CloudWatch È possibile utilizzare questi registri per monitorare gli eventi della simulazione (come l'avvio e l'arresto delle app) e per il debug. Per ulteriori informazioni, consulta [SimSpace Weaver accessi in Amazon CloudWatch Logs](#).

CloudWatch Allarmi Amazon

Utilizzando Amazon CloudWatch alarms, controlla una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o a una politica di Auto AWS Scaling di Amazon SNS. CloudWatch gli allarmi vengono attivati quando il loro stato cambia e vengono mantenuti per un determinato numero di periodi, non perché si trovano in uno stato particolare. Per ulteriori informazioni, consulta [Monitoraggio SimSpace Weaver con Amazon CloudWatch](#).

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in SimSpace Weaver. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata inviata SimSpace Weaver, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta [Registrazione di chiamate API AWS SimSpace Weaver con AWS CloudTrail](#).

Convalida della conformità per AWS SimSpace Weaver

SimSpace Weaver non rientra nell'ambito di eventuali programmi di conformità di AWS.

I revisori esterni valutano la sicurezza e la conformità di altri AWS servizi nell'ambito di più programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per sapere se il Servizio AWS è coperto da programmi di conformità specifici, consulta i [Servizi AWS coperti dal programma di conformità](#) e scegli il programma di conformità desiderato. Per informazioni generali, consulta [Programmi per la conformità di AWS](#).

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La responsabilità di conformità durante l'utilizzo dei Servizi AWS è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWS mette a disposizione le seguenti risorse:

- [Guide Quick Start per la sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono la procedura per l'implementazione di ambienti di base su AWS incentrati sulla sicurezza e sulla conformità.
- [Architetture per la sicurezza e la conformità HIPAA su Amazon Web Services](#): questo whitepaper descrive come le aziende possono utilizzare AWS per creare applicazioni conformi alla normativa HIPAA.

Note

Non tutti i Servizi AWS sono conformi ai requisiti HIPAA. Per ulteriori informazioni, consulta la sezione [Riferimenti sui servizi conformi ai requisiti HIPAA](#).

- [Risorse per la conformità AWS](#): una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- [Valutazione delle risorse con le regole](#) nella Guida per gli sviluppatori di AWS Config: il servizio AWS Config valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti.
- [AWS Security Hub](#): questo Servizio AWS fornisce una visione completa dello stato di sicurezza all'interno di AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [AWS Audit Manager](#): questo Servizio AWS aiuta a verificare continuamente l'utilizzo di AWS per semplificare la gestione dei rischi e della conformità alle normative e agli standard di settore.

Resilienza nell'AWS SimSpace Weaver

L'infrastruttura globale dei servizi AWS è progettata attorno a regioni AWS e zone di disponibilità. Le regioni di Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e a velocità di trasmissione effettiva elevata. Con le zone di disponibilità, puoi progettare e gestire applicazioni e

database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle Regioni AWS e le zone di disponibilità, consulta [Infrastruttura globale di AWS](#).

Sicurezza dell'infrastruttura in AWS SimSpace Weaver

Come servizio gestito, AWS SimSpace Weaver è protetto dalla sicurezza di rete globale AWS. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina [Sicurezza del cloud AWS](#). Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina [Protezione dell'infrastruttura](#) nel Pilastro della sicurezza di AWS Well-Architected Framework.

Utilizza le chiamate API pubblicate di AWS per accedere all'AWS SimSpace Weaver tramite la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Modello di sicurezza della connettività di rete

Le tue simulazioni vengono eseguite su istanze di calcolo all'interno di un Amazon VPC situato in una AWS regione selezionata. Un Amazon VPC è una rete virtuale nel AWS cloud, che isola l'infrastruttura in base al carico di lavoro o all'entità organizzativa. Le comunicazioni tra le istanze di calcolo all'interno di Amazon VPC rimangono all'interno della AWS rete e non viaggiano su Internet. Alcune comunicazioni di servizio interno attraversano Internet e sono crittografate. Le simulazioni per tutti i clienti che operano nella stessa AWS regione condividono lo stesso Amazon VPC. Le simulazioni per clienti diversi utilizzano istanze di calcolo separate all'interno dello stesso Amazon VPC.

Comunicazioni tra i tuoi client di simulazione e le simulazioni eseguite durante i viaggi su Internet. SimSpace Weaver SimSpace Weavernon gestisce queste connessioni. È tua responsabilità proteggere le connessioni dei tuoi client.

Le connessioni al SimSpace Weaver servizio attraversano Internet e sono crittografate. Ciò include le connessioni che utilizzanoAWS Management Console, AWS Command Line Interface (AWS CLI), i kit di sviluppo AWS software (SDK) e l'SDK dell'SimSpace Weaverapp.

Analisi della configurazione e delle vulnerabilità in AWS SimSpace Weaver

La configurazione e i controlli IT sono una responsabilità condivisa tra AWS l'utente e l'utente. Per ulteriori informazioni, consulta il [modello di responsabilità AWS condivisa](#) di . AWSgestisce le attività di sicurezza di base per l'infrastruttura sottostante, come l'applicazione di patch al sistema operativo sulle istanze di calcolo, la configurazione del firewall e il disaster recovery AWS dell'infrastruttura. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta [Best practice per la sicurezza, l'identità e la conformità](#).

Sei responsabile della sicurezza del tuo software di simulazione:

- Gestisci il codice dell'app, inclusi aggiornamenti e patch di sicurezza.
- Autentica e crittografa le comunicazioni tra i tuoi client di simulazione e le app a cui si connettono.
- Aggiorna le tue simulazioni per utilizzare le versioni SDK più recenti, inclusi l'SDK e l'SDK dell'AWSapp. SimSpace Weaver

Note

SimSpace Weavernon supporta gli aggiornamenti delle app in una simulazione in esecuzione. Se devi aggiornare le app, devi interrompere ed eliminare la simulazione, quindi creare una nuova simulazione con il codice dell'app aggiornato. Ti consigliamo di salvare lo stato della simulazione in un archivio dati esterno in modo da poterlo ripristinare se devi ricreare la simulazione.

Best practice relative alla sicurezza di SimSpace Weaver

Questa sezione descrive le migliori pratiche di sicurezza specifiche per SimSpace Weaver. Per ulteriori informazioni sulle best practice di sicurezza in AWS, consulta [Best practice for Security, Identity and Compliance](#).

Argomenti

- [Crittografa le comunicazioni tra le tue app e i relativi client](#)
- [Effettua periodicamente il backup dello stato della simulazione](#)
- [Gestisci le tue app e gli SDK](#)

Crittografa le comunicazioni tra le tue app e i relativi client

SimSpace Weaver non gestisce le comunicazioni tra le tue app e i loro client. È necessario implementare una qualche forma di autenticazione e crittografia per le sessioni client.

Effettua periodicamente il backup dello stato della simulazione

SimSpace Weaver non salva lo stato della simulazione. Le simulazioni interrotte (a seguito di una chiamata API, di un'opzione di console o di un arresto anomalo del sistema) non salvano il loro stato e non hanno un modo intrinseco per ripristinarle. Le simulazioni interrotte non possono essere riavviate. L'unico modo per eseguire l'equivalente di un riavvio è ricreare la simulazione utilizzando la stessa configurazione e gli stessi dati. È possibile utilizzare i backup dello stato della simulazione per inizializzare la nuova simulazione. AWS offre servizi di [archiviazione](#) e [database](#) su cloud altamente affidabili e disponibili che è possibile utilizzare per salvare lo stato della simulazione.

Gestisci le tue app e gli SDK

Gestisci le app, le installazioni locali dei kit di sviluppo AWS software (SDK) e l'SDK dell'SimSpace Weaver app. Puoi scaricare e installare nuove versioni degli SDK. AWS Prova le nuove versioni dell'SDK dell'SimSpace Weaver app con versioni di app non di produzione per assicurarti che le tue app continuino a funzionare come previsto. Non puoi aggiornare le app in una simulazione in esecuzione. Per aggiornare le app:

1. Aggiorna e testa il codice dell'app localmente (o in un ambiente di test).
2. Smetti di cambiare lo stato della simulazione e salvalo (se necessario).
3. Interrompi la simulazione (una volta interrotta, non può essere riavviata).

4. Elimina la simulazione (le simulazioni interrotte che non sono state eliminate vengono conteggiate ai fini dei limiti del servizio).
5. Ricrea la simulazione con la stessa configurazione e il codice dell'app aggiornato.
6. Inizializza la simulazione utilizzando i dati di stato salvati (se disponibili).
7. Inizia la tua nuova simulazione.

Note

Una nuova simulazione creata con la stessa configurazione è separata dalla vecchia simulazione. Avrà un nuovo ID di simulazione e invierà i log a un nuovo flusso di log in Amazon. CloudWatch

Registrazione e monitoraggio in SimSpace Weaver

Il monitoraggio è importante per mantenere l'affidabilità, la disponibilità e le prestazioni di SimSpace Weaver e delle altre soluzioni AWS. AWS fornisce i seguenti strumenti di monitoraggio per controllare SimSpace Weaver, segnalare eventuali problemi ed eseguire operazioni automatiche quando appropriato:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai dati di log provenienti dai tuoi SimSpace Weaver dipendenti e da altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni contenute nei dati di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo Account AWS e fornisce i file di log a un bucket Simple Storage Service (Amazon S3) specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la [Guida per l'utente AWS CloudTrail](#).

Argomenti

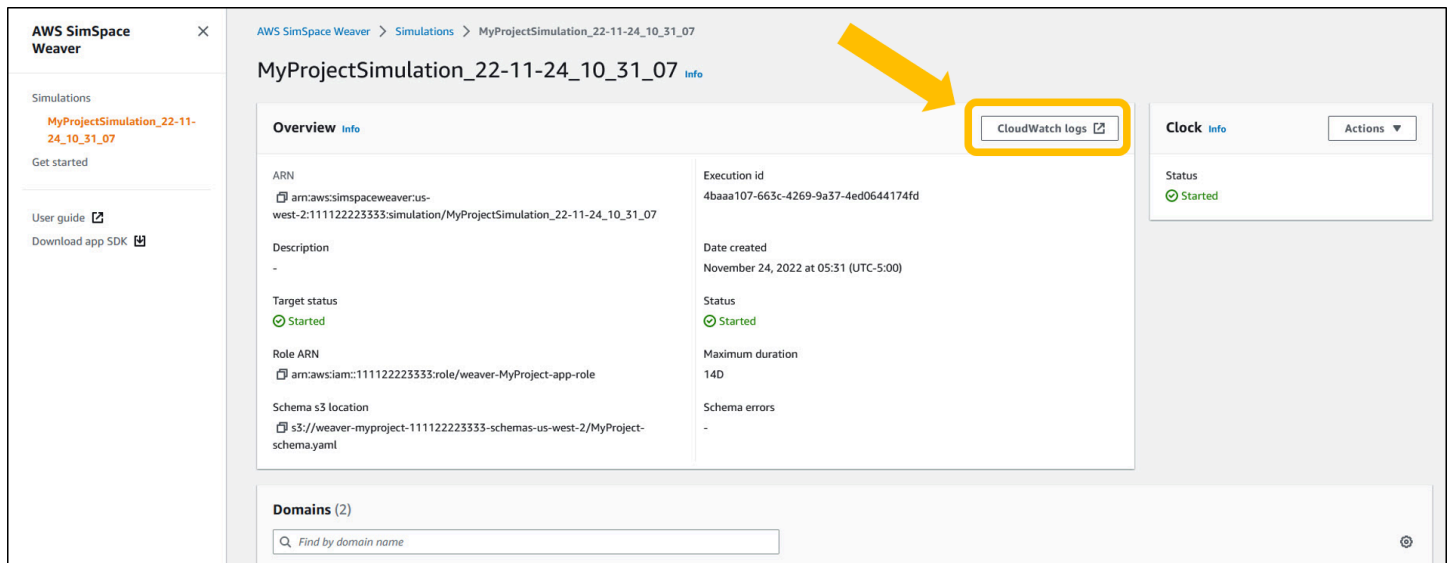
- [SimSpace Weaver accessi in Amazon CloudWatch Logs](#)
- [Monitoraggio SimSpace Weaver con Amazon CloudWatch](#)
- [Registrazione di chiamate API AWS SimSpace Weaver con AWS CloudTrail](#)

SimSpace Weaver accessi in Amazon CloudWatch Logs

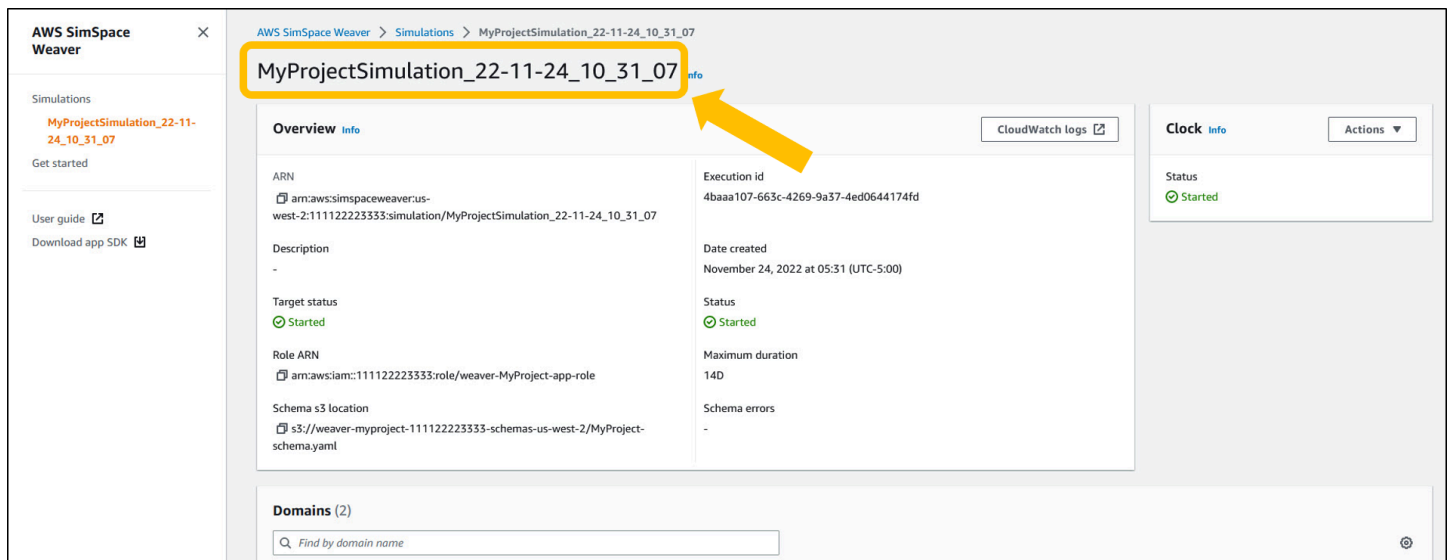
Accedere ai tuoi registri SimSpace Weaver

Tutti i log generati dalle tue SimSpace Weaver simulazioni vengono archiviati in Amazon CloudWatch Logs. Per accedere ai log, puoi utilizzare il pulsante CloudWatch Logs nel riquadro Panoramica della

simulazione nella SimSpace Weaver console, che ti porterà direttamente ai log di quella specifica simulazione.



Puoi accedere ai log anche tramite la console. CloudWatch Avrai bisogno del nome della simulazione per cercarne i log.



Log SimSpace Weaver

SimSpace Weaverscrive i messaggi di gestione della simulazione e l'output della console dalle tue app su Amazon CloudWatch Logs. Per ulteriori informazioni sull'utilizzo dei log, consulta [Working with log groups and log stream](#) nella Amazon CloudWatch Logs User Guide.

Ogni simulazione che crei ha il proprio gruppo di log in Logs. CloudWatch Il nome del gruppo di log è specificato nello schema di simulazione. Nel frammento di schema

seguito, il valore di `log_destination_service` è `logs`. Ciò significa che il valore di `log_destination_resource_name` è il nome di un gruppo di log. In questo caso, il gruppo di log è `MySimulationLogs`.

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

Puoi anche utilizzare `DescribeSimulationAPI` per trovare il nome del gruppo di log per la simulazione dopo averlo avviato.

Important

Se utilizzi AWS IAM Identity Center o nomi di profili per AWS Command Line Interface (AWS CLI), devi utilizzare la versione 1.12.1 o successiva di SimSpace Weaver dell'app SDK. La versione più recente è la 1.16.0. Per ulteriori informazioni sulle versioni di SimSpace Weaver, consulta [Versioni SimSpace Weaver](#). Gli script SDK di SimSpace Weaver dell'app utilizzano il CLI di AWS. Se utilizzi IAM Identity Center, puoi copiare il tuo profilo IAM Identity Center sul tuo profilo `default` o fornire il nome del tuo profilo IAM Identity Center per applicare gli script di SimSpace Weaver SDK con il parametro `AWS CLI --profile cli-profile-name`. Per ulteriori informazioni, consulta [Configurazione dell'uso di AWS IAM Identity Center nella Guida AWS CLI per l'utente](#) e [Impostazioni dei file di configurazione e credenziali](#) nella Guida per l'utente di AWS Command Line Interface.

Docker

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

WSL

⚠ Important

Forniamo queste istruzioni per la tua comodità. Sono destinate all'uso con Windows Subsystem for Linux (WSL) e non sono supportate. Per ulteriori informazioni, consulta [Configurazione dell'ambiente locale per SimSpace Weaver](#).

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

L'esempio seguente mostra la parte dell'output DescribeSimulation che descrive la configurazione di registrazione. Il nome del gruppo di log viene visualizzato alla fine di LogGroupArn.

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

Ogni gruppo di log di simulazione contiene diversi flussi di log:

- Flusso di log di gestione: messaggi di gestione della simulazione prodotti dal servizio. SimSpace Weaver

```
/sim/management
```

- Flusso di registro degli errori: messaggi di errore prodotti dal SimSpace Weaver servizio. Questo flusso di log esiste solo in presenza di errori. SimSpace Weaver archivia gli errori scritti dalle tue app nei rispettivi flussi di log delle app (vedi i seguenti flussi di log).

```
/sim/errors
```

- Stream di log spaziali delle app (1 per ogni app spaziale su ciascun worker): output della console prodotto dalle app spaziali. Ogni app spaziale scrive nel proprio flusso di log. Sono *spatial-app-id* tutti i caratteri dopo la barra finale alla fine di *worker-id*

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

- Stream di log delle app personalizzati (1 per ogni istanza di app personalizzata): output della console prodotto da app personalizzate. Ogni istanza dell'app personalizzata scrive nel proprio flusso di log.

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

- Stream di log delle app di servizio (1 per ogni istanza dell'app di servizio): output della console prodotto dalle app di servizio. Ogni app di servizio scrive nel proprio flusso di log. Sono *service-app-id* tutti i caratteri dopo la barra finale alla fine di *service-app-name*

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

Monitoraggio SimSpace Weaver con Amazon CloudWatch

Puoi monitorare SimSpace Weaver utilizzando Amazon CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione Web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

Il SimSpace Weaver servizio riporta le seguenti metriche nel `AWS/simspaceweaver` namespace.

SimSpace Weaver metriche a livello di account

Il SimSpace Weaver namespace include le seguenti metriche relative all'attività a livello di account.

AWS

Parametro	Descrizione
SimulationCount	Il numero di simulazioni per l'account corrente. Unità: numero Dimensioni: nessuna Statistiche: media, minima, massima

Registrazione di chiamate API AWS SimSpace Weaver con AWS CloudTrail

AWS SimSpace Weaver è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, da un ruolo o da un AWS servizio in SimSpace Weaver. CloudTrail acquisisce tutte le chiamate API SimSpace Weaver come eventi. Le chiamate acquisite includono le chiamate dalla console di SimSpace Weaver e le chiamate di codice alle operazioni delle API SimSpace Weaver. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per SimSpace Weaver. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console in Event history. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata inviata SimSpace Weaver, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la [Guida AWS CloudTrail per l'utente](#).

SimSpace Weaver informazioni in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività in SimSpace Weaver, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio in Event history. Puoi visualizzare, cercare e scaricare gli eventi recenti in Account AWS. Per ulteriori informazioni, vedere [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nell'Account AWS che includa gli eventi per SimSpace Weaver, crea un trail. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le

Regioni AWS. Il percorso registra gli eventi di tutte le Regioni nella partizione AWSe distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Tutte SimSpace Weaver le azioni vengono registrate CloudTrail e documentate nell'[AWS SimSpace WeaverAPI Reference](#). Ad esempio, le chiamate a `DescribeSimulation` e `ListSimulations` le `DeleteSimulation` azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Comprensione delle voci dei file di log di SimSpace Weaver

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, come la data e l'ora dell'azione, i parametri della richiesta e altri dettagli. CloudTrail i file di registro non sono una traccia ordinata delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`ListSimulations` azione.

```
{
  "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 Endpoint e quote Weaver

Le tabelle seguenti descrivono gli endpoint e le quote di servizio per Weaver. SimSpace Le service quotas, a cui si fa riferimento anche come limiti, rappresentano il numero massimo di risorse di servizio o operazioni per l'Account AWS. Per ulteriori informazioni, consulta [AWS Service Quotas](#) in Riferimenti generali di AWS.

Endpoint del servizio

Nome regione	Regione	Endpoint	Protocollo
US East (N. Virginia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
Stati Uniti orientali (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europa (Stoccolma)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nome regione	Regione	Endpoint	Protocollo
Europe (Frankfurt)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Quote del servizio

Nome	Predefinita	Adattata	Descrizione
Unità di risorse di calcolo per ogni app	Ogni regione supportata: 4	No	Il numero massimo di unità di risorse di calcolo che puoi allocare a ciascuna app.
Unità di risorse di calcolo per ogni worker	Ogni regione supportata: 17	No	Il numero di unità di risorse di calcolo disponibili per ogni worker.
Campi di dati per ciascuna entità	Ogni regione supportata: 7	No	Il numero massimo di campi di dati (non indice) che può avere un'entità.

Nome	Predefinita	Adattate	Descrizione
Entità in una partizione	Ogni regione supportata: 8.192	No	Il numero massimo di entità in 1 partizione.
Dimensione del campo di dati dell'entità	Ogni regione supportata: 1.024 byte	No	La dimensione massima di un campo di dati (non indice) di un'entità.
Trasferimenti di entità tra worker	Ogni regione supportata: 25	No	Il numero massimo di trasferimenti di entità tra worker, per ogni partizione e ogni tick.
Trasferimenti di entità sullo stesso worker	Ogni regione supportata: 500	No	Il numero massimo di trasferimenti di entità sullo stesso worker, per ogni partizione e ogni tick.
Campi di indice per ogni entità	Ogni regione supportata: 1	No	Il numero massimo di campi di indice che può avere un'entità.
Durata massima (in giorni) per una simulazione	Ogni regione supportata: 14	No	Il numero massimo di giorni che è possibile specificare come durata massima per una simulazione. Tutte le simulazioni hanno una durata massima, anche se non si specifica il valore. Una simulazione si interrompe automaticamente quando raggiunge la sua durata massima.

Nome	Predefinita	Adattate	Descrizione
Memoria per ogni unità di risorse di calcolo	Ogni regione supportata: 1 GB	No	La quantità di memoria ad accesso casuale (RAM) che un'app ottiene per ogni unità di risorsa di calcolo.
Sottoscrizioni remote per ogni worker	Ogni regione supportata: 24	No	Il numero massimo di sottoscrizioni remote per ogni worker.
Numero di simulazione	Ogni regione supportata: 2	Sì	Il numero massimo di simulazioni con lo stato obiettivo di STARTED nel tuo account. Puoi richiedere un aumento della quota fino a 10.
Worker per una simulazione	Ogni regione supportata: 2	Sì	Il numero massimo di lavoratori che è possibile assegnare a 1 simulazione. Puoi richiedere un aumento della quota fino a 10.
vCPU per ogni unità di risorse di calcolo	Ogni regione supportata: 2	No	Il numero di unità di elaborazione centrale virtuali (vCPU) che un'app riceve per ogni unità di risorse di calcolo.

Quote di messaggistica

Le seguenti quote si applicano alla messaggistica da app a app per SimSpace Weaver Local e in Cloud AWS

Nome	Predefinita	Adattabile	Descrizione
Dimensione massima del messaggio	Ogni regione supportata: 256 byte	No	La dimensione massima del payload di un messaggio.
Velocità massima di invio dei messaggi	Ogni regione supportata: 128	No	Il numero massimo di messaggi che ogni app può inviare per segno di spunta.

Frequenze di clock

Lo schema di simulazione specifica la frequenza di clock (chiamata anche frequenza di ticking) per una simulazione. La tabella seguente specifica le frequenze di clock valide che è possibile utilizzare.

Nome	Valori validi	Descrizione
Frequenza di clock	Ogni regione supportata: «10", «15", «30", «illimitato»	Le frequenze di clock valide per una simulazione.
Frequenza di clock (versioni 1.13 e 1.12)	Ogni regione supportata: 10, 15, 30	Le frequenze di clock valide per una simulazione.

Service Quotas per SimSpace Weaver Local

Le seguenti Service Quotas si applicano solo a SimSpace Weaver Local. Tutte le altre quote si applicano anche a SimSpace Weaver Local.

Nome	Predefinita	Adattabile	Descrizione
Numero massimo di partizioni	SimSpace Weaver Local: 24	No	Il numero massimo di partizioni per una simulazione.

Nome	Predefinita	Adattabile	Descrizione
Numero massimo di app	SimSpace Weaver Local: 24	No	Il numero totale massimo di app (di qualsiasi tipo) per una simulazione.
Numero massimo di domini	SimSpace Weaver Local: 24	No	Il numero totale massimo di domini (di qualsiasi tipo) per una simulazione.
Entità per partizione	SimSpace Weaver Local: 4.096	No	Il numero massimo di entità in ogni partizione.
Campi per entità	SimSpace Weaver Local: 8	No	Il numero massimo di campi per ogni entità.
Dimensione dei campi	SimSpace Weaver Local: 1024 byte	No	La dimensione massima di un campo di entità.

Risoluzione dei problemi in SimSpace Weaver

Argomenti

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [Autorizzazione negata durante l'avvio della simulazione](#)
- [Problemi legati al tempo di utilizzo Docker](#)
- [PathfindingSample il client della console non riesce a connettersi](#)
- [Il AWS CLI non riconosce simspaceweaver](#)

AssumeRoleAccessDenied

Potresti ricevere il seguente errore se la simulazione non si avvia:

```
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
```

Puoi ricevere questo errore se si verifica una delle seguenti condizioni relative al ruolo AWS Identity and Access Management (IAM) per la tua simulazione:

- L'Amazon Resource Name (ARN) si riferisce a un ruolo IAM che non esiste.
- La politica di fiducia per il ruolo IAM che non consente al nome della nuova simulazione di assumere il ruolo.

Verifica che il ruolo esista. Se il ruolo esiste, controlla la tua politica di fiducia per il ruolo.

L'aws:SourceArnesempio seguente, la politica di fiducia consente di assumere il ruolo solo con una simulazione (nell'account 111122223333) il cui nome inizia conMySimulation.

```
{
  "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*"
        }
      }
    }
  ]
}

```

Per consentire a un'altra simulazione il cui nome inizia con `MyOtherSimulation` di assumere il ruolo, la politica di fiducia deve essere modificata come nel seguente esempio modificato:

```

{
  "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

Potresti ricevere il seguente errore durante la creazione di un progetto:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

Hai ricevuto questo errore perché il nome SimSpace Weaver passato ad Amazon Simple Storage Service (Amazon S3) violava le regole di denominazione dei bucket (per ulteriori informazioni, [consulta le regole di denominazione dei bucket nella Amazon Simple Storage Service User Guide](#)).

Lo `create-project` script nell'SDK dell'SimSpace Weaverapp crea i nomi dei bucket utilizzando il nome del progetto che fornisci allo script. I nomi dei bucket utilizzano i seguenti formati:

- Versione 1.13.x o successiva
 - `weaver-lowercase-project-name-account-number-region`
- Versione 1.12.x
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

Ad esempio, date le seguenti proprietà del progetto:

- Nome del progetto: MyProject
- Account AWSnumero: 111122223333
- Regione AWS: us-west-2

Il progetto avrebbe i seguenti bucket:

- Versione 1.13.x o successiva
 - `weaver-myproject-111122223333-us-west-2`
- Versione 1.12.x
 - `weaver-myproject-111122223333-app-zips-us-west-2`
 - `weaver-myproject-111122223333-schemas-us-west-2`

Il nome del progetto non deve violare le regole di denominazione di Amazon S3. È inoltre necessario utilizzare un nome di progetto sufficientemente breve in modo che i nomi dei bucket creati dallo `create-project` script non superino il limite di lunghezza dei nomi per i bucket Amazon S3.

ServiceQuotaExceededException

Potresti ricevere il seguente errore quando avvii una simulazione:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

Riceverai questo errore se tenti di avviare una nuova simulazione ma il tuo account ha attualmente il numero massimo di simulazioni con lo stato obiettivo di `STARTED`. Ciò include l'esecuzione di simulazioni, le simulazioni fallite e le simulazioni che si sono interrotte perché hanno raggiunto la durata massima. È possibile eliminare una simulazione interrotta o non riuscita per consentire l'avvio di una nuova simulazione. Se tutte le simulazioni sono in esecuzione, puoi interrompere ed eliminare una simulazione in esecuzione. Puoi anche richiedere un aumento delle quote di servizio se non hai già raggiunto il limite di richieste. Per ulteriori informazioni, consulta [SimSpace Endpoint e quote Weaver](#). Per informazioni su come eliminare le simulazioni non necessarie, consulta il [Passaggio 6: interrompi e ripulisci la simulazione](#) tutorial di avvio rapido.

TooManyBuckets

Potresti ricevere il seguente errore durante la creazione di un progetto:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) prevede un limite al numero di bucket che puoi avere nel AWS tuo account (per ulteriori informazioni, [consulta Restrizioni e limitazioni dei bucket](#) nella Guida per l'utente di Amazon Simple Storage Service).

Prima di continuare, devi eseguire una delle seguenti operazioni:

- Elimina 2 o più bucket Amazon S3 esistenti che non ti servono.
- Richiedi un aumento del limite di Amazon S3 (per ulteriori informazioni, consulta le [restrizioni e le limitazioni dei bucket](#) nella Guida per l'utente di Amazon Simple Storage Service).

- Usa un account diverso. AWS

Note

L'`DeleteSimulationAPI` in SimSpace Weaver non elimina le risorse Amazon S3 associate alla simulazione. Ti consigliamo di rimuovere tutte le risorse associate alle simulazioni quando non ne hai più bisogno. Per indicazioni su come ripulire la simulazione, consulta il tutorial [Passaggio 6: interrompi e ripulisci la simulazione](#) di avvio rapido.

Autorizzazione negata durante l'avvio della simulazione

Quando avvii una simulazione, potresti ricevere un messaggio di errore che indica che l'autorizzazione è stata negata o che si è verificato un errore durante l'accesso agli artefatti dell'app. Questo problema può verificarsi quando specifichi bucket Amazon S3 per la simulazione che SimSpace Weaver non sono stati creati per te (tramite la console o gli script SDK dell'`SimSpace Weaverapp`).

Le seguenti situazioni sono le cause principali più probabili:

- Il servizio non dispone dell'autorizzazione per accedere a uno o più bucket Amazon S3 che hai specificato nello schema di simulazione: controlla la politica di autorizzazione dei ruoli dell'app, le policy dei bucket Amazon S3 e le autorizzazioni dei bucket Amazon S3 per assicurarti che disponga delle autorizzazioni corrette per accedere ai tuoi bucket. `simspaceweaver.amazonaws.com` Per ulteriori informazioni sulla politica di autorizzazione dei ruoli dell'app, consulta [Autorizzazioni create per te SimSpace Weaver](#)
- Il bucket Amazon S3 potrebbe trovarsi in una situazione Regione AWS diversa dalla simulazione: i bucket Amazon S3 per gli artefatti di simulazione devono trovarsi nella stessa posizione della simulazione. Regione AWS Controlla la tua console Amazon S3 per vedere in cosa consiste Regione AWS il tuo bucket. Se il tuo bucket Amazon S3 si trova in un altro bucket Regione AWS, seleziona un bucket che si trovi nello stesso Regione AWS della simulazione.

Problemi legati al tempo di utilizzo Docker

Se utilizzi Docker e ricevi errori relativi all'ora durante l'esecuzione di script dall'`SDK` dell'`SimSpace Weaverapp`, la causa potrebbe essere che l'orologio della tua macchina Docker virtuale non è

corretto. Questo può accadere se il computer era in funzione Docker e poi riprende dalla modalità di sospensione o ibernazione.

Soluzioni da provare

- Riavviare Docker.
- Disabilita e riattiva la sincronizzazione dell'ora in: Windows PowerShell

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample il client della console non riesce a connettersi

Potresti ricevere il seguente errore dal client della console quando ti connetti alla PathfindingSample simulazione descritta nel tutorial di [avvio rapido e nel tutorial dettagliato](#). Questo errore si verifica perché il client non può aprire una connessione di rete ViewApp all'indirizzo IP e al numero di porta combinati che hai fornito.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Per una simulazione in Cloud AWS

- La tua connessione di rete funziona correttamente? Verifica di poterti connettere ad altri indirizzi IP o siti Web che dovrebbero funzionare. Assicurati che il tuo browser web non stia caricando un sito web dalla sua cache.
- La simulazione è in esecuzione? Puoi utilizzare l'ListSimulationsAPI per conoscere lo stato della simulazione. Per ulteriori informazioni, consulta [Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#). Puoi anche usare la [SimSpace Weaverconsole](#) per controllare lo stato delle tue simulazioni.
- Le tue app sono in esecuzione? Puoi utilizzare l'DescribeAppAPI per conoscere lo stato delle tue app. Per ulteriori informazioni, consulta [Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#). Puoi anche utilizzare la [SimSpace Weaverconsole](#) per controllare lo stato delle tue simulazioni.
- Le tue app sono in esecuzione? Puoi utilizzare l'DescribeAppAPI per conoscere lo stato delle tue app. Per ulteriori informazioni, consulta [Passaggio 4: Ottieni il tuo indirizzo IP e il numero di porta](#). Puoi anche utilizzare la [SimSpace Weaverconsole](#) per controllare lo stato delle tue simulazioni.

- Hai usato l'indirizzo IP e il numero di porta corretti? Quando ti connetti tramite Internet, devi utilizzare l'indirizzo IP e il numero di Actual porta di ViewApp. Puoi trovare l'IP Address e il numero di Actual porta nel EndpointInfo blocco dell'output dell'DescribeAppAPI. Puoi anche utilizzare la [SimSpace Weaverconsole](#) per trovare l'indirizzo IP (URI) e il numero di porta (porta di ingresso) ViewApp nella pagina dei MyViewDomain dettagli.
- La tua connessione di rete attraversa un firewall? Il firewall potrebbe bloccare la connessione all'indirizzo IP o al numero di porta (o a entrambi). Controlla le impostazioni del firewall o rivolgiti all'amministratore del firewall.

Per una simulazione locale

- Puoi connetterti al tuo indirizzo di loopback (127.0.0.1)? Se disponi dello strumento da riga di ping comando in Windows, puoi aprire una finestra del prompt dei comandi e provare a eseguire il ping di 127.0.0.1. Premi Ctrl - C per terminare il ping.

```
ping 127.0.0.1
```

Example uscita ping

```
C:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
Control-C
^C
C:\>
```

Se il ping indica la perdita di pacchetti, è possibile che altri software (ad esempio un firewall locale, impostazioni di sicurezza o programmi antimalware) stiano bloccando la connessione.

- Le tue app sono in esecuzione? La simulazione locale viene eseguita come finestre separate per ogni app. Assicurati che le finestre delle tue app spaziali ViewApp siano aperte. Per ulteriori informazioni, consulta [Sviluppo locale](#).
- Hai usato l'indirizzo IP e il numero di porta corretti? È necessario utilizzare `tcp://127.0.0.1:7000` quando ci si connette a una simulazione locale. Per ulteriori informazioni, consulta [Sviluppo locale](#).
- Disponi di un software di sicurezza locale che potrebbe bloccare la tua connessione? Controlla le impostazioni di sicurezza, il firewall locale o i programmi antimalware per vedere se bloccano la connessione alla porta `127.0.0.1 TCP. 7000`

Il AWS CLI non riconosce **simspaceweaver**

Se ti AWS CLI dà errori che suggeriscono che non sia a conoscenza SimSpace Weaver, esegui il comando seguente.

```
aws simspaceweaver help
```

Se ricevi un errore che inizia con le righe seguenti ed elenca tutte le scelte disponibili, è AWS CLI possibile che si tratti di una versione precedente.

```
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:
```

Esegui il comando seguente per verificare la versione del tuo AWS CLI.

```
aws --version
```

Se il numero di versione è precedente alla 2.9.19, devi aggiornare la tua AWS CLI. Tieni presente che la versione corrente di AWS CLI è successiva alla 2.9.19.

Per aggiornare il tuo AWS CLI, consulta [Installare o aggiornare la versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface](#) della versione 2.

SimSpace Weaveriferimento allo schema di simulazione

SimSpace Weaverutilizza un file YAML per configurare le proprietà di una simulazione. Questo file è chiamato schema di simulazione (o semplicemente schema). La simulazione di esempio inclusa nell'SDK dell'SimSpace Weaverapp include uno schema che puoi copiare e modificare per la tua simulazione.

Argomenti

- [Esempi di uno schema completo](#)
- [Formato dello schema](#)

Esempi di uno schema completo

L'esempio seguente mostra il file di testo in YAML formato -format che descrive una SimSpace Weaver simulazione. Questi esempi includono valori fittizi per le proprietà. Il formato del file varia in base al valore `sdk_version` specificato in esso. [Formato dello schema](#)Per una descrizione completa delle proprietà e dei relativi valori validi, vedere.

```
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"]
```

Formato dello schema

L'esempio seguente mostra la struttura generale di uno schema. L'ordine delle proprietà a ciascun livello dello schema non ha importanza, purché le relazioni padre-figlio siano le stesse. L'ordine è importante per gli elementi di un array.

```
sdk_version: "sdk-version-number"
simulation_properties:
  simulation-properties
workers:
  worker-group-configurations
clock:
  tick_rate: tick-rate
partitioning_strategies:
  partitioning-strategy-configurations
domains:
  domain-configurations
placement_constraints:
  placement-constraints-configuration
```

Sections

- [Versione SDK](#)
- [Proprietà di simulazione](#)
- [Worker](#)
- [Orologio](#)
- [Strategie di partizionamento](#)
- [Domini](#)
- [Vincoli di posizionamento](#)

Versione SDK

La `sdk_version` sezione (obbligatoria) identifica la versione dell'SDK dell'SimSpace Weaverapp che supporta questo schema. Valori validi: 1.16, 1.15, 1.14, 1.13, 1.12

Important

Il valore di `include sdk_version` solo il numero della versione principale e il numero della prima versione secondaria. Ad esempio, il valore 1.12 specifica tutte le versioni `1.12.x`, ad esempio `1.12.01.12.1`, e `1.12.2`.

```
sdk_version: "1.16"
```

Proprietà di simulazione

La `simulation_properties` sezione (obbligatoria) specifica varie proprietà della simulazione. Utilizzate questa sezione per configurare la registrazione e specificare un'immagine del contenitore predefinita. Questa sezione è obbligatoria anche se non configuri la registrazione o scegli di specificare un'immagine del contenitore predefinita.

```
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"
```

Proprietà

`log_destination_resource_name`


Specifica la risorsa su cui SimSpace Weaver verranno scritti i log.

Obbligatorio: No. Se questa proprietà non è inclusa, SimSpace Weaver non scriverà i log per la simulazione.

Tipo: stringa

Valori validi:

- Il nome di un gruppo di log di CloudWatch Logs (ad esempio,) `MySimulationLogs`
- L'Amazon Resource Name (ARN) di un gruppo di log CloudWatch Logs (ad esempio,) `arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`

 Note

SimSpace Weaversupporta solo una destinazione di log nello stesso account e Regione AWS nella simulazione.

`log_destination_service`

Indica il tipo di risorsa di destinazione per la registrazione quando si specifica una risorsa `logging_destination_resource_name` che non è un ARN.

Obbligatorio: è necessario specificare questa proprietà se `log_destination_resource_name` è specificato e non è un ARN. Non è possibile specificare questa proprietà se `log_destination_resource_name` non è specificata o è un ARN.

Tipo: stringa

Valori validi:

- `logs`: La risorsa di destinazione del registro è un gruppo di log.

`default_entity_index_key_type`

Speciifica il tipo di dati per il campo chiave dell'indice delle entità di simulazione.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: `Vector3<f32>`

`default_image`

Specifica l'immagine del contenitore di default per la simulazione (non supportata per la versione e). 1.13 1.12 Se viene specificata questa proprietà, i domini che non lo specificano `image` utilizzano. `default_image`

Required: No

Tipo: stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Worker

La `workers` sezione (obbligatoria) specifica le configurazioni per i gruppi di lavoro (gruppi di lavoratori). SimSpace Weaver utilizza queste informazioni insieme `placement_constraints` per configurare l'infrastruttura sottostante della simulazione. Attualmente è supportato solo 1 gruppo di lavoro.

Per specificare le proprietà di un gruppo di lavoro, *worker-group-name* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificate le proprietà del gruppo di lavoro dopo il nome.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Proprietà

type

Speciifica il tipo di lavoratore.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: `sim.c5.24xlarge`

desired

Speciifica il numero di lavoratori desiderato per questo gruppo di lavoratori.

Campo obbligatorio: sì

Tipo: integer

Valori validi: 1 -3. La quota di servizio (limite) per il numero di lavoratori per le simulazioni determina il valore massimo di questa proprietà. Ad esempio, se la quota di servizio è2, il valore massimo per questa proprietà è2. È possibile richiedere un aumento della quota di servizio. Per ulteriori informazioni, consulta [SimSpace Endpoint e quote Weaver](#).

Orologio

La `clock` sezione (obbligatoria) specifica le proprietà dell'orologio di simulazione.

```
clock:  
  tick_rate: tick-rate
```

Proprietà

`tick_rate`

Specifica il numero di segni di spunta al secondo che l'orologio pubblica nelle app.

Campo obbligatorio: sì

Type:

- Versione *1.14* e: stringa *1.15*
- Versione *1.13* e *1.12*: numero intero

Valori validi:

- Versione *1.14* e *1.15*: "10" | "15" | "30" "unlimited"
 - "unlimited": l'orologio invia il segno di spunta successivo non appena tutte le app terminano le operazioni di commit per il segno di spunta corrente.
- Versione *1.13* e *1.12*: | 10 15 30

Strategie di partizionamento

La `partitioning_strategies` sezione (obbligatoria) specifica l'organizzazione delle partizioni per un dominio spaziale.

Note

SimSpace Weaver supporta solo 1 strategia di partizionamento.

Per specificare le proprietà di una strategia di partizionamento, sostituisci la *partitioning-strategy-name* con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificare le proprietà della strategia di partizionamento dopo il nome.

```
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
```

Proprietà**topology**

Specifica la topologia (schema di disposizione delle partizioni) per questa strategia di partizionamento.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: "Grid"

aabb_bounds

Specificate i limiti del riquadro di delimitazione allineato all'asse principale per la simulazione. (AABB) I limiti vengono specificati come matrici ordinate a 2 elementi che descrivono i valori minimo e massimo (in quest'ordine) per ciascun asse (e). x y

Obbligatorio: condizionale. Questa proprietà è obbligatoria (e può essere specificata) solo se la topologia è impostata su. "Grid"

Tipo: Float array (per ogni asse)

Valori validi: $-3.4028235e38$ - $3.4028235e38$

grid_placement_groups

Specificate il numero di gruppi di posizionamento lungo ciascun asse (x e y) in una topologia a griglia. Un gruppo di posizionamento è una raccolta di partizioni (nello stesso dominio) spazialmente adiacenti.

Obbligatorio: condizionale. Questa proprietà è obbligatoria (e può essere specificata) solo se la topologia è impostata su "Grid". Se non specificate una configurazione dei gruppi di posizionamento, ne SimSpace Weaver calcolerà una per voi. Qualsiasi dominio che utilizza una strategia di partizionamento senza una configurazione dei gruppi di posizionamento deve specificare un `grid_partition` (vedi [Strategia di partizionamento del dominio spaziale](#)).

Tipo: numero intero (per ogni asse)

Valori validi: 1 -20. Si consiglia che $x * y$ sia uguale al numero di lavoratori desiderato. Altrimenti, SimSpace Weaver cercherà di bilanciare i gruppi di collocamento tra i lavoratori disponibili.

Domini

La `domains` sezione (obbligatoria) specifica le proprietà per ciascuno dei tuoi domini. Tutte le simulazioni devono avere almeno una sezione per un dominio spaziale. È possibile creare più sezioni per domini aggiuntivi. Ogni tipo di dominio ha il proprio formato di configurazione.

Important

Versioni 1.13 e 1.12 non supportano più domini spaziali.

Important

SimSpace Weaversupporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

```
domains:
```

```
  domain-name:
```

```

domain-configuration
domain-name:
  domain-configuration
  ...

```

Configurazione del dominio

- [Configurazione del dominio spaziale](#)
- [Configurazione personalizzata del dominio](#)
- [Configurazione del dominio di servizio](#)

Configurazione del dominio spaziale

Per specificare le proprietà di un dominio spaziale, sostituiscilo *spatial-domain-name* con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificate le proprietà del dominio spaziale dopo il nome.

```

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"

```

Strategia di partizionamento del dominio spaziale

La `launch_apps_by_partitioning_strategy` sezione (obbligatoria) specifica la strategia di partizionamento e le dimensioni (in numero di partizioni) dello spazio di simulazione.

```

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

```

Proprietà

`partitioning_strategy`

Specifica la strategia di partizionamento per questo dominio spaziale.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: il valore di questa proprietà deve corrispondere al nome di una strategia di partizionamento definita nella sezione `partitioning_strategies`. Per ulteriori informazioni, consulta [Strategie di partizionamento](#).

`grid_partition`

Specifica il numero di partizioni lungo ciascun asse (x e y) in una topologia a griglia. Queste dimensioni descrivono lo spazio totale di simulazione per questo dominio.

Obbligatorio: condizionale. Questa proprietà può essere specificata solo se la topologia è impostata su "Grid". Questa proprietà dipende dalla `grid_placement_groups` proprietà della strategia di partizionamento specificata per questo dominio:

- Questa proprietà è obbligatoria se la strategia di partizionamento di questo dominio non specifica una configurazione `grid_placement_groups`.
- Se esiste una `grid_placement_groups` configurazione ma non viene specificata `grid_partition`, SimSpace Weaver utilizzerà le stesse dimensioni della `grid_placement_groups` configurazione.
- Se si specificano entrambe le dimensioni `grid_placement_groups` e `grid_partition`, le dimensioni di `grid_partition` devono essere multipli di quelle di `grid_placement_groups` (ad esempio, se le dimensioni sono 2x2, alcune `grid_placement_groups` dimensioni valide per `grid_partition` sono 2x2, 4x4, 6x6, 8x8, 10x10).

Tipo: numero intero (per ogni asse)

Valori validi: 1 - 20

Configurazione spaziale dell'app

La `app_config` sezione (obbligatoria) specifica il pacchetto, la configurazione di avvio e i requisiti di risorse per le app in questo dominio.


```
app_config:  
  package: "app-package-s3-uri"  
  launch_command: ["app-launch-command", "parameter1", ...]  
  required_resource_units:  
    compute: app-resource-units
```

Proprietà

package

Specifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio, `s3://example-bucket/MySpatialApp.zip`.

launch_command

Specifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

required_resource_units

Specifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale su un lavoratore. (RAM) Per ulteriori informazioni sulle unità di risorse, vedere [Endpoint e quote di servizio](#). La compute proprietà specifica un'allocazione di unità di risorse per la compute famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - 1 4

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni 1.13 e 1.12). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la `default_image`. Per ulteriori informazioni, consulta [Contenitori personalizzati](#).

```
image: "ecr-repository-uri"
```

Proprietà

`image`

Specifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configurazione personalizzata del dominio

Per specificare le proprietà di un dominio personalizzato, *custom-domain-name* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ (trattino). Specificate le proprietà del dominio personalizzato dopo il nome. Ripeti questa procedura per ogni dominio personalizzato.

```
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"
```

Proprietà

launch_apps_via_start_app_call

Questa proprietà è necessaria per avviare le app personalizzate utilizzando l'StartAppAPI.

Campo obbligatorio: sì

Tipo: N/A

Valori validi: {}

Configurazione personalizzata dell'app

`app_config` section(obbligatorio) specifica il pacchetto, la configurazione di avvio, i requisiti di risorse e le porte di rete per le app in questo dominio personalizzato.

```
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, ...]
```

Proprietà

package

Specifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio, `s3://example-bucket/MyCustomApp.zip`.

`launch_command`

Specifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

`required_resource_units`

Specifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale su un lavoratore. (RAM) Per ulteriori informazioni sulle unità di risorse, vedere [Endpoint e quote di servizio](#). La compute proprietà specifica un'allocazione di unità di risorse per la compute famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - 1 4

`endpoint_config`

Specifica gli endpoint di rete per le app in questo dominio. Il valore di `ingress_ports` specifica le porte a cui si collegano le app personalizzate per le connessioni client in entrata. SimSpace Weaver mappa le porte allocate dinamicamente alle porte di ingresso specificate. Le porte di ingresso sono sia TCP che UDP. Usa l'DescribeAppAPI per trovare il numero di porta effettivo per connettere i tuoi client.

Obbligatorio: No. Se non specifichi la configurazione degli endpoint, le app personalizzate in questo dominio non avranno endpoint di rete.

Tipo: matrice intera

Valori validi: 1024 -49152. I valori devono essere univoci.

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni 1.13

e1.12). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la `default_image`. Per ulteriori informazioni, consulta [Contenitori personalizzati](#).

```
image: "ecr-repository-uri"
```

Proprietà

image

Specifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configurazione del dominio di servizio

Per specificare le proprietà di un dominio di servizio, *service-domain-name* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ (trattino). Specificare le proprietà del dominio del servizio dopo il nome. Ripetere questa procedura per ogni dominio di servizio.

```
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"
```

Avvia app per lavoratore

La `launch_apps_per_worker` sezione (obbligatoria) indica che si tratta di una configurazione del dominio di servizio e specifica il numero di app di servizio da avviare per lavoratore.

```
launch_apps_per_worker:  
  count: number-of-apps-to-launch
```

Proprietà

count

Questa proprietà specifica il numero di app di servizio da avviare per lavoratore.

Campo obbligatorio: sì

Tipo: integer

Valori validi: {} | 1 | 2. Un valore di {} specifica il valore predefinito di 1.

Configurazione dell'app di servizio

La `app_config` section (obbligatoria) specifica il pacchetto, la configurazione di avvio, i requisiti di risorse e le porte di rete per le app in questo dominio di servizio.

```
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, ...]
```

Proprietà

package

Specifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio, `s3://example-bucket/MyServiceApp.zip`.

`launch_command`

Specifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

`required_resource_units`

Specifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale su un lavoratore. (RAM) Per ulteriori informazioni sulle unità di risorse, vedere [Endpoint e quote di servizio](#). La compute proprietà specifica un'allocazione di unità di risorse per la compute famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - 1 4

`endpoint_config`

Specifica gli endpoint di rete per le app in questo dominio. Il valore di `ingress_ports` specifica le porte a cui si collegano le app di servizio per le connessioni client in entrata. SimSpace Weaver mappa le porte allocate dinamicamente alle porte di ingresso specificate. Le porte di ingresso sono sia TCP che UDP. Usa l'DescribeAppAPI per trovare il numero di porta effettivo per connettere i tuoi client.

Obbligatorio: No. Se non specifichi la configurazione degli endpoint, le app di servizio in questo dominio non avranno endpoint di rete.

Tipo: matrice intera

Valori validi: 1024 -49152. I valori devono essere univoci.

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni 1.13 e 1.12). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la `default_image`. Per ulteriori informazioni, consulta [Contenitori personalizzati](#).

```
image: "ecr-repository-uri"
```

Proprietà

image

Specifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Vincoli di posizionamento

La `placement_constraints` sezione (opzionale) specifica quali domini spaziali SimSpace Weaver devono essere collocati insieme sugli stessi worker. Per ulteriori informazioni, consulta [Configurazione dei domini spaziali](#).

Important

Versioni 1.13 e 1.12 non supportate. `placement_constraints`

```
placement_constraints:  
- placed_together: ["spatial-domain-name", "spatial-domain-name", ...]
```



```
on_workers: ["worker-group-name"]
```

Proprietà

placed_together

Specifica i domini spaziali da unire SimSpace Weaver.

Campo obbligatorio: sì

Tipo: matrice di stringhe

Valori validi: nomi dei domini spaziali specificati nello schema

on_workers

Specifica il gruppo di lavoro su cui SimSpace Weaver collocare i domini.

Campo obbligatorio: sì

Tipo: matrice di stringhe a 1 elemento

Valori validi: nome di un gruppo di lavoro specificato nello schema

SimSpace Weaver Riferimenti API

SimSpace Weaver dispone di 2 diversi set di interfacce di programmazione delle applicazioni (API):

- API di servizio: queste API controllano il servizio e le risorse del servizio, come simulazioni, orologi e app. Fanno parte del kit di sviluppo AWS software principale (SDK) e puoi utilizzare l'interfaccia a riga di AWS comando (CLI) per chiamarli. Puoi anche usare lo script di convenienza nella cartella tools del tuo progetto e della tua piattaforma (ad esempio, *project-folder*\tools\windows\weaver-*project-name* -cli.bat) per chiamare queste API. [Per ulteriori informazioni sulle API di servizio, consulta il riferimento alle API. SimSpace Weaver](#)
- API SDK dell'app: queste API controllano i dati all'interno della simulazione. Le usi nel codice dell'app per eseguire operazioni come leggere e scrivere i dati dei campi delle entità, lavorare con gli abbonamenti e monitorare gli eventi nella simulazione. Per ulteriori informazioni, consulta la documentazione dell'SDK SimSpace Weaver dell'app nella cartella SDK dell'app decompressa: *sdk-folder*\SimSpaceWeaverAppSdk-1.16.0\documentation

Note

sdk-folder è la cartella in cui hai decompresso il pacchetto. SimSpaceWeaverAppSdkDistributable Se *sdk-folder*\SimSpaceWeaverAppSdk-1.16.0 non esiste, assicurati di avere la versione 1.16.0 dell'app SDK. SimSpace Weaver SimSpaceWeaverAppSdkDistributable Non contiene l'intero SimSpace Weaver SDK dell'app. La prima volta che utilizzi gli script SDK dell'SimSpace Weaver app per creare un progetto, gli script creano la cartella SimSpaceWeaverAppSdk -1.16.0 all'interno della cartella sdk e scaricano il resto dell'**SDK** dell'app al suo interno. SimSpace Weaver

Versioni AWS SimSpace Weaver

Miglioriamo continuamente AWS SimSpace Weaver. Devi scaricare l'SDK più recente SimSpace Weaver dell'app quando rilasciamo una nuova versione se desideri sfruttare le nuove funzionalità e gli aggiornamenti delle funzionalità. Per eseguire una simulazione esistente con una versione più recente, potrebbe essere necessario aggiornarne lo schema e il codice, quindi avviare una nuova istanza della simulazione. Non è necessario eseguire l'aggiornamento e puoi continuare a eseguire simulazioni esistenti con le versioni precedenti. Puoi controllare questa pagina per vedere quali sono le differenze tra le versioni. Tutte le versioni sono attualmente supportate.

Important

L'ultima versione della [Guida AWS SimSpace Weaver per l'utente](#) copre solo la versione più recente del servizio. È possibile trovare la documentazione per le versioni precedenti nel [catalogo delle AWS SimSpace Weaver guide](#), disponibile nella [pagina principale dedicata alla documentazione](#).

Versione più recente

L'ultima versione è: 1.16.0

Come trovare la tua versione attuale

Se hai creato una simulazione con l'SDK dell'SimSpace Weaver app, lo `create-project` script ha scaricato una versione delle librerie SDK in una sottodirectory del tuo `sdk-folder`. La sottodirectory che contiene le librerie SDK ha un nome che include il numero di versione dell'SDK: `SimSpaceWeaverAppSdk-sdk-version`. Ad esempio, le librerie per la versione 1.15.3 sono disponibili. `SimSpaceWeaverAppSdk-1.15.3`

Puoi anche trovare la versione del pacchetto distribuibile dell'SimSpace Weaver app SDK nel file di testo del tuo `app_sdk_distributable_version.txt` `sdk-folder`

Scarica la versione più recente

Utilizza uno dei seguenti link per scaricare la versione più recente.

- [Pacchetto completo distribuibile dell'app SDK](#)
- [Solo le librerie SDK dell'app](#)

Puoi anche scaricare il pacchetto completo distribuibile SimSpace Weaver dell'app SDK dalla [SimSpace Weaverconsole](#) in AWS Management Console Scegli Scarica app SDK dal pannello di navigazione.

Warning

Non utilizzarlo AWS CLI per scaricare nulla che sembri essere il pacchetto distribuibile dell'App SDK. Utilizza solo i link per il download in questa pagina o il link per il download nella console. Qualsiasi altro metodo o percorso di download non è supportato e potrebbe contenere codice obsoleto, errato o dannoso.

Risoluzione dei problemi relativi ai download dell'SDK delle app

Utilizziamo Amazon CloudFront (CloudFront) per distribuire i file.zip dell'SDK dell'app. Potrebbero verificarsi alcune delle seguenti situazioni.

- Il pacchetto scaricato non è la versione più recente
 - Se il file.zip scaricato non contiene la versione più recente, è possibile che la cache dell'CloudFront edge location non sia ancora aggiornata. Effettua nuovamente il download dopo 24 ore.
- Viene visualizzato un errore HTTP 4xx o 5xx utilizzando un link per il download
 - Riprova dopo 24 ore. Se ricevi lo stesso errore, utilizza il link Feedback nella parte inferiore della [SimSpace Weaverconsole](#) per segnalare il problema. Seleziona Segnala un problema come Tipo di feedback.
- Il tuo browser segnala che non è in grado di caricare la pagina
 - Potresti avere un problema di configurazione della rete locale o del browser. Verifica di poter caricare altre pagine. Svuota la cache del browser e riprova. Assicurati di non avere regole firewall che potrebbero bloccare l'URL di download.
- Viene visualizzato un errore quando si tenta di salvare il file
 - Controlla le autorizzazioni del file system locale per assicurarti di disporre delle autorizzazioni corrette per salvare il file.

- Il tuo browser visualizza AccessDenied
 - Se hai inserito manualmente l'URL nel browser, verifica che sia corretto. Se hai usato un link per il download, assicurati che qualcosa non interferisca con l'URL del browser; usa di nuovo il link.

Installa la versione più recente

Per installare la versione più recente

1. [Scarica la versione più recente.](#)
2. Decomprimi il SimSpaceWeaverAppSdkDistributable file.zip in una cartella.
3. Esegui `docker-create-image.bat` (o `docker-create-image.sh` per WSL) dalla cartella SDK dell'app decompressa dell'ultima versione. SimSpace Weaver
4. Utilizza la cartella SDK dell'SimSpace Weaverapp dell'ultima versione decompressa anziché la versione precedente.

Versioni del servizio

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.16.0	<p>Nuova funzionalità:</p> <ul style="list-style-type: none"> • Ora puoi utilizzare e le API di messaggistica nell'SDK dell'SimSpace Weaverapp per inviare e ricevere messaggi tra le tue app. Questa funzionalità è disponibile per le integrazioni C++, Python e 	12 febbraio 2024	Questa guida	<ul style="list-style-type: none"> • Pacchetto completo • Solo biblioteche <p>Vedi Risoluzione dei problemi.</p>

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	Unity e Unreal Engine 5. Per ulteriori informazioni, consulta Messaggistica .			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.3	<p>SimSpace Weaver Localaggi ornamento:</p> <ul style="list-style-type: none">L'abbiamo modificato SimSpace Weaver Local per allinearlo maggiormente allo sviluppo diCloud AWS. Queste modifiche influiscono sui progetti e sui flussi di lavoro di C++, Python, Unity e Unreal Engine per. SimSpace Weaver Local <p>Per ulteriori informazioni, consulta Differenze di sviluppo locale nella versione 1.15.3.</p>	4 dicembre 2023	Consulta il catalogo delle guide. AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.2	<p>Aggiornamento del pacchetto distribubile dell'App SDK:</p> <ul style="list-style-type: none">• Abbiamo aggiornato il file <code>Dockerfile</code> e per utilizzare la versione specifica richiesta di <code>cmake</code>. Le build dei container Docker potrebbero fallire senza questa modifica.	2 novembre 2023	Consulta il AWS SimSpace Weaver catalogo delle guide .	Non disponibile per il download
1.15.1	<p>Aggiornamento delle funzionalità:</p> <ul style="list-style-type: none">• Python SDK: questa versione corregge un problema che causava il fallimento delle simulazioni basate su Python in Cloud AWS. Per ulteriori informazioni, consulta le note sulla versione.	22 settembre 2023	Consulta il catalogo delle AWS SimSpace Weaver guide .	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.0	<p>Nuova funzionalità:</p> <ul style="list-style-type: none">• Python SDK: ora puoi sviluppare le tue simulazioni in Python. Il pacchetto distribuibile SimSpace Weaver dell'app SDK include un modello per un progetto Python di esempio e il relativo client di visualizzazione Python. Per ulteriori informazioni, consulta Uso di Python.	31 agosto 2023	Consulta il catalogo delle guide. AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.14.0	<p>Nuove caratteristiche:</p> <ul style="list-style-type: none">• Contenitori personalizzati: crea la tua immagine di contenuto basata su Amazon Linux 2 (AL2), archivia in Amazon Elastic Container Registry (Amazon ECR) e usala per eseguire le tue app in SimSpace Weaver Cloud AWS. Per ulteriori informazioni, consulta Contenitori personalizzati.• Domini spaziali multipli: crea più di un dominio spaziale in una simulazione. Separa la logica	26 luglio 2023	Consulta il AWS SimSpace Weaver catalogo delle guide .	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	<p>di simulazione invece di combinarla tutta in un'unica app spaziale. Assegna risorse diverse ai domini spaziali in base alle loro esigenze. Per ulteriori informazioni, consulta Configurazione dei domini spaziali.</p> <ul style="list-style-type: none">• Tick rate illimitato: consenti alla simulazione di funzionare alla stessa velocità di esecuzione del codice. Imposta l'orologio della simulazione in modo che invii il segno di spunta successivo non appena tutte le app terminano le operazioni di commit per il segno di spunta corrente. Per ulteriori informazi			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	<p>oni, consulta Orologio.</p> <p>SimSpace WeaverSDK per app:</p> <ul style="list-style-type: none">• Il valore di <code>tick_rate</code> è ora una stringa. Il valore deve includere virgolette doppie ("). Il tick rate per le versioni precedenti è ancora un numero intero. Per ulteriori informazioni, consulta Orologio.			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.13.1	<p>SimSpace WeaverSDK dell'app:</p> <ul style="list-style-type: none">• Aggiornamento delle funzionalità: la creazione del progetto ora funziona correttamente con il PathfindingSampleUnreal modello.	7 giugno 2023	Consulta il catalogo della AWS SimSpace Weaver guida .	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.13.0	<p>SimSpace WeaverAPI di servizio:</p> <ul style="list-style-type: none"> • Nuova CreateSnapshot azione • Modifiche all'StartSimulation azione: <ul style="list-style-type: none"> • Aggiunto un <code>SnapshotS3Location</code> parametro per iniziare da un'istantanea. • Il <code>SchemaS3Location</code> parametro è ora facoltativo. • Modifiche all'DescribeSimulationOutput: <ul style="list-style-type: none"> • <code>SchemaError</code> è obsoleta • È stato aggiunto un 	29 aprile 2023	Consulta il catalogo della guida. AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	<p>StartError campo.</p> <ul style="list-style-type: none">È stato aggiunto un SnapshotS3Location campo.È stato aggiunto uno stato di SNAPSHOT_IN_PROGRESS simulazione.Nuovo tipo di S3Destination dati <p>SimSpace WeaverConsole :</p> <ul style="list-style-type: none">Nuova funzionalità per creare istantanee.Nuova funzionalità per avviare una simulazione da un'istanza.			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	<p>SimSpace WeaverSDK per app:</p> <ul style="list-style-type: none"> • Nuovi script per supportare le istantanee <ul style="list-style-type: none"> • create-snapshot- <i>project-name</i> .bat • start-from-snapshot- <i>project-name</i> .bat • quick-start-from-snapshot- <i>project-name</i> -cli.bat • list-snapshots- <i>project-name</i> .bat • <i>I progetti ora utilizzano un singolo bucket Amazon S3 per progetto: weaver- -lowercase -project-</i> 			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	<p><i>name account-number - region</i></p> <p>Per ulteriori informazioni sugli snapshot, consulta Snapshot.</p>			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.3	<p>SimSpace WeaverSDK dell'app:</p> <ul style="list-style-type: none"> • I seguenti script ora supportano il <code>--maximum-duration</code> parametro: • <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>Per ulteriori informazioni, consulta Durata massima di</p>	27 marzo 2023	<p>Consulta il catalogo delle AWS SimSpace Weaver guide.</p>	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	un flusso di un flusso.			
1.12.2	SimSpace WeaverSDK dell'app: <ul style="list-style-type: none">• Correzione di bug: docker-create-image.bat ora funziona correttamente.	1 marzo 2023	Consulta il catalogo delle AWS SimSpace Weaver guide.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.1	<p>SimSpace WeaverSDK dell'app:</p> <ul style="list-style-type: none">• Gli script ora accettano un AWS CLI profilo da utilizzare per AWS l'autenticazione.• Gli script ora supportano AWS IAM Identity Center l'autenticazione. AWS <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none">• Correzione di bug: <code>Api::BeginUpdateWillBlock</code> ora restituisce correttamente <code>true</code> se tutte le app spaziali non hanno aderito alla simulazione.	28 febbraio 2023	Consulta il AWS SimSpace Weavercatalogo delle guide .	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.0	Versione per disponibilità generale (GA)	29 novembre 2022	Consulta il catalogo delle AWS SimSpace Weaver guide .	Non disponibile per il download

AWS SimSpace Weaver versione 1.15.1

Questa versione è un aggiornamento richiesto per Python SDK originariamente rilasciato nella SimSpace Weaver versione 1.15.0. Risolve un problema di mancata corrispondenza delle versioni che causava il fallimento delle simulazioni basate su Python in Cloud AWS. Usa questa versione invece della 1.15.0.

Aggiorna un progetto Python esistente a 1.15.1

Se disponi di un progetto Python esistente creato con la versione 1.15.0 Python SDK, devi eseguire i seguenti passaggi per aggiornarlo alla 1.15.1 in modo che possa essere eseguito in Cloud AWS.

Invece di seguire questa procedura, puoi anche creare un nuovo progetto Python con l'SDK Python 1.15.1 e spostare il codice personalizzato nel nuovo progetto.

Per aggiornare un progetto Python 1.15.0 a 1.15.1

1. Vai alla cartella del tuo progetto Python.
2. In `src/PythonBubblesSample/bin/run-python` modifica la seguente riga:

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

alla seguente:

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. In `CMakeLists.txt` eliminare le seguenti righe:

- ```
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)
```

Risoluzione dei problemi per la versione 1.15.1

Dopo aver aggiornato una simulazione Python 1.15.0, non riesce ad avviarsi in Cloud AWS

Sintomi: dopo circa 5-10 minuti dall'avvio della simulazione, il registro di gestione della simulazione riporta un messaggio e lo stato della simulazione è. `internal error FAILED`

Ciò può accadere se un file di libreria dell'SDK Python 1.15.0 è incluso in un file zip dell'app. Assicurati di aver completato i passaggi per aggiornare il progetto e assicurati che non sia presente nei file zip o che `libweaver_app_sdk_python_v1.so` non vi sia alcun riferimento.

Domande frequenti sulla versione 1.15.1

Questa versione influisce su qualcosa di diverso dall'SDK Python?

No.

Devo aggiornare alla versione 1.15.1?

Non è necessario eseguire l'aggiornamento alla versione 1.15.1 se non si intende utilizzare Python per le app spaziali. Se hai eseguito l'aggiornamento alla 1.15.0, le simulazioni basate su Python non verranno eseguite in. Cloud AWS Ti consigliamo di eseguire l'aggiornamento alla versione 1.15.1 se usi 1.15.0.

Cos'è `$LD_LIBRARY_PATH`?

È la posizione dell'SDK Python quando la simulazione viene eseguita in. Cloud AWS Questa è una novità per 1.15.1. Abbiamo apportato questa modifica per evitare problemi futuri con la versione di Python. Il collegamento a quella directory è funzionalmente lo stesso del collegamento alla versione 1.15.0. `libweaver_app_sdk_python_v1.so`

Cronologia dei documenti per AWS SimSpace Weaver

Nella seguente tabella sono descritte importanti modifiche alla documentazione di SimSpace Weaver.

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
12 febbraio 2024	Contenuti aggiornati	Aggiornato il Versioni AWS SimSpace Weaver capitolo per la versione 1.16.0.	N/D
12 febbraio 2024	Nuovo contenuto	Aggiunta la Messaggistica sezione come parte della versione 1.16.0. Questa sezione descrive le API di messaggistica aggiunte all'SDK dell'SimSpace Weaverapp. Puoi utilizzare queste API per inviare e ricevere messaggi tra le tue app.	N/D
12 febbraio 2024	Contenuti aggiornati	Aggiornato il SimSpace Weaverriferimento allo schema di simulazione capitolo per la versione 1.16.0.	N/D
12 febbraio 2024	Contenuti aggiornati	Aggiunte quote di servizio per la messaggistica al SimSpace Endpoint e quote Weaver capitolo.	N/D
12 febbraio 2024	Nuove guide	Suddividi il contenuto delle versioni precedent	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
		<p>i alla 1.16.0 in una guida separata. È stato aggiunto il catalogo delle AWS SimSpace Weaver guide (disponibile nella pagina principale della documentazione) per accedere alle guide delle versioni precedenti.</p>	
4 dicembre 2023	Contenuti aggiornati	<p>Aggiornato il Versioni AWS SimSpace Weaver capitolo per la versione 1.15.3.</p>	N/D
4 dicembre 2023	Contenuti aggiornati	<p>È stato aggiornato il Versioni AWS SimSpace Weaver capitolo per includere le istruzioni di installazione per la versione più recente.</p>	N/D
4 dicembre 2023	Contenuti aggiornati	<p>Aggiornato il Service Quotas per SimSpace Weaver Local.</p>	N/D
4 dicembre 2023	Contenuti nuovi e aggiornati	<p>Ha ristrutturato la Sviluppo locale sezione e ha aggiunto una nuova pagina che descrive le differenze SimSpace Weaver Local introdotte nella versione 1.15.3.</p>	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
7 novembre 2023	Contenuti aggiornati	Sono state aggiornate le istruzioni per configurare Docker e WSL per utilizzare il link/URL di download diretto per l'SDK dell'app. Per ulteriori informazioni, consulta Configurazione dell'ambiente locale per SimSpace Weaver .	N/D
2 novembre 2023	Contenuti aggiornati	È stata aggiornata la pagina delle versioni del servizio per la versione 1.15.2. Per ulteriori informazioni, consulta Versioni del servizio .	N/D
23 ottobre 2023	Contenuti aggiornati	È stata aggiornata la pagina delle versioni del servizio con nuove istruzioni per scaricare il pacchetto distribuibile SDK dell'app. I clienti devono ora utilizzare solo uno dei nostri link per il download diretto approvati e non utilizzare lo AWS CLI per scaricare il pacchetto distribuibile SDK dell'app. Per ulteriori informazioni, consulta Scarica la versione più recente .	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
22 settembre 2023	Nuovo contenuto	È stata aggiunta una pagina di note sulla versione con le istruzioni di aggiornamento per la versione 1.15.1. Per ulteriori informazioni, consulta AWS SimSpace Weaverversione 1.15.1 .	N/D
10 settembre 2023	Nuovo contenuto	È stata aggiunta una sezione di risoluzione dei problemi per le situazioni in cui AWS CLI non riconosce SimSpace Weaver. Per ulteriori informazioni, consulta Il AWS CLI non riconosce simspaceweaver .	N/D
10 settembre 2023	Contenuti aggiornati	Istruzioni di installazione aggiornate per il AWS CLI WSL. Per ulteriori informazioni, consulta Configurazione Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL) .	N/D
7 settembre 2023	Aggiornamento dell'API	BucketName e ora ObjectKey sono necessari per il tipo di dati S3Location . BucketName è ora necessario per il tipo di dati S3Destination .	AWSSDK: 2023-09-07

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
31 agosto 2023	Nuovo contenuto	Aggiunta una nuova sezione per la versione 1.15.0.: Uso di Python	N/D
15 agosto 2023	Contenuti aggiornati	Le istruzioni di download sono state aggiornate Versioni AWS SimSpace Weaver per elencare solo i SimSpace Weaver bucket Amazon S3 ufficiali. Altre posizioni di download non sono controllate da AWS e potrebbero contenere codice dannoso.	N/D
26 luglio 2023	Contenuti aggiornati	Aggiornato Orologio .	N/D
26 luglio 2023	Contenuti aggiornati	Aggiornato Configurazione dei domini spaziali .	N/D
26 luglio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Contenitori personalizzati ,	N/D
26 luglio 2023	Contenuti aggiornati	Aggiornato Versioni AWS SimSpace Weaver per la versione 1.14.0.	N/D
6 luglio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Pathfindi ngSample il client della console non riesce a connettersi ,	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
7 giugno 2023	Contenuti aggiornati	Aggiornato Versioni AWS SimSpace Weaver per la versione 1.13.1.	N/D
15 maggio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Utilizzo delle istantanee con AWS CloudFormation ,	N/D
29 aprile 2023	Nuovo contenuto	Aggiunto contenuto per la versione 1.13.0. Per ulteriori informazioni, consulta Versioni AWS SimSpace Weaver .	AWSSDK: 2023-04-28
27 marzo 2023	Nuovo contenuto	Aggiunta una sezione che descrive la durata massima delle simulazioni. Sono state aggiunte delle note nei tutorial per la versione 1.12.3, che hanno aggiunto il supporto per il <code>--maximum-duration</code> parametro agli script SDK delle app. SimSpace Weaver	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
9 marzo 2023	Contenuti modificati	Abbiamo chiarito che forniamo solo istruzioni per l'attivazione Windows e per Docker Windows Subsystem for Linux (WSL) e che WSL (e qualsiasi altro ambiente Linux) non è supportato.	N/D
28 febbraio 2023	Nuovo contenuto	È stato aggiunto un capitolo che descrive le versioni. SimSpace Weaver	N/D
28 febbraio 2023	Contenuti modificati	È stato modificato il contenuto sull'autenticazione per includere l'uso di AWS IAM Identity Center profili denominati per AWS Command Line Interface (AWS CLI).	N/D
17 febbraio 2023	Nuovo contenuto	È stata aggiunta una sezione sulla gestione delle risorse con AWS CloudFormation.	N/D
23 gennaio 2023	Nuovo contenuto	Sono state aggiunte istruzioni per il debug delle simulazioni locali.	N/D
29 novembre 2022	Avvio del servizio	Rilasciati la Guida per l'utente e il riferimento API per SimSpace Weaver	AWSSDK: 2022-11-29

Glossario

Questo glossario definisce i termini specifici di AWS SimSpace Weaver.

Per la terminologia AWS più recente, consulta il [glossario AWS](#) nei AWS General Reference (Riferimenti generali AWS).

A

app	Codice eseguibile (chiamato anche binari) creato dall'utente. Il termine app può fare riferimento al codice o a un'istanza in esecuzione di tale codice. Un'app incapsula il comportamento di simulazione. Le app creano, eliminano, leggono e aggiornano le entità.
SDK per app	Un kit di sviluppo software (SDK) con cui integrare un'app. SimSpace Weaver L'SDK fornisce API per la lettura e la scrittura dei dati delle entità e il monitoraggio del tempo di simulazione. Per ulteriori informazioni, consulta SimSpace Weaver app SDK .

C

cliente	Processi (o relative definizioni) che esistono al di fuori della simulazione SimSpace Weaver e che interagiscono con essa tramite un' app personalizzata o un'app di servizio . È possibile utilizzare un client per visualizzare o modificare lo stato della simulazione.
orologio	Un'astrazione dei processi SimSpace Weaver di pianificazione interni. L'orologio pubblica segni di spunta nelle app per mantenere la sincronizzazione dell'ora. Ogni simulazione ha il suo orologio.
frequenza	Il numero di tick al secondo che l'orologio pubblica nelle app . Per ulteriori informazioni sulle frequenze di clock supportate, consulta. SimSpace Endpoint e quote Weaver
frequenza di ticchettamento	Vedi la frequenza dell'orologio .
unità di risorse di calcolo	Un'unità di risorse di calcolo (processore e memoria) su un lavoratore. A una singola istanza di un' app viene normalmente assegnata 1 unità di

risorse di calcolo. Puoi allocare più di 1 unità di risorse di calcolo per ogni app.

app personalizzata Un tipo di [app](#) che usi per leggere e interagire con lo stato della simulazione. Le app personalizzate possono creare entità nella simulazione ma non le possiedono. Quando un'app personalizzata crea un'entità, deve trasferire l'entità nel dominio [spaziale](#). Puoi controllare il ciclo di vita di un'app personalizzata utilizzando le API dell'app. Per ulteriori informazioni sulle SimSpace Weaver API, consulta. [SimSpace Weaver Riferimenti API](#)

dominio personalizzato Un [dominio](#) che contiene [app personalizzate](#).

partizione personalizzata La [partizione](#) di un'app [personalizzata](#).

D

scadenza Un [momento effettivo entro il](#) quale un'operazione (ad esempio l'elaborazione di un segno di [spunta](#)) deve essere completata.

dominio Un gruppo di istanze di [app](#) che eseguono lo stesso codice eseguibile (app binario) e hanno le stesse opzioni di avvio.

E

endpoint (servizio) Un nome di dominio completo (FQDN) utilizzato da programmi (come ilAWS Command Line Interface) per connettersi al servizio. SimSpace Weaver

endpoint (simulazione) Un indirizzo IP e un numero di porta utilizzati dai client per connettersi a una simulazione. Puoi configurare gli endpoint su [app personalizzate e app di servizio](#).

entità oggetti di dati del cliente (o relative definizioni). Le entità possono essere statiche (rimanere in una posizione) o dinamiche (spostarsi nello spazio di simulazione). Ad esempio, persone ed edifici in una simulazione.

I

indice (simulazione) Descrizione delle proprietà spaziali di una simulazione, compresi i confini spaziali e il sistema di coordinate.

L

ciclo di vita (di un'app) Una descrizione dei passaggi logici previsti che un'app deve seguire durante una simulazione. I cicli di vita sono gestiti (SimSpace Weaver avvia e arresta l'app) o non gestiti (l'utente avvia e arresta l'app).

load (dati del campo dell'entità) Leggi i dati del campo di [entità](#) da [State Fabric](#).

P

partizione Un segmento di memoria condivisa su un [lavoratore](#). [Ogni partizione contiene un sottoinsieme discreto di entità all'interno di un dominio](#). A ogni [app](#) è assegnata una partizione. Un'app possiede tutte le entità nella sua partizione. Quando un'app crea un'entità, la crea nella sua partizione. Quando le entità si spostano da una partizione all'altra, la proprietà viene trasferita dall'app della partizione di origine all'app della partizione di destinazione.

R

unità di risorse Per informazioni, consulta [???](#).

S

schema Un documento YAML o JSON che descrive la configurazione di una simulazione. SimSpace Weaver [utilizza uno schema per creare una risorsa di simulazione](#).

app di servizio Un tipo di [app](#) che si utilizza per leggere e interagire con lo stato della simulazione. [Le app di servizio possono creare entità nella simulazione ma devono trasferirle nel dominio spaziale](#). SimSpace Weaver [gestisce il](#)

	ciclo di vita di un'app di servizio e ne avvia 1 (o più, come specificato nello schema di simulazione) su ogni lavoratore della simulazione.
dominio di servizio	Un dominio che contiene app di servizio .
partizione di servizio	La partizione di un'app di servizio .
simulazione (risorsa)	Un'astrazione di un cluster di calcolo che gestisce uno spazio virtuale simulato. Puoi avere più simulazioni. Si configura una simulazione utilizzando uno schema.
app spaziale	Un tipo di app che incapsula la logica di simulazione di base. Ogni app spaziale possiede 1 (e solo 1) partizione.
dominio spaziale	Un dominio che contiene app spaziali .
partizione spaziale	La partizione di un'app spaziale.
State Fabric	SimSpace Weaver è un database in memoria. State Fabric memorizza lo stato delle simulazioni, incluse entità e dati interni SimSpace Weaver.
store (dati del campo dell'entità)	Scrivi i dati del campo dell'entità in State Fabric .
abbonamento	Una richiesta a lungo termine per una specifica istanza di app per ricevere dati da un' area di abbonamento . L'app per gli abbonamenti utilizza un abbonamento per scoprire le modifiche alle entità all'interno dell'area di abbonamento.
area di sottoscrizione	Una regione bidimensionale dello spazio di simulazione. Un abbonamento si riferisce a un'area di abbonamento. Un'area di sottoscrizione può estendersi su più di una partizione e includere anche parti di partizioni. Un'area di sottoscrizione è continua entro i limiti definiti.

T

segno di spunta	Un valore temporale discreto (orario dell'orologio da parete o tempo di simulazione). Le app possono iterare più velocemente della durata del segno di spunta, ma ci si aspetta che scrivano tick specifici entro scadenze specifiche. Tutte le operazioni per tutte le app relative a un
-----------------	---

determinato segno di spunta devono essere completate prima che possa iniziare il segno di spunta successivo.

tasso di selezione

Vedi la frequenza.

ora (effettiva)

L'ora attuale dal punto di vista della realtà. SimSpace Weaver utilizza un timestamp POSIX a 64 bit che è il numero di nanosecondi dall'epoca Unix (January 1, 1970, 00:00:00 UTC).

tempo (simulazione)

L'ora corrente dal punto di vista della simulazione. SimSpace Weaver utilizza un contatore di tacche logico intero a 64 bit, che potrebbe non corrispondere direttamente all'ora effettiva.

W

lavoratore

Un'istanza Amazon Elastic Compute Cloud (Amazon EC2) Elastic Compute EC2) che esegue codice di simulazione.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.